VILNIUS UNIVERSITY

Paulius Danėnas

SUPPORT VECTOR MACHINES BASED CLASSIFIERS IN INTELLIGENT DECISION SUPPORT SYSTEM FOR CREDIT RISK EVALUATION

PhD thesis
Physical sciences, Informatics (09P)

Vilnius 2013

Thesis was prepared during 2008-2012 in Vilnius University

Thesis supervisor:

prof. dr. Gintautas Garšva (Vilnius University, Physical Sciences, Informatics - 09P)

Consultant:

prof. habil. dr. Rimvydas Simutis (Kaunas University of Technology, Physical Sciences, Informatics - 09P)

VILNIAUS UNIVERSITETAS

Paulius Danėnas

ATRAMINIŲ VEKTORIŲ MAŠINOMIS GRINDŽIAMI KLASIFIKAVIMO METODAI INTELEKTUALIOJE SPRENDIMŲ PARAMOS SISTEMOJE KREDITO RIZIKOS VERTINIMUI

Daktaro disertacija
Fiziniai mokslai, informatika (09P)

Vilnius 2013

Disertacija rengta 2008-2012 Vilniaus universitete

Mokslinis vadovas:

prof. dr. Gintautas Garšva (Vilniaus universitetas, fiziniai mokslai, informatika - 09P)

Konsultantas:

prof. habil. dr. Rimvydas Simutis (Kauno technologijos universitetas, fiziniai mokslai, informatika - 09P)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

|X|	Size of vector X
ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under Curve measure
BPNN	Backpropagation Neural Network
BVM	Ball Vector Machines
C-SVC	C-Support Vector Classification (original SVM classifier formulation)
CART	Classification and Regression Tree algorithm
CBR	Case Based Reasoning
CSV	Comma Separated Values
CV	Cross-Validation
CVM	Core Vector Machines
CVM-LS	Least Squares Core Vector Machines
DB	Database
DDD	Domain Driven Design
dim(X)	Dimension of feature vector X
DM	Data Mining
DSS	Decision Support System
DT	Decision Tree
FDD	Feature Driven Development
FN (FNR)	False Negative (False Negative Rate)
FP (FPR)	False Positive (False Positive Rate)
FS-SVM	SVM with feature selection
$FS\text{-}SVM^{DA}$	SVM with feature selection classifier based on discriminant analysis evaluations
$FS\text{-}SVM^{SWTest}$	Proposed technique using feature selection, SVM clasification and sliding window approach for testing and evaluation
FSVM	fuzzy SVM
GA	Genetic Algorithm
GA-LinSVM	Proposed linear SVM optimized by Genetic Algorithm
GA-SVM	SVM optimized by Genetic Algorithm
GDM-BME	group decision making and binary majority evaluation based algorithm
GS	Genetic Search
GUI	Graphical User Interface
IDSS	Intelligent Decision Support System
IR	Information Retrieval
KNN	k-nearest Neighbor algorithm
L2-RLR	L2-regularized loss regression (SVM based algorithm in LIBLINEAR package)
LDA	Linear Discriminant Analysis
LR	Linear Regression

LS-SVM	Least Squares SVM
LSVM	Linear SVM
LVQ	Learning Vector Quantization
MDA	1) Multiple Discriminant Analysis
	2) Model Driven Architecture
MDD	Model Driven Development
MDDB	Multidimensional Database
ML	Machine Learning
MLP	Multilayer Perceptron
N/A	Not available (e.g., not given or described in paper or book)
NN	Neural Network
OAA (OVA)	one-against-all (one-vs-all) training principle
OAO (OVO)	one-against-one (one-vs-one) training principle
OLAP	Online Analytic Processing
PCA	Principal Component Analysis
PLS	Partial Least Squares
PNN	Probabilistic Neural Network
PSO	Particle Swarm Optimization
PSO-LinSVM	Proposed linear SVM optimized by Particle Swarm Optimization
PSO-SVM	SVM optimized by Particle Swarm Optimization
QDA	Quadratic Discriminant Analysis
RBF	Radial Basis Function
RS	Rough Sets
RS-SVM	Rough Sets SVM
RVM	Relevance Vector Machines
SA	Simulated Annealing
SEC	U.S. Securities and Exchange Commission
SMO	Sequential Minimal Optimization algorithm, proposed by Platt
SOM	Self-Organizing Map
SQL	Structurized Query Language
SVM	Support Vector Machines
SVM-RFE	Support Vector Machine - Recursive Feature Elimination algorithm
SV	Support Vector
TN (TNR)	True Negative (True Negative Rate)
TP (TPR)	True Positive (True Positive Rate)
UML	Unified Modelling Language
XBRL	eXtensible Business Reporting Language financial reporting standard
XML	Extensible Markup Language
XML Schema	XML standard proposed by Microsoft Corporation

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

INTRODUCTION

Financial crisis of 2008 and 2010 exposed the need of more precise techniques for credit risk evaluation, together with tools necessary to develop them. The main objective of credit risk management is to evaluate possibility that debtor will fail to meet his obligations before by agreed terms, which helps to reduce the probability to lose invested money. Minimization of such debts is critical for managing risk and optimal capital allocation in financial institutions as Basel II capital accord defines new regulatory standards which have to be met. Thus proper, efficient and effective credit risk evaluation tools for credit risk, such as highly discriminative credit scoring models, are mandatory for every financial institution. This problem is solved in multiple dimensions, including debtor type (individual, organization, government), financial instrument type (loan issue, financial derivatives), modeling techniques (parametric, non-parametric, VaR, probability default and etc.), length and others. Classification technique, with emphasis to associate an obligor with one of risk classes or identify whether it is prone to bankruptcy, is one of the most popular techniques widely applied and discussed in various papers as a solution to credit risk related problems, with various modern and complex techniques including statistical, econometric and artificial intelligence based ones. Support Vector Machines (abbr. as SVM) at the moment of writing is one of most widely developed, researched and applied techniques in this field, proposed by Vapnik [225] and further developed, extended and discussed in books and papers by Scholkopf et al. [191, 192], Cristianini et al. [54], Baesens, van Gestel et al. [17, 223], Mangasarian et al. [85, 86, 153, 154], Huang et al. [116], Lai, Yu et al. [253], Chang et al. [36], Steinwart et al. [202] and etc. It is used to solve various classification problems in different domains, including bioinformatics and computational biology [25, 250], document classification [122, 129] and etc. Lithuanian scientists have also worked with SVM based classifiers, e.g., Varoneckas applied SVM to sleep stages recognition [226], Martisius et al. [156] and Verikas et al. used it in medicine domain [229]. Various approaches which combine SVM with other techniques or apply inner modifications for initial SVM algorithm in order to obtain faster, more accurate and efficient solutions are permanently proposed; they are also reviewed further in this work. Balthazar refers that SVM-based model is used by Standard & Poor's rating company [18]. Yet complexity of this technique, various analytical, computational and

development issues makes this task more sophisticated; this is one of the reasons which make SVM “a classification technique for experts”. Identification and analysis of such problems as well as possible solutions to overcome such barriers are also important part of overall solutions; therefore, a lot of attention in this work is given to practical implementation aspects.

1.1. Research problem

As it is mentioned in the introduction, this topic is widely researched and important for financial institutions, although development of SVM-based algorithms is important for the whole computational science. The results of this research can be applied in practice, including integration of developed techniques into intelligent decision support system for both scientific and business purposes. Increasing amount of available open and linked data (including financial) offers new possibilities to develop new models or improve existing by integrating new knowledge within them, combining available expert knowledge and experience with this data. Integration with various Semantic Web technology based standards from financial domain becomes important in this context. The lack of research aimed at intelligent financial decision support system development, including both integration of complex banking domain frameworks such as Extensible Business Reporting Language (XBRL), and development methodologies for similar systems, is also one of the main inspirations to propose possible implementation possibilities, with both theoretical and engineering viewpoints. Basel II regulatory framework support for XBRL is also discussed in various papers from both theoretical and engineering viewpoints, including development of Web Services for semi-automated or automated reporting [91] or COREP/FINREP taxonomies used in banking domain which fully cover Basel II Pillar 1 [70]; however, much less attention is given for XBRL-driven decision support based on statistical and machine learning techniques. Therefore, design and research of resulting DSS framework is also relevant for software engineering science, offering new viewpoints for engineering of complex modern decision support systems, which can be further developed, reused and enhanced.

1.2. The object of research

This work analyzes intelligent credit risk evaluation techniques based on Support Vector Machines. Therefore, the main object of this work is hybrid Support

Vector Machines based classification techniques for credit risk evaluation and bankruptcy prediction. A framework comprising such and similar techniques, financial standards, design and development methodology for intelligent systems based on these techniques, possible implementation scenario is also defined as secondary research object.

1.3. The goal and objectives of the research

The aim of the research is to propose an approach to develop Support Vector Machines classification based classifier for credit risk evaluation which combines existing financial data and external evaluations (e.g., expert evaluations) available. Another goal of this research is propose and develop a framework for intelligent decision support systems for financial domain which integrates financial standards, a solution based on proposed classifier, design and development methodology, together with main components which are common for such type of DSS.

The objectives of the dissertation are as following:

1. Investigate statistical, econometric and artificial intelligence techniques, current developments and previous works in credit risk domain based on these techniques, identify their main advantages.
2. Analyse developed structures of decision support systems for researched field, financial standards and regulations, their purpose and fields of application, identify and propose possible ways of their integration and application in intelligent decision support system for credit risk domain.
3. Develop hybrid intelligent classification method and/or approach, based on artificial intelligence techniques, for researched problem.
4. To carry out experimental evaluation of developed techniques, analyse and evaluate obtained results.
5. Design and develop a framework for intelligent decision support system for credit risk evaluation which includes developed approaches, components, common for such systems, integration of financial standards, design, development and implementation scenarios.
6. Implement a decision support system using the designed framework.

1.4. Research methodology and tools

The following methods were used in the research: general cognition

(formulation of research tasks and aims of research, collection and analysis of information; generalization; formulation of conclusions); general scientific research techniques such as induction, deduction, comparison (techniques, characteristics, similarities, differences); data analysis and modeling; structuring, grouping, generalization, abstraction and presentation.

Open source machine learning framework WEKA [236], SVM toolboxes LibSVM [36], LIBLINEAR [80] were used to implement the algorithms and techniques presented in this research. These tools, together with RapidMiner [178] and various SVM implementations, were also used in research for benchmarking implementations. Technical computing system MATLAB was used for initial developing, modelling and testing PSO-LinSVM algorithm. Subsets of SEC EDGAR database, comprising financial ratios from yearly and quarterly balance and income statements in 1999-2008 of 9365 USA based companies from 9 sectors, UCLA LoPucki bankruptcy database, which contains actual bankruptcy data of 911 USA bankruptcy companies (with 253 companies directly mapped to EDGAR database used in the research), and Australian and German credit datasets from UCI machine learning repository (with 690 and 1000 instances respectively) were used for experimental research of developed algorithms and techniques. UML and BPMN notations were used for framework and method design; diagrams for design and development methodology were prepared on custom notation based on UML and recommendations of Domain Driven Design author given in his book [79]. Graphical modelling tools such as MagicDraw and Microsoft Visio were used to develop the diagrams.

1.5. The statements of the thesis

1. Particle Swarm Optimization and linear Support Vector Machines based classifier, which can automatically select optimal classifier together with its parameters from a set of classifiers with the same set of parameters, can perform efficiently with both small and large datasets.

2. The developed classification technique, comprising feature selection, SVM classification and sliding window testing principle, can be used to develop and test classification model for credit risk domain.

3. Integration of XBRL financial standard to decision support for financial domain can improve model development process with additional data variables,

enable automated import of standardized and structured financial data, real-time model development and update.

4. A framework integrating proposed techniques, financial standards, design and development methodology, possible implementation scenario based on cross-platform and data source independency is an important tool to develop modern intelligent decision support systems for credit risk evaluation.

1.6. Scientific significance of this work

New hybrid classification technique PSO-LinSVM, which uses particle swarm optimization based procedure for automatic selection of linear SVM classifier, is proposed in this work. Differently from techniques proposed previously, this algorithm selects linear SVM classifier together with its complexity and bias parameters from a set of linear SVM classifiers with these parameters. Its classification efficiency is tested with datasets of various sizes. Proposed technique can be used to solve classification problems in various domains such as finance, text analysis, bioinformatics and etc. This work also proposes credit risk evaluation technique, based on discriminant models or external evaluations, feature selection, classification and sliding window testing approach. Differently from previous techniques, this approach enables testing of developed model using data from one or more sequential periods which helps to evaluate its performance in several periods. Research context, such as large amount and dimensionality of used data, integration with external data sources and standards is also important as it is not typical for such research but becomes relevant as the number of available data sources and amounts of data tend to rise. Proposed framework for decision support system, together with design and development methodology, are important for software engineering science as they describe framework for development of distributed component and computational intelligence based systems, main components and processes, using researched DSS for credit risk evaluation as case study. This helps to enhance development of such systems using principles of component-based software engineering.

1.7. Practical results of proposed work

Proposed techniques might be used to develop models based on external evaluators, exploiting existing data to produce new models or improve existing. They

can also be used: to map rating data (internal or external) to existing financial data, identify their inner dependencies and help to improve quality of financial analysis as well as identify key factors; to embed expert knowledge and experience into a particular data-driven model which can be moved to a new environment. Developed intelligent model may also be used to evaluate the instances which cannot be evaluated by the expert either because of the missing data or mathematical problems (e.g., division by zero), as well as improve quality of base evaluator.

The proposed PSO-LinSVM classification technique, based on Particle Swarm Optimization and linear SVM, can be applied to solve classification problems in any domain. The developed platform and data source independent framework might be used to develop complex, with financial standards integrated, modern decision support system for credit risk evaluation. The described design and development methodology can be adapted and applied to develop other large scale DSS.

1.8. Presentation and approbation of results

Research results were presented in "Transformations in Business & Economics" international journal, included in Scientific Master Journal List (ISI), international and local conferences.

Conferences

1. Information Society and University Studies (IVUS) 2010, Kaunas, Lithuania, 2010.
2. 13th International Conference on Business Information Systems, Berlin, Germany, 2010.
3. 17th International Conference on Information and Software Technologies (IT 2011), Kaunas, Lithuania, 2011.
4. Information Technology, 16th Conference for Master and PhD students, Kaunas, Lithuania, 2011.
5. International Conference on Computational Science (ICCS 2011), Singapore, 2011.
6. ICAI'11 - International Conference on Artificial Intelligence, Las Vegas, USA, 2011.
7. 3rd International Workshop on Methods of Data Analysis for Information Systems, Druskininkai, Lithuania, 2011.

8. Information Technology, 17th Conference for Master and PhD students, Kaunas, Lithuania, 2012.

9. 15th International Conference on Business Information Systems, Vilnius, Lithuania, 2012.

10. International Conference on Computational Science (ICCS 2012), Omaha, USA, 2012.

11. 14th International Conference on Enterprise Information Systems (ICEIS 2012), Wrocław, Poland, 2012.

12. 18th International Conference on Information and Software Technologies (ICIST 2012), Kaunas, Lithuania, 2012.

Publications

In international journals, which are included in Scientific Master Journal List (ISI):

1. Danenas P., Garsva G. Support Vector Machines and their Application in Credit Risk Evaluation Process. Transformations in Business & Economics (2009), Vol. 8, No. 3 (18), pp. 46-58, ISSN 1648-4460.

2. Garsva G., Danenas P. XBRL Integration Into Intelligent System For Credit Risk Evaluation. Transformations in Business & Economics (2011), Vol. 10, No. 2 (23), pp. 88-103, ISSN 1648-4460.

In proceedings of scientific conferences, indexed in Scientific Master Journal Proceeding List (ISI):

1. Danenas P., Garsva G. Credit risk evaluation using SVM-based classifier. Lecture notes in business information processing (2010), Berlin, Springer, Vol. 57, Part 1, pp. 7-12, ISBN 978-3-642-15401-0

2. Danenas P., Garsva G., Gudas S. Credit Risk Evaluation Model Development Using Support Vector Based Classifiers. Procedia Computer Science, Vol. 4 (2011), Elsevier, pp. 1699-1707, ISSN 1877-0509

3. Danenas P., Garsva G. SVM and XBRL Based Decision Support System for Credit Risk Evaluation. Proceedings of the 17th International Conference on Information and Software Technologies (IT) (2011), Technologija, Kaunas, Lithuania, pp. 190-198, ISSN 2029-0020.

4. Danenas P., Garsva G. Simutis R. Development of Discriminant Analysis and Majority-Voting Based Credit Risk Assessment Classifier. Proceedings of the 2011 International Conference on Artificial Intelligence (ICAI 2011), CSREA Press,

Vol.1, pp. 204-209, ISBN: 1-60132-183-X, 1-60132-184-8 (1-60132-185-6).

5. Danenas P., Garsva G. Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach. Procedia Computer Science Vol. 9 (2012), Elsevier, pp. 1324 – 1333, ISSN: 1877-0509.

6. Danenas P., Garsva G. Credit Risk Modeling of USA Manufacturing Companies Using Linear SVM and Sliding Window Testing Approach. Lecture Notes in Business Information Processing, Vol. 117, Part 8 (2012), pp. 249-259, ISBN: 978-3-642-30359-3.

7. Danenas P., Garsva G. PSO-based Linear SVM Classifier Selection for Credit Risk Evaluation Modeling Process. Proceedings of 14th International Conference on Enterprise Information Systems (ICEIS 2012), Vol. 1 (2012), SciTePress, ISBN: 978-989-8565-10-5.

8. Danenas P., Garsva G. Domain Driven Development and Feature Driven Development for Development of Decision Support Systems. Information and Software Technologies: Proceedings of 18th International Conference (ICIST 2012), Communications in Computer and Information Science, Vol. 319, Part 4 (2012), pp. 187-198, Springer-Verlag Berlin Heidelberg, ISSN 1865-0929.

In proceedings of other conferences:

1. Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers. Proceedings of the 15th Conference for Master and PhD students "Information Society and University studies" (2010), Kaunas, Lithuania, pp. 27-32, ISSN 2029-4824.

2. Danenas P., Garsva G. A model for multidimensional analysis for credit risk evaluation based on intelligent techniques (in Lithuanian). Conference Proceedings of "Information Technology", 16th Conference for Master and PhD students (2011), Kaunas, Lithuania, pp. 49-52, ISSN 2029-249X

3. Galkus E., Danenas P., Garsva G. Application of ensemble classification methods in credit risk evaluation (in Lithuanian). Conference Proceedings of "Information Technology 2012", 17th Conference for Master and PhD students (2012), Kaunas, Lithuania, pp. 70-73, ISSN 2029-249X.

1.9. Thesis structure

The thesis contains the introduction (including list of the author's publications), 4 chapters, conclusions (6 chapters in total), list of references and 14

appendixes. The total volume of the dissertation is 234 pages, including 34 tables, 38 pictures and 8 algorithms. The list of references contains 259 various sources, including books, scientific papers, technical reports, Internet sources.

The work consists of seven main parts – introduction, analytical part, methodological part, experimental part, implementation part, future work and conclusions. Each of these parts discusses particular aspects of research and decision support system which is discussed, designed and developed.

- *Introduction (first chapter)* presents research problem, object, aims and objectives, research findings and results, its scientific significance, describes possible practical application, presents information about the papers, in which the main results of the research were published.
- *Analytical part (second chapter)* discusses main concepts and definitions of artificial intelligence, problems that it can solve, main techniques from this field such as artificial neural networks, decision trees, evolutionary and swarm intelligence techniques and etc. with their previous research in credit risk domain as well as various statistical and intelligent feature selection techniques. Support Vector Machines, as technique discussed in this work, is described in more detail in separate section. Main concepts from credit risk domain such as various kinds of other risks, sample ratios, taxonomies of ratings, evaluation techniques are also shortly described. Finally, financial standards, their structure and adoption for financial reporting and evaluation, together with intelligent decision support systems, their taxonomies, main components and use cases for financial decision support are discussed.
- *Methodology part (third chapter)* describes developed techniques and tools for evaluation such as machine learning metrics. It gives generic framework for hybrid model development which generalizes previously made researches and proposes a generic structure for further classifier research. Implemented classification approaches (SVM with feature selection and its extension for sliding window testing based approach), together with PSO-LinSVM and GA-LinSVM classifiers, combining correspondingly Particle Swarm Optimization and Genetic Algorithm techniques with based linear SVM classifier selection from a family of similar classifiers together with its parameters are also described.
- *Experimental part (fourth chapter)* describes experimental research made with developed techniques, presents obtained results and their analysis.

- *Implementation part (fifth chapter)* describes functionality for intelligent decision support system, describes framework for its development with integration of financial standards, together with its design and development methodology, possible implementation scenario and development possibilities. It also describes developed prototype, its current capabilities and future development.
- *Conclusions part (seventh chapter)* presents final conclusions of the dissertation.

At the end of the dissertation a list references and appendixes is given.

2. A REVIEW OF EXISTING TECHNIQUES AND PROBLEM DOMAIN

The concept of credit risk is used often, but despite the fact that it is widely described and researched in many papers, books and other sources, it is still permanently researched, with many new techniques and approaches being constantly developed. This problem can be viewed as a multidimensional problem related to solution of several related topics:

- Problem identification;
- Research and evaluation of related risks such as operational, financial, liquidity, market risks;
- Data collection, pre-processing and analysis;
- Identification of risk significant factors;
- Selection of techniques or their development and implementation in evaluation system;
- Modelling using developed model and available data;
- Analysis of results obtained during modelling process, their interpretation and further use in optimization of credit risk management and model development.
- Implementation of developed model in expert credit risk evaluation system or financial environment.

This part of work presents a short review of existing techniques and methods for solution of each of these problems, discusses current implementations or possible ways to implement them.

2.1. Machine learning techniques

2.1.1. Artificial intelligence, machine learning and data mining – main concepts

The use of information technology allowed automating most complex and computationally demanding decision processes, as well as performing analysis and prediction processes faster and more effectively. Many researchers currently focus on intelligent techniques to solve classification and forecasting tasks as combination of these techniques proved to be in more efficient in various fields, as well as particularly on credit risk evaluation solutions [63].

The last decade of 20^{th} age saw a boost of wide development and adoption of

techniques in artificial intelligence field. This field has been discussed in various sources since 6th decade [85]. This topic soon became more important and interesting to researchers than it was considered before as it offered a possibility to use new kinds of efficient techniques which offered capabilities to imitate human thinking, as well as formalize and integrate external knowledge provided by experts. The concept of artificial intelligence describes many aspects thus there are many definitions. Russell and Norwig [185] categorize such definitions according to their abilities to think and act rationally or imitate human thinking. These definitions express main actions taken by the systems (thinking and acting) as well as characteristics (rationality and human imitation). Systems which only think can be viewed as tools which help to make decision, while acting systems also take actions triggered by such decisions. Therefore, development of such systems can be viewed both as imitation of human thinking, as well as improving it with stability of decision support that can be violated by emotional factors arising from human nature. Rationality which defines decision consistency in compliance with defined constraints and rules can be viewed as one of such characteristics.

Machine learning paradigm is closely related to artificial intelligence field and is often referred as one of its subfields, targeted at numerical imitation of thinking and behaviour of human, nature, living species and their groups, which comes to optimal solutions. *Data mining* (also often referred as or used in context of *knowledge discovery*) can be considered as separate scientific field which extends possibilities offered by statistics with machine learning algorithms as well as philosophy of their development and application. It is also concerned with tasks of obtaining data, its preprocessing, imputation, analysis and storage; CRISP-DM (Cross Industry Standard Process for Data Mining) methodology [38] provides guidelines to solve problems using data mining techniques. The evolution of AI field was driven by development of new statistical, data mining, information retrieval and machine learning techniques, as well as business intelligence and database technologies. The main steps of data mining, statistics, AI and expert system evolution are described in various sources [73, 85].

Machine learning can be described as process targeted at learning on examples and obtaining the most generalizing structure. According to statistical learning theory described in Vapnik and Chervonenkis [225], it can be described as

interaction between generator, target operator and learning machine components, with a goal to select the most suitable function from a given set of functions. The main task is to obtain and imitate learning operator which can produce best prediction results for the output of given generator. Generator describes the context for learning machine and learning operator; this is usually input vector and probability distribution function *f(x)*. Target operator transforms input vectors *x* to output vectors *y*, using distribution function *F(y|x)*. Learning machine can be described as an operator that generates a set of functionals $g(\mathbf{x}, \alpha), \alpha \in \Lambda$ according to given independent learning data $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ identically distributed by joint distribution function $F(x,y) = F(y|x)F(x)$ and predicts an answer of the target operator y_i for each vector x_i. This is done by approximating by unknown operator or choosing an optimal function from the set of functionals.

According to [225] machine learning task can be formalized as minimization of functional $R \cdot = R(g(z)), z \in Z, Z \subset R^n$ which is defined as criteria for quality of chosen function, with an objective to find function *g*(z)* from set *{g(z)}* that minimizes *R*. This functional can be defined as

$$R(g(z)) = \int L(z, g(z))dF(z) \tag{2.1}$$

where *F(z):* $z \in Z, Z \subset R^n$ is a given probability distribution function and *L(z, g(z))* is integrated with each $g(z) \in \{g(z)\}$.The search of fitting function and problem of obtaining minima of the functional in the given set of functions are the biggest challenges, thus constructive criteria for function selection can be preferred rather than searching for this function in *{g(z)}*. Thus given a set of functions *g(z)* in form $\{g(z,\alpha)\}, \alpha \in \Lambda$ such that Λ is scalar, vector data or abstractions, the goal is to find parameter $\alpha \in \Lambda$. Given functional can be rewritten as [225]

$$R(\alpha) = \int Q(z,\alpha)dF(z), \alpha \in \Lambda \tag{2.2}$$

where $Q(z,\alpha) = L(z, g(z,\alpha))$ is referred as *loss function*. Then possible loss or precision (*risk functional* or *risk*) is [225]

$$R(\alpha^*) = \int Q^*(z,\alpha)dF(z), \alpha \in \Lambda \tag{2.3}$$

Minimization of this functional is solved two possible methods of risk estimation. Most widely applied are *empirical risk minimization* techniques, when empirical risk functional is minimized with function representing *learning error*; for e.g., *maximum likelihood method*. *Structural risk minimization* (SRM) induction principle proposed by Vapnik in [225] seeks to minimize risk for separate subsets of

data and obtain subset with optimum (smallest) risk margin. Given a nested set of structures $S_1 \subset S_2 \subset ... \subset S_n$ and sets of functions $Q\{g(z,\alpha)\}, \alpha \in \Lambda$ such that Λ is scalar, for each structure, the SRM method chooses the element S_k of the structure for which the smallest bound on the risk (the smallest guaranteed risk) is achieved for a given dataset D. Thus the main idea of SRM is provide the given set of functions with an admissible structure and then finding the function that minimizes guaranteed risk (6.8) (or (6.9)) over given elements of the structure [225]. Another important concept of statistical learning theory is Vapnik-Chervonenkis dimension or VC dimension, defined for a set of functions $Q=\{g(z,\alpha)\}, \alpha \in \Lambda$, which describes capacity of Q and is equal to the largest number h of vectors in set D that can be separated into two different classes in all the 2^h possible ways using this set of functions [225].

Pattern recognition (identification of connection for monitored instances to one of k classes) is one of the biggest subsets of problems which are solved by machine learning. The decision rule is formulated as $F(\omega \mid x), \omega \in \{0,1,..., k-1\}$ and the problem is defined as minimization of functional $R(\alpha)$ with a $n+1$ dimensional vector of known random independent pairs of instances $z = \{(\omega_i, x_i), i \in \boldsymbol{Z}\}$, where coordinates ω have meaning only with finite set of values and coordinates $x^1, x^2, ..., x^n$ of vector x. *Classification* (prediction of the class of an unseen input vector), *pattern matching* (producing a pattern best associated with a given input vector) and *control* (suggesting an appropriate action using a given vector tasks) can be referred as relative problems. Dunham formulates classification problem as following [73]: given a database $D = \{t_1, t_2, ..., t_n\}$ and a set of classes $C = \{C_1, C_2, ..., C_n\}$ the classification problem is a mapping $f : D \to C$, where each t_i is assigned to one of the classes from set C. $C_j = \{t_i \mid f(t_i) = C_j, 1 \le i \le n, t_i \in D\}$, i.e., class C_j contains only entries assigned with it.

2.1.2. Feature selection techniques

Feature selection techniques are usually applied in initial phases of model development process. Such techniques allow selecting statistically important subset of independent variables or extract basic components from the whole set. It reduces the number of dimensions in the data needed for model development thus speeding model training process, reducing the complexity of the model and often providing better results in terms of accuracy and variable dependency. Currently a large number

of feature selection and extraction methods ar available; most popular of them are:

- *Correlation based* – correlation coefficient is used as a measurement to identify level of relation between two quantitative variables (i.e., if changes or movements in both of these variables occur at the same time). This measurement can be defined as ratio of covariance between two random variables *X* and *Y* and product of their standard deviations [155]:

$$\rho_{X,Y} = corr(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{2.4}$$

while sample correlation coefficient for $X = \{x_i \mid 1 \le i \le n, i \in N\}$ and $Y = \{y_i \mid 1 \le i \le n, i \in N\}$ is defined as [155]

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{(n-1)\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} = \frac{n\sum_{i=1}^{n}x_i y_i - \sum_{i=1}^{n}x_i \sum_{i=1}^{n}y_i}{\sqrt{\sum_{i=1}^{n}x_i^2 - (\sum_{i=1}^{n}x_i)^2}\sqrt{\sum_{i=1}^{n}y_i^2 - (\sum_{i=1}^{n}y_i)^2}} \tag{2.5}$$

where x and y are average value of samples for *X* and *Y*. Hall proposed a CFS (*Correlation-based Feature Selection*) technique based on (2.5) which applies heuristic measure of the "merit" of a feature subset from pair-wise feature correlations for both classification and regression problems [96, 97]. This technique is preferable of its low computational demand – according to Hall, CFS requires $m((n^2 - n)/2)$ operations for computing the pairwise feature correlation matrix, where *m* is the number of instances and *n* is the initial number of features, and the feature selection search requires $(n^2 - n)/2$ operations (worst case) for a forward selection or backward elimination search [96]. Possibility to use stopping criterion also can reduce probability of exploring the entire search space [96].

- *t-test* method is a statistical technique used to determine whether there is a significant difference between two group's means. It helps to identify if the two groups come from the same population, or if these two groups have statistically significant difference [105].

- *Factor analysis* – technique for data projection to smaller dimensional space based on extraction of components (common factors CF_X and unique factors UF_X) from dataset D with feature vector X_D such that $\dim(CF_X) < \dim(X_D)$, $\dim(UF_X) < \dim(X_D)$ and $|CF_X|$ is minimized. Unique factors are not related to common factors and to other unique factors while common factors are components which describe common variance (correlation) shared between variables in the set. Of such factors

selected, the first factor describes the most common variance between variables and the second factor explains the most variance after eliminating the first factor [123, cited by 105].

- *Principal component analysis* (*PCA*, also called the *Karhunen- Loeve*, or K-L method) – similarly to factor analysis, the main objective is to extract $c < dim(X_D)$ components (a set of principal components) from dataset D with feature vector X_D that best represent the data used, are uncorrelated and ordered so that the first few retain most of the variation present in the entire original variables [99, 105]. Similarly to factor analysis, the first principal component describes as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible [105]. Several authors (Yang et al. [248], Min [165]) reported that application of PCA in classification tasks for credit risk domain resulted in increased accuracy.

- *Wavelet analysis* - the discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector D, transforms it to a numerically different vector D' of wavelet coefficients with the same length as D. Wavelet transforms can be applied to multidimensional data by first applying the transform to the first dimension, then to the second, and so on. They give good results on sparse or skewed data and on data with ordered attributes [99].

- *Exhaustive search* – one of the most simple yet expensive techniques because of its exponential complexity – for each dataset with d dimensions there are 2^d possible combinations; therefore, if in case of binary classification oriented classifiers which have to employ some multiclass extensions such as *one-vs-one* or *one-vs-all* (described in Section 2.2.3), it becomes even larger.

- *Stepwise forward selection* – heuristic technique, which is starts with an empty set of attributes. The best of the original attributes is determined and added to the set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set [99].

- *Stepwise backward elimination* – conversely from stepwise forward technique, it starts with full set of attributes and at each step the worst attribute remaining in the set is removed [99].

- *Combination of forward selection and backward elimination* - at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes [99].
- *Support Vector Machine - Recursive Feature Elimination* (SVM-RFE) – this feature selection technique is based on training a support vector machine using training samples with class labels to determine a value of each feature, where features are removed based on their the value. One or more features having the smallest values are removed and an updated kernel matrix is generated using the remaining features. The process is repeated until a predetermined number of features remain which are capable of accurately separating the data into different classes [216].
- *Metaheuristic techniques* –techniques based on evolutionary, swarm intelligence and other soft computing techniques. They are discussed in Section 2.1.10.

Tsai [105] performed a comparative analysis of some of these techniques in bankruptcy prediction, using correlation matrix, t-test, factor analysis, principal component analysis and stepwise regression with German, Japanese, Australian credit datasets. He concluded that t-test is superior to others and is more stable than stepwise forward selection. Although stepwise selection gave highest feature reduction rate, it was not as stable as t-test and several other techniques.

Many authors used feature selection as one of their steps in their proposed techniques for credit risk evaluation. Kim et al. [130] used stepwise discriminant analysis method combined with F-score to remove insignificant variables; Min et al. [164] used stepwise logistic regression for the same task. Ping et al. [173] compared t-test, correlation, stepwise, CART, MARS, rough set and neighborhood rough set approaches for feature selection for German and Australian credit datasets and concluded that rough sets approaches outperformed statistical techniques. Wang et al. [231], Wang et al. [232], Zhou et al. [263], Zhou et al. [262] also applied rough sets for feature selection. Supervised feature extraction techniques such as PLS (Yang et al. [251]) and genetic algorithm (Zhang et al. [259]) also proved to be efficient solution. Other authors used SVM-RFE (Belotti et al. [23]), genetic algorithm combined with SVM (Huang et al. [113]) or SVM with mixture of kernel (Wei et al. [235]) for feature selection. Yun et al. [256] combined both feature and parameter selection into PSO-based approach. Wang [234] also showed that PSO based feature

selection resulted in more efficient classification compared stepwise discriminant analysis, stepwise linear regression and t-test techniques. Therefore, it is difficult to exclude technique which guarantees best performance; thus correlation-based feature selection proposed by Hall [96] is chosen for the experimental part in this work.

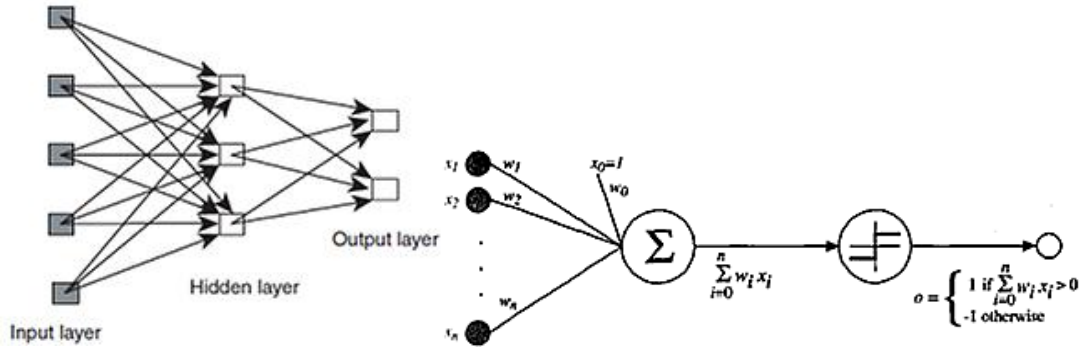
2.1.3. Artificial neural networks

Artificial neural networks (ANN) is one of the most widely discussed, developed and researched biological learning driven machine learning techniques, based on building of very complex webs of interconnected neurons and capable to imitate performance of other system. This is a computing architecture modelled using simplified neural system in human brain that imitates the capabilities of the human brain to recognize, identify, adapt and learn from patterns in the past, process information simultaneously[12, 163, 166]. Thus ANNs are built out of a densely interconnected set of neurons, which take a number of real-valued inputs (initial training data or the outputs of other neurons) and produce a single output. The latter can be used as the input to other neurons or learning units. This technique can be parallelized, enabling implementation of algorithms on highly parallel machines or specialized hardware. Dunham [73] defines ANN as oriented graph and as a computational model. As an oriented graph, ANNs contain many processing elements and arcs (connections) between them; each of these elements functions independently from others and uses only local input and output data for his runtime management, which allows their application in distributed and/or parallel environment. Thus neural network can be described as oriented graph $F = V, A$ having set of vertices $V=\{1,2,...,n\}$ and a set of arcs $A = \langle i, j \rangle | 1 \le i, j \le n$ and implying such constraints [73]:

1. V is divided to a set of input elements V_I, a set of hidden elements V_H and a set of output elements V_O;
2. Vertices can also be divided to layers $\{1,..,k\}$ with all input nodes in layer 1 and output nodes in layer k. All hidden nodes are in layers from 2 to $k - 1$, called hidden layers;
3. Each arc $\langle i, j \rangle$ must have node i in layer $h - 1$ and node j in layer h;
4. Arc $\langle i, j \rangle$ has a real value w_{ij};
5. Node i is marked as function f_i.

As an algorithm ANN can be defined as a computational model with three elements:

1. Neural network graph which describes data structure for neural network
2. Learning algorithm describing network learning;
3. Memorizing techniques which describe how information is obtained from the network.



Source: T. Mitchell. Machine learning.

Figure 1. Examples of architecture of feed forward multilayer perceptron and perceptron

Engelbrecht [76] also gives a formal definition of ANN: a neural network is basically a realization of a nonlinear mapping from R^I to R^K, $f_{NN} : R^I \rightarrow R^K$, where I and K are respectively the dimensions of the input and target (desired output) space. The function f_{NN} is usually a complex function of a set of nonlinear functions, one for each neuron in the network. Thus, according to the definitions given, neural network can be defined in terms of graph theory, using set of components which are necessary in its performance or terms of mappings to different data spaces.

Neural network training depends on algorithms used, number of neurons, number of layers. Two terms – neurodynamics and architecture – are used to describe the configuration of neural network. Neurodynamics describes the features of each neuron as a unit, such as transfer function and how are input data joined. The architecture of neural network describes its structure, such as the number of neurons in each layer and number of inner connections. Figure 1 gives an illustration of architectures for single and multilayer perceptrons.

ANN architecture describes such characteristics for neural network learning;

- *Number of hidden layers.* Hidden layers enable generalization of data. It might best choice to use ANN with one or two hidden layers. Cybenko [50, cited in

166] proved that it is possible to approximate any function using ANN with two hidden layers.

- *Number of hidden neurons***.** Although this is an important factor, the number of hidden neurons can be obtained only experimentally. Selecting the number of hidden neurons too small might result in underapproximated target function, yet if the number is too large it can cause overfitting problems [73].

- *Number of output neurons.* The task to identify the umber of outpu neurons required is much easier as usually only a single neuron is used in case of regression; in case of classification it is usually presumed that the number of neurons for output is the same as the number of classes although this is not necessarily correct for all cases [73].

- *Transfer (transformation, activation) functions*. These are mathematical formulas which describe the output of processing neuron; they can be any linear or nonlinear functions.

- *Initial weights*. They are usually initialized as random, although usage of expert defined weights can result in much faster training and obtaining optimum weights.

The objective of training process is obtaining a set of weights for neurons to minimize error function. However, one of the main problems of ANN is related to approximation of function or hyperplane which has many local minimas as ANN training can get stuck in these minimas.

Mitchell gives such characteristics for problems which can be solved using ANNs (note that the same characteristics can be applied for problems solved by SVM) [166]:

- Instances are represented by vectors consisting of a vector of predefined features (input values), that may be represented as real values and be highly correlated or independent of one another, and output values, which can be real-valued (to solve regression tasks) or nominal (for classification).

- The target function output may be discrete-valued, real-valued, or a vector of several real- or discrete-valued attributes.

- Robustness to noise in the training data.

- Long training times are acceptable. ANNs are trained slower than various other machine learning techniques based models, although this is highly dependent on

the architecture of ANN, learning algorithm parameters or the length of training vector.

- Fast evaluation of the learned target function may be required. Although ANN training might be relatively long, applying it to a subsequent instance is typically very fast.
- The ability of humans to understand the learned target function is not important. The weights learned by neural networks are often difficult for humans to interpret.

Many types of multilayer neural networks have been developed. More details of their implementations, algorithms for network training and weight updating a can be found in various sources, such as [73, 78, 166, 228]. The most important types of ANN which should be noted are single perceptron, multilayer perceptron (MLP), radial basis function network (RBF), feedforward neural networks (FFNN) which do not send any information back to nodes in previous layers and are processed from left to right, backpropagation neural networks (BPNN) which structure corresponds to a directed graph, possibly containing cycles [166], functional link neural networks (FLNN), simple recurrent neural networks (SRNN) which extend FFNN with feedback connections that enable to learn the temporal characteristics of the data set and cascading NN (CNN), a multilayer FFNN where all input units have direct connections to all hidden and to all output units [76]. Self-organizing maps (SOM), also known as Kohonen maps - a multidimensional scaling method to project an n-dimensional input space to k-dimensional output space where $n > k$ (often k =2) by performing a compression of input space onto a set of codebook vectors. The SOM tries to keep the topological structure of input space [78]. Yet, if two vectors are close to one another in input space, they are represented in the map in such way as well.

According to Vellido et al. [227], back propagation gradient descent was the most popular technique for training among researchers in credit risk and business field in 1992-1998; self-organizing maps were also widely used for clustering-targeted tasks. A taxonomy of neural network architectures used in this periods for research can also be found in his paper. This survey also proves that complicated access to credit risk data limits possibilities to compare the results of different researchers – only several authors (Richeson, Zimmerman, & Barnett, 1994; Williamson, 1995; Jagielska & Jaworski, 1996; Desay & Crook, 1996; Torsun, 1996;

Glorfeld & Hardgrave, 1996; cited by [227]) have worked with real data, which comprised loan data from 40 to 310000 instances. These works used from 6 to 27 variables, yet only several authors (Williamson, 1995; Glorfeld, 1996; Glorfeld & Hardgrave, 1996; cited by [227]) used feature selection techniques.Yet only (Glorfeld 1996; Glorfeld & Hardgrave. 1996; Desay & Crook, 1996, cited by [227]) used cross validation techniques. Also as one of the drawbacks Vellido et al. distinguish the lack of research for different credit risk fields, such as sales credit risk.

Wong et al. [238] also reported that backpropagation algorithm was used mostly for research in business domain in 1994-1998. Another survey of neural network based research by Wong [239] is targeted at financial domain and includes survey of works targeted at bankruptcy prediction of firms and banks or credit risk evaluation; however, only 20 related researches are mentioned in his survey. This shows that ANNs are widely researched and applied in financial and credit risk domain for more than 20 years, especially for classification tasks. More recent research proposes a lot of hybrid models developed on basis of ANN, such as fuzzy neural networks with particle swarm optimization for parameter selection [114], wavelet neural networks with Gaussian wavelet function and differential evolution applied for their training [40], knowledge-based artificial neural network (KBANN) with rule extraction from trained neural networks [16], neurofuzzy systems [43] and other. Ensemble ANN models are also a widely developed technique [140, 213]. These papers show that neural network application for credit risk related problems offers a possibility to obtain better results and develop more complex models compared to other statistical and machine learning techniques.

Self-organizing maps were also applied in credit risk domain [221, 162] – to analyse financial reports and bankruptcy prediction [126], as well as formation of credit classes and prediction [161, 162]. Deboeck showed more financial and economical fields which can make use of this techniques such as financial analysis and prediction, ranking of financial instruments, disaster and failure prediction, investment analysis, analysis of credit risk for commercial and governmental levels, financial monitoring, analysis of economic trends, marketing, user rankings and etc. [69].

2.1.4. Support Vector Machines

Support Vector Machines (abbr. as SVM) is a machine learning technique

similar to Neural Networks, developed in 7th decade by Russian scientist V. Vapnik. In fact, it might be viewed as a universal feedforward multilayer perceptron [228]. The main task is finding an optimal hyperplane. As this work concentrates mostly on research of SVM based techniques, this technique is later discussed in more details in Section 2.2.

2.1.5. Decision trees

Decision trees are one of the oldest and most widely applied machine learning techniques. The learned pattern can be described as a decision tree or a set of *if* ... *then* rules forming a tree-based structure. Mitchell defines decision tree learning as "sorting the instances down the tree from the root to some leaf node, specifying one of the attributes, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute" [166]. According to Mitchell's definition, decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances [166]. This algorithm can be applied in both of the cases, when the attributes are either nominal or numerical; it can also perform well when training data contains missing or erroneous values, although it suffers from large adaptability to training data which results in worse generalization. ID3, C4.5, CART techniques are widely used and implemented in most popular statistical and data mining packages such as SAS, SPSS, Statistica. They are also often applied as benchmarking techniques in various papers which describe new hybrid methods based on SVM [44, 90, 233, 235, 252, 265].

2.1.6. Fuzzy logic

Fuzzy sets are an extension of crisp (two-valued) sets to handle the concept of partial truth, which enables the modeling of the uncertainties of natural language [1]. In contrast to Boolean logic fuzzy logic enables expressions which are not evaluated strictly to *true* of *false*, but also have partial degree of truth expressed in probabilities, i.e., $p \in [0;1]$ is used instead of $p \in \{0;1\}$. Table 1 gives an example of fuzzy rules for credit risk evaluation based in indicators used in widely used Altman Z-Score model. This is more consistent to human reasoning which uses both logical and statistical reasoning thus it can be used to integrate expert knowledge. Fuzzy sets enable computing systems to understand linguistic terms that express ambiguity and to

reason with these terms in computationally inexpensive way. Most fuzzy systems enable more than one implication for each rule. A set of rules is referred as knowledge base, the process which applies functions for input variables to obtain output values – inference (also referred as deduction), which consists of 4 main subprocesses: fuzzification, inference, composition and defuzzification.

Table 1. Probability of default analysis using fuzzy logic rules

Rule 1	If	EBIT/Total assets	is	Large
		Retained earnings	is	Small
	Then	Probability default	is	Small
Rule 2	If	Retained earnings	is	Small
		EBIT/Total assets	is	Small
	Then	Probability default	is	Large
Rule 3	If	Retained earnings	is	Large
		EBIT/Total assets	is	Small
	Then	Probability default	is	Average
Rule 4	If	Retained earnings	is	Large
		EBIT/Total assets	is	Large
	Then	Probability default	is	Very small

Fuzzy logic is rarely used alone but one can find a lot of examples when fuzzy logic is combined with neural networks (neurofuzzy systems) [152,174] or SVM as fuzzy-SVM [39,49,100,101,266] for credit risk evaluation. Neurofuzzy systems use perceptron of multilayer ANNs with different weights and transition functions to obtain fuzzy rules and sets. By using fuzzy logic, it is possible to check all possible connections between variables and identify significant factors more effectively. Development of such systems has similar problems as in case of canonical ANN – choice of architecture, mainly membership functions, fuzzy set connections and operators for fuzzification, composition and defuzzification. Fuzzy SVM combine fuzzy logic and SVM in a similar way yet they need less architecture parameters which should be set.

The survey of SVM–based methods showed that fuzzy logic integration helped to achieve better results than using techniques without it [63].

2.1.7. Case-based reasoning

Case-based reasoning (CBR) is a learning paradigm based on two principles: deferring the decision of how to generalize beyond the training data until a new query instance is observed and classifying new query instances by analysing similar instances while ignoring instances that are very different from the query [166]. This is

very similar to the principle of human decision making. This approach has also been used in credit risk and bankruptcy evaluation field [4,71].

2.1.8. Rough sets

Rough set theory was developed by Z. Pawlak in 1982 as a technique of approximate reasoning. It is based on the assumption that objects characterized by similar information are indistinguishable or indiscernible. The indiscernibility relation indicates that we are unable to deal with single objects but we have to consider clusters of indiscernible objects or equivalence classes of the indiscernibility relation. In rough set theory, a pair of precise concepts – the lower and the upper approximations replace any vague concept [78]. It can be used for classification or to get a set of rules from data. It is also widely used in finance, particularly in credit risk, especially in hybrid models [150,263] as well as in combination with SVM [173,231,232].

2.1.9. Bayesian methods

This is one of the most popular and widely applied classifiers. Bayesian reasoning provides a probabilistic approach to inference. It is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data [166]. There are various modifications of this classifier; Naïve Bayes and Bayesian Belief Networks are most widely known and researched. They were also successfully applied in financial domain, particularly in credit risk [2,17,204] Bayesian based Relevance Vector Machines (RVM) kernel technique has been introduced by Tipping which is identical in the form to SVM but uses Bayes inference for classification and regression [212]. It has been also applied in financial and credit risk domains [145,181,182].

2.1.10. Heuristic optimization techniques

Genetic algorithms and other natural computing based techniques currently are one of the most promising and effective heuristic techniques applied to solve various tasks. Good heuristics is essential in solving various problems which deal with finding an optimal solution in large state spaces as exhaustive search in these spaces is generally infeasible. Heuristics help to estimate of the remaining distance from a particular state to the final goal. Many techniques are suggested to solve such

problems; for an extensive reference of such techniques refer to [75, 76]. This section reviews only genetic algorithms, simulated annealing, tabu search and swarm intelligence based methods, which are often used combined with other machine learning techniques described in this work.

2.1.10.1. Genetic algorithm

Genetic algorithms (GA) enable to solve problems by obtaining the solution in evolving manner. Optimization techniques known as predecessors for modern GA were used to solve real problems by Rechenberg and Schwefel in 8th decade [180,193], although these technique became popular after early works of Holland, and particularly his book *Adaptation in Natural and Artificial Systems*, released in 1975. GA is based on the structure and evolution of cells found in living organisms. These cells consist of a set of chromosomes (genome), which consists of a set of genes (genotype). Each gene has its own position in the chromosome (known as locus) and contains information about sam particular characteristics, such as eye color and etc. Recombination (or crossover) occurs during reproduction, when genes of parent objects form a new chromosome. Then newly created chromosome mutates as a consequence of DNR elements' change due to improper copy of parents' genes (the chromosome evolves). During execution of genetic algorithm the decision searching process generates other possible solutions (new points in search space) by evolving itself. Instead of optimal solution, often only best possible solution might be available, as it would be difficult to decide which solution is optimal.

Initial input for genetic algorithm is the set of solutions represented by chromosomes, called population. Solutions from one population are used to form new, "better" population. Solutions for new population are selected according to their fitness – the more fit they are, the more chances they have to reproduce. It is believed that the better fitness of parent chromosomes, the bigger possibility they have to generate "better" offspring. Thus the average fitness for further populations should increase, since only the best organisms from the previous generation are selected for breeding, along with a small proportion of less fit solutions.

According to Dunham, genetic algorithm can be described as following [73]:

- Given alphabet A, a chromosome or individual is a string $I = I_1,I_2,...., I_n,$, where $I_j \in A$. Each symbol of the string I_j is called a gene. Population P is a set of individuals, randomly initialized using operation *Init(p).*

- Operations *Crossover(P_i)* and *Mutate(P_i)* are defined for Each individual $P_i \in P$. Fitness function *fitness(P_i)* is a mapping $f : P \to R$.

Therefore GA can be defined as computational model *GA = <Init(P); Crossover(P_i); Mutate(P_i); f; A=(P, Crossover(P_i), Mutate(P_i), f)>, i=1,..,SP*, here *SP* – size of population *P*, *A* – algorithm which iteratively uses crossover and mutation techniques in set *P* and fitness function *f* to identify best remaining individuals, which changes predefined number of individuals in each iteration and ends execution after particular threshold is reached or an optimal solution is obtained.

Generic model of GA is described in Algorithm 1. There are three key GA parameters:

1. *Crossover probability $Pr(h_i)$,* which describes how often the crossover is done, $Pr(h_i) \in [0;1]$. If *$Pr(h_i)$* = 1, then the population consists of all new offsprings formed during crossover; if *$Pr(h_i)$* = 0, then all new population is formed from exact copies of chromosomes from old population (although it does not mean that new population itself is the same as old population).

2. *Mutation probability p_M,* which describes, how often parts of chromosome mutate, $p_M \in [0;1]$. If there is no mutations (p_M = 0), then nothing is changed in the chromosome, if p_M = 1, then whole chromosome is changed. Mutation should not happen too often as it might become random search.

GA(Fitness, Fitness_threshold, p, r, m)

Input: *Fitness: A function that assigns an evaluation score, given a hypothesis.*
Fitness_threshold: A threshold specifying the termination criterion.
p: The number of hypotheses to be included in the population.
r: The fraction of the population to be replaced by Crossover at each step.
m: The mutation rate.

1. Initialize population: $P \leftarrow Init(p)$
2. Evaluate: $\forall h \in P : \leftarrow Fitness(h)$

while $\max_h(Fitness(h)) < Fitness\_threshold$

Create a new generation, P_s:

1. Select: Probabilistically select *(1 - r)p* members of P to add to P_s. The probability *$Pr(h_i)$* of selecting solution h_i from *P* is given by

$$Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

2. Crossover: Probabilistically select *(r-p)/2* pairs of solutions from *P* to set *k*, according to *$Pr(h_i)$*.

Generate offsprings using crossover:
$\forall (h_1, h_2) \in k : h_1' \rightarrow Crossover(h_1'), h_2' \rightarrow Crossover(h_2')$ Add all offspring to P_S.

3. Mutate: $k \leftarrow m$ percent of the members of P_S with uniform probability. $\forall h \in k : h \rightarrow Mutate(h)$

4. Update: $P \leftarrow P_S$.

5. Evaluate: $\forall h \in P_S : h \leftarrow Fitness(h)$

Output: The solution from P that has the highest fitness.

Source: adopted and modified from Mitchell T. Machine learning.

Algorithm 1. Generic scheme of genetic algorithm

3. *Population size*, which influences search space size and possibility of crossover. Population size which is too large might heavily slow down performance of GA, also it is experimentally shown that usage of large population is not an effective solution.

Various modifications of genetic algorithms can be identified; Haupt and Haupt [102] describe binary, continuous, hybrid, messy, parallel genetic algorithms, together with their advantages and disadvantages. Binary algorithm (most often referred) uses bitstring encoding, thus additional procedure of encoding/decoding is needed; it is also not suitable for continuous variables. These disadvantages are eliminated in continuous GA; this version of genetic algorithm is also more suitable in further research presented in this work. Parallel technique of GA can be used when complex fitness functions with complex and high performance computations have to be used. Several types are described in various literature, e.g. Cantú-Paz describes global single-population master-slave GA, single-population fine-grained GA and multiple-population coarse-grained GA algorithms which enable to use parallel computing infrastructure and obtain several solutions, better computing speed and performance [32]. Usage of subpopulations in different machines enables testing of different combinantions thus avoiding domination of one or several individuals. Complex implementation and need for technical resources can be marked as main disadvantages.

According to various authors, such as [102], such advantages of GA can be identified: ability to solve nonlinear and nonflexible problems, keeping non-optimal and unfinished solutions, self-optimization, conceptual simplicity (although resulting in complex GA configuration and implementation).

Although genetic algorithms are usually applied in optimization and

parameter selection for classification models, Schlotmann and Seese applied them for improvement of structural and modern credit risk models, such as CreditRisk+ [189, 190] while Barthelemy and Apoteker used it to develop a vulnerability indicator to analyse financial crises [19].

2.1.10.2. Simulated annealing

Simulated annealing (SA) method simulates the annealing process in which a substance is heated above its melting temperature and then gradually cooled to produce the crystalline lattice, which minimizes its energy probability distribution thus finding its optimal structure. Cooling too quickly or quenching the liquid retards the crystal formation, and the substance becomes an amorphous mass with a higher than optimum energy state. Thus forming a crystal requires to carefully control the rate of change of temperature [102]. Technically, the rising temperature means random change of variable values; the more the temperature is changed, the bigger are random fluctuations. The decrease in temperature is known as the cooling schedule; different cooling schedules might be used, such as linear, geometrical or Hayjek optimal decrease [102].

Procedure SimulatedAnnealing
Input: State space min. problem, initial temperature T

$t \leftarrow 0$;	*(Iteration counter)*
$u \leftarrow s$;	*(Start search from initial state)*
while ($T > \varepsilon$)	*(T not too close to 0)*
Succ(u)←Expand(u)	*(Generate successors)*
v←Select(rand(Succ(u)))	*(Choose random successor)*
if ($f(v) < f(u)$)	
$u \leftarrow v$	*(Evaluation improved, select v)*
else	*(Evaluation worse)*
r←Select(rand(0,1))	*(Choose random probability)*
if ($r < e^{\frac{f(u)-f(v)}{T}}$)	*(Check Boltzmann condition)*
$v \leftarrow u$	*(Continue search at v)*
$t \leftarrow t+1$	*(Evaluation improved, select v)*
T←Cooling(T, t)	*(Decrease T according to iteration count)*
return u	*(Output solution)*

Output: State with low evaluation (optimal result) u

Source: S. Edelkamp, S. Schroedl. Heuristic Search Theory and Applications

Algorithm 2. Simulated annealing algorithm

The algorithm of SA is shown in Algorithm 2. After cost function variable values are randomly initialized, their values are randomly modified (analogy to heating process). If the output of the cost function decreases, the set of variables, associated with it, replaces the old variable set. If cost function value increases, then the new set is accepted with a certain probability and a random step is taken to obtain the new variable set. At the beginning of the process, the algorithm is forced to make large changes in variable values. Sometimes such change force to move away from the optimum, thus enabling searching in new parts of variable space. After a certain number of iterations, the new variable sets no longer lead to lower costs. The algorithm stops when *T* is near or equal to 0. This technique is known to show good performance and obtain good results in relatively small number of iterations, therefore the global optimum can be reached soon. It also performs considerably better with multimodal cost functions than local optimizers [75]

2.1.10.3. Tabu search

Tabu search, similarly to simulated annealing, moves over through all possible solution space by checking all "neighbours" of current solution; however, the neighbours which should not be checked are marked as "tabu" are put in tabu list and are excluded from further search. These help avoid being trapped in a local optimum [75].

Procedure TabuSearch
Input: State space min. problem

```
Tabu←{s}                                  Initialize tabu list
best←s                                    Initialize currently best state
Terminate←false                           Initialize termination flag
u←s                                       Start search from initial state
while (¬Terminate)
                                          Generate successors
    v←Select(Random(Succ(u) \Tabu))       Choose (random) successor
    if ( f (v)<f (u))  best←u             Evaluation improved, select v
    Tabu←Update(Tabu)                     Update tabu list
    Terminate←Update(Terminate)
    v←u                                   Continue with v
```

Output: State with low evaluation *best*

Source: S. Edelkamp, S. Schroedl. Heuristic Search Theory and Applications

Algorithm 3. Tabu search algorithm

If all neighbors are tabu, a move is accepted that worsen the value of the objective function to which an ordinary deepest decent method would be trapped [75]. Pseudocode for generic tabu search algorithm is given in Algorithm 3. There are some parallels between tabu search and simulated annealing, for e.g., randomized tabu search algorithm combining tabu list usage for search space reduction with selection used in simulated annealing can be viewed as generalization to simulated annealing; more information is given in [75].

2.1.10.4. Particle swarm optimization

Swarm intelligence techniques are inspired by the social behaviour of groups of various beings, such as ants, birds or bees. After analysing performance and movement of bird flocks or ants, patterns which described their synchronous movement in groups and regrouping after some influential change were identified. Thus the success of neighbouring individuals in the same groups combined with their own success was used as basis to develop a set of efficient optimization techniques, such as particle swarm optimization, ant colony optimization, bee colony algorithms, stochastic diffusion search and other.

Particle swarm optimization (abbr. as PSO) algorithm was introduced by Kennedy [127]. This technique is based on behaviour of flock of birds which search for food randomly in some area, knowing only the distance from the food. In PSO, each possible solution is represented as this bird and is called a particle, and its location relative to the object which is searched (food in this example) is defined by the fitness value. Thus all the particles have one fitness value defined by a function which is optimized, and each particle has one velocity to determine its flying direction and distance. All the particles perform search in the solution space by following currently the most optimal particle.

PSO is initialized to be a group of random particles and iteratively find the optimal solution. In each iteration each particle is updated itself by two extremums that are tracked. The first extremum is the optimal solution found by the particle itself (*pbest*), the other is the optimal solution found by the whole swarm (*gbest*). As the whole swarm can be regarded as the neighbour of the particles, the extremum in all the neighbors are called partial extremum.

gbest ParticleSwarmOptimization
Input: State space minimization problem

n – size of swarm
k – number of dimensions in particle
Initialize an n_k-dimensional swarm: $P \leftarrow Init(n)$
$t \leftarrow 0$ *number of iterations*
repeat
for $\forall p_x \in P$ *set the personal best position*
if $f(x_p) < f(y_p)$ *set the global best position*
$y_p = x_p$;
if $f(y_p) < f(\hat{y})$ $\hat{y} = y_p$
for $\forall p_x \in P$
for $j = 1..k$
update velocity of the particle

$$v_{pj}(t+1) = w * v_{pj}(t) + c_1 * Rand(0,1) * (y_{pj}(t) - x_{pj}(t)) + c_2 * Rand(0,1) * (\hat{y}_j(t) - x_{pj}(t))$$

$x_p(t+1) = x_p(t) + v_p(t+1)$ *update position of the particle*

$$y_p(t+1) = \begin{cases} y_p(t), if \ \ f(x_p(t+1)) \le f(y_p(y))) \\ y_p(t+1), if \ \ f(x_p(t+1)) > f(y_p(y))) \end{cases}$$

$\hat{y}(t) = \min(f(y_0(t)), ..., f(y_n(t)))$
$t \leftarrow t+1$
until stopping condition is true;
Output: State with low evaluation (optimal result) u
Source: adopted from Engelbrecht A. Computational intelligence: an introduction, 2nd Ed.

Algorithm 4. Particle swarm optimization algorithm

A canonical particle swarm optimization algorithm is given in Algorithm 4. Here $v_{pj}(t)$ is the velocity of particle p in dimension j at time step (iteration) t, $x_{pj}(t)$ is the position of particle i in dimension j at time step t. At each step of the algorithm, particles are displaced from their current position by applying a velocity vector to them. The magnitude and direction of their velocity at each step is influenced by their velocity in the previous iteration of the algorithm, simulating momentum, and the location of a particle relative to the location of its *pbest* and the *gbest*. At each step a particle is stochastically accelerated towards its previous best position and towards a neighbourhood (global) best position, thereby forcing particles to continually search in the most-promising regions found so far in the solution space. This move is a function of own history (experience), and the social influence of its peer group [27].

Note that Algorithm 4 represents global best PSO or *gbest* version of PSO algorithm where the neighbourhood for each particle is the entire swarm and star topology is used as a "social network". Local best PSO uses smaller neighbourhoods

for each particle and ring topology. The velocity equation has minor changes, as there is a vector of local best positions for each neighbourhood ($\hat{y}_{ij}(t)$) instead of one global best position $\hat{y}_j(t)$ in *gbest*

$$v_{pj}(t+1) = v_{pj}(t) + c_1 * rand(0,1) * (y_{pj}(t) - x_{pj}(t)) + c_2 * rand(0,1) * (\hat{y}_{ij}(t) - x_{pj}(t)) \quad (2.6)$$

The global best position $\hat{y}_j(t)$ is defined as a minimum of all values (as minimization problem is solved) [78]

$$\hat{\mathbf{y}}(t) \in \{\mathbf{y}_0(t), \ldots, \mathbf{y}_n(t)\} \mid \min(f(\mathbf{y}_0(t)), \ldots, f(\mathbf{y}_n(t))) \quad (2.7)$$

therefore, the local best position $\hat{y}_{ij}(t)$ is defined for each neighbourhood [78]

$$\hat{\mathbf{y}}_p(t+1) \in \{\mathbf{N}_p \mid f(\hat{\mathbf{y}}_p(t+1)) = \min(f(\mathbf{x})), \forall \mathbf{x} \in \mathbf{N}_p\} \quad (2.8)$$

where the neighbourhood is defined as [78]

$$\mathbf{N}_p = \{\mathbf{y}_{p-n_{N_i}}(t), \mathbf{y}_{p-n_{N_i}+1}(t), \ldots, \mathbf{y}_{p-1}(t), \mathbf{y}_p(t), \mathbf{y}_{p+1}(t), \ldots, \mathbf{y}_{p+n_{N_i}}(t)\} \quad (2.9)$$

Table 2. c_1 and c_2 parameter influence for Particle Swarm Optimization algorithm

c_1	c_2	**Possible effects**
$c_1 = 0$	$c_2 = 0$	Particle speed does not change (if it is not affected by inertia)
$c_1 > 0$	$c_2 = 0$	All particles are not influenced by the goals of the whole swarm and seek to obtain best solutions by themselves, turning search process into multiple independent local searches
$c_1 = 0$	$c_2 > 0$	The entire swarm is attracted to single point and perform single stochastic hill-climbing search procedure
$c_1 \to 0$	$c_2 \to 0$	If c_1 and c_2 are close to zero, each particle is encouraged to explore far from already found good points
$c_1 >> 0$	$c_2 >> 0$	More intensive search of regions close to already uncovered good points is encouraged
$c_1 >> c_2$		Cognitive factor dominates over social influence – more confidence in individual solutions, which results in excessive wandering of particles
$c_2 >> c_1$		More attraction to global best position which results in reduced level of exploration of global search space but global optima region is explored more intensively
$c_1 \approx c_2$, $c_1 = c_2$		Often most effective solution, as cognition and cooperativity influence to particle velocity is balanced

Source: created by the author using [27, 78].

PSO also has several parameters which have to be considered while setting the algorithm:

- Swarm size n - the number of particles in the swarm. The larger this parameter is, the more distributed in search space initial swarm is, yet, the larger number of computations has to be done and there are bigger chances for the search to become too random. This parameter is problem dependent – the more complex and

having more local minimas, the larger number of particles is preferred to obtain better solution.

- Neighbourhood size for lbest PSO – the smaller the neighbourhoods are, the less interaction between particles occurs, however, the less chances to get in local minimas [78].

- Number of iterations – this parameter is also problem dependable. Number of iterations which is too small might not lead to optimal solution.

- The acceleration coefficients, c_1 (nostalgia) and c_2 (envy), together with random vectors r_1 and r_2, control the stochastic influence of the cognitive (in itself) and social (confidence on neighbours) components on the overall velocity of a particle and the relative impact of the *pbest* and *gbest* locations on the velocity of a particle [27, 78]. Selection of these parameters is problem dependent; however, both Kennedy and Engelbrecht provide some guidelines for their selection which are summarized in Table 2.

- w is the inertia weight factor, introduced by Shi and Eberhart as a mechanism to better control the exploration and exploitation abilities of the swarm, avoiding usage of the maximum velocity V_{max} which serves as a constraint to control the global exploration ability of a particle swarm [74, 78]. Large w values encourage global exploration while small (but not too small) w promotes local exploitation. However, too small values eliminate the exploration ability of the swarm [76].

Another important parameter of PSO is the network structure defining particle communication for *lbest;* the most canonical and mostly investigated by various authors, including PSO author Kennedy [128] and Engelbrecht, are star, ring, wheel topologies. According to [78], star topology is referred to be the best for unimodal problems while ring topology can be a better choice for multimodal optimization problems. However, *gbest* algorithm will be considered in this research, therefore topology problem will not be analysed in more detail.

Velocity equation includes such components [76]:

- The previous velocity (also referred as *inertia* or *momentum*) $v_p(t)$, representing memory of the previous movement direction, i.e. movement in the immediate past. This prevents the particle from drastically changing direction, and to bias towards the current direction.

- The cognitive component (also referred as *nostalgia*) $c_1r_1(y_p - x_p)$, which quantifies the performance of particle *p* relative to past performances and enables the particle to return to their own best positions, resembling the tendency of individuals to return to situations or places that satisfied them most in the past.
- The social component, which quantifies the performance of particle *p* relative to a group of particles, or neighbours and represents a group norm or standard that individuals seek to attain. It is defined as $c_2r_2(\hat{y} - x_p)$ in case of *gbest* PSO and $c_2r_2(\hat{y}_p - x_p)$ in case of *lbest* PSO. This component enables the particle to head to the best position found by the particle's neighbourhood.
- Positive acceleration constants (also referred as learning factors) c_1 and c_2 used to scale the contribution of the cognitive and social components.

Overall, PSO technique has such advantages as relatively simple implementation, good abilities to solve problems with complex value functions which have many local minimas. Furthermore, there are many extensions to this PSO algorithm, such as Social-Based Particle Swarm Optimization, Sub-Swarm Based PSO, binary and discrete PSO, niching PSO; these are discussed in detail in [27, 78, 127].

2.1.10.5. Ant colony optimization

Ant colony optimization (abbr. as ACO) is another popular natural optimization technique based on swarm intelligence. This technique is based on coopearative behaviour of ants demonstrated when they look for food which helps to lead to this target other ants using pheromone trail. The more pheromone is on the path, the more chances it has to be selected. As more ants follow a specific trail, the more pheromone is left on that path, which attracts more ants to follow that path; thus an indirect communication as well as collective memory is enabled. Over time, shorter paths will have stronger pheromone concentrations, since ants return faster on those paths. Pheromone evaporates over time, with the consequence that the pheromone concentrations on the longer paths decrease more quickly than on the shorter paths. A number of techniques and their improvements which exploit ant behaviour are developed, including Simple Ant Colony Optimization (SACO), Ant System, Ant Colony System, Elitist Ant System, Rank-based AS (RAS), Max-Min Ant System (MMAS). Their review can be found in [26, 78].

Similarly to PSO, this technique also enables to solve complex problems, although basic ACO (SACO) has some problems which were identified by its creators, such as less stable and more dependent on parameter choice performance on complex graphs, convergence to non-optimal solution or even non-convergence, especially when the number of ants is large or pheromone evaporation setting is non-optimal [71].

2.2. Support Vector Machines

Support Vector Machines (abbr. as SVM) technique is also part of Vapnik-Chervonenkis statistical learning theory described in [225]. SVM as technique was first described in a paper by Cortes and Vapnik [52]. It has been targeted at industrial solutions and allowed to achieve good performance results in regression, time series analysis, pattern recognition and etc. This technique has wide adoption in various domains, such as bioinformatics, finance, engineering, text processing, image recognition and etc.

2.2.1. Basic SVM principles

Support Vector Machines are learning machines which are capable of performing binary classification or approximation of real value functions (regression tasks). SVM performs nonlinear mapping *n*-dimensional input data space to another feature space (possibly of even larger dimensions) which can be used in linear classification. At the same time empirical classification error is minimized and geometrical margin is maximized; because of these features SVM is also called maximum margin classifier.

Further only SVM for classification tasks will be discussed. In case of SVM, as well as other machine learning algorithms for classification, the main task is to evaluate function $f:X \rightarrow \{\pm 1\}$ which maps input and output data. According to [225], SVM can be formulated as following: if empirical data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \chi \times \{\pm 1\}$ is given with $x_i \in R^N$, $y_i \in \{-1,1\}$, the task is to find a decision function $f_{\mathbf{w},b}$ with the property $f_{\mathbf{w},b}(x_i) = y_i$, $i = 1..n$. Here χ is a non-empty set, containing x_i structures; y_i are called *targets* or more often - *labels*. This means that for each new structure described by vector $x \in \chi$ it is needed to find $y \in \{\pm 1\}$ by selecting y in a way that (x,y) would be as similar as possible to the instances used for training classification machine. The similarity measures for χ and y

can be formally described as *kernel* function $k: \chi \times \chi \to R, (x, x') \to k(x, x')$, which returns a scalar that characterizes the similarity of *x* and *x'*. Dot products are simple examples od similarity measure; a *dot product* for *x* and *x'* is defined as

$$(x \cdot x') := \sum_{i=1}^{N} (x_i)(x')_i \tag{2.10}$$

with x_i ir x'_i as corresponding *x* ir *x'* i-th elements. To use product with dot vectors they have to be represented in feature space R^L, $L < N$. It is defined as representation of data χ in feature space R^L as a mapping $\Phi: \chi \to F, x \to \bar{x}$. SVM is also deterministic: given a machine defined as a set of possible mappings $x \to f(x, \alpha)$ which has to learn a mapping $x_i \to y_i$ where functions *f(x, α)* have particular parameters *α*. This machine always gives the same result *f(x, α)* for each input data instance *x* and parameter *α* [225].

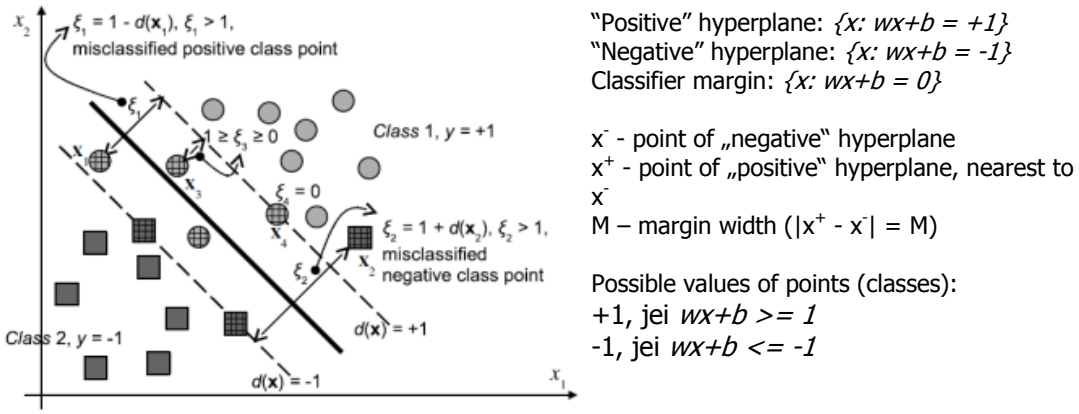
Scholkopf gives three advantages of such SVM representation [191]:

1. Dot product in feature space F gives an ability to define similarity measure

$$k(x, x') := (\bar{x} \cdot \bar{x}') = (\phi(x) \cdot \phi(x')) \tag{2.11}$$

2. Possibility to work with data using both linear algebra and analytical geometry concepts.

3. The possibility to select mapping Φ gives an opportunity to develop a large set of various learning algorithms as data representation can be more precisely adopted to the problem by selecting more suitable nonlinear mapping.



"Positive" hyperplane: *{x: wx+b = +1}*
"Negative" hyperplane: *{x: wx+b = -1}*
Classifier margin: *{x: wx+b = 0}*

x^- - point of „negative" hyperplane
x^+ - point of „positive" hyperplane, nearest to x^-
M – margin width ($|x^+ - x^-| = M$)

Possible values of points (classes):
+1, jei *wx+b >= 1*
-1, jei *wx+b <= -1*

Source: Huang T. M., Kecman V., Kopriva I. Kernel based algorithms for mining huge data sets: supervised, semi-supervised, and unsupervised learning.

Figure 2. Linear support vector machine

Figure 2 gives an illustration of a simple binary SVM for classification, together with graphical illustration. The classifier is described as separating

hyperplane with binary solutions on both of its sides (i.e., solutions equal to +1 or -1). The main objective is to find a hyperplane which would minimize margin error. This hyperplane is described as a set of support vectors (data points, for which and only for which Lagrangian is not equal to zero). Finding these vectors from training data can be formulated as an optimization problem to maximize separating margin *M*

$$\max \frac{2}{\|\mathbf{w}\|} = \min \frac{1}{2}\|\mathbf{w}\| = \min \frac{1}{2}\mathbf{w}^T\mathbf{w} \qquad (2.12)$$
$$s.t.\ \ y_i(w^T\phi(x_i)+b) \geq 1,\ i = 1,..,n$$

where n denotes a number of training data points. The optimization problem can be solved using Lagrangian [116]

$$L_p(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n}\alpha_i\{y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\} \qquad (2.13)$$

according to Karush-Kuhn-Tucker (KKT) conditions for the optimum of a constrained function, which are defined as vanishing derivatives of $L_p(\mathbf{w}, b, \alpha)$ with respect to primal variables **w** and *b*

$$\frac{\partial L_p}{\partial \mathbf{w}_o} = 0 \qquad \mathbf{w}_o = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \qquad (2.14)$$

$$\frac{\partial L_p}{\partial b} = 0 \qquad \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad (2.15)$$

and the KKT complementarity conditions

$$\alpha_i\{y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\} = 0, i = 1,...,n \qquad (2.16)$$

Thus dual formulation is obtained using Lagrangian [36, 116]:

$$\min \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n}\alpha_i \qquad (2.17)$$
$$\sum_{i=1}^{n} y_i\alpha_i = 0,\ \forall i : 0 \leq \alpha_i \leq C$$

where the number of training examples is denoted by *n*, training vectors $X_i \in R, i = 1,..., n$ and a vector $y \in R^n$such that $y \in \{-1; 1\}$. α is a vector of *n* Lagrange multipliers, where each α_i corresponds to a training example (x_i, y_i). According to [116], parameters for optimal hyperplane $\mathbf{w}_0$ and b_0 are obtained using

$$\mathbf{w}_o = \sum_{i=1}^{n}\alpha_{oi} y_i \mathbf{x}_i \qquad (2.18)$$

$$b_0 = \frac{1}{N_{SV}}\sum_{s=1}^{N_{SV}}(y_S - \mathbf{x}_s^T\mathbf{w}_0), s = 1,..., N_{SV} \qquad (2.19)$$

where N_{SV} is the number of support vectors. Support vectors are defined as instances

which have nonzero α_{oi} and support forming the decision function. The decision function (optimal separating hyperplane) then becomes

$$\operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x_j) + b\right) \tag{2.20}$$

The problem of the data overlapping can be solved by generalizing the optimal 'hard' margin algorithm, introducing the nonnegative slack variables ξ_i *(i =1,…,n)* for the overlapped data points; they are also referred as distances of the points crossing the margin from the corresponding margin [116]. Therefore, Vapnik's SVM (further referred as C-SVC classifier) is defined as a primal convex quadratic optimization problem [36]:

$$\min_{w,b,\zeta} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \zeta_i \tag{2.21}$$

$$s.t.\ \ y_i(w^T\phi(x_i) + b) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1,.., n$$

where C is a regularization parameter that determines the trade-off between the maximum margin and the minimum classification error which comes from data left on the "wrong" side of a decision boundary (data inside soft margin). C is also referred as penalty parameter that determines the trade-off between the training error and VC dimension of the model [116]. The cost function, second part of optimization problem in (2.21), can be generalized; thus the optimization problem becomes

$$\min_{w,b,\zeta} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \zeta_i^k \tag{2.22}$$

$$s.t.\ \ y_i(w^T\phi(x_i) + b) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1,.., n$$

This is usually solved only for $k = 1$ or $k = 2$, and such soft margin SVMs are referred as L1 and L2 SVMs. More information on their Lagrangian derivation and computing issues can be found on [116]. The decision function for SVM is defined as [36]

$$\langle \phi(\mathbf{x}) \cdot \mathbf{w} \rangle + b = 0 \tag{2.23}$$

If training vectors x_i are not linearly separable, they are mapped into a higher (maybe infinite) dimensional space in which a linear hyperplane be can produced by the kernel function $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$. The most popular, implemented in various implementations and commonly referred as used are [36, 225]:

- linear: $K(x,z) = x*z$, (2.24)
- polynomial: $K(x,z) = (\langle x \cdot z \rangle + \theta)^d$, (2.25)

- radial basis function (RBF): $K(x,z) = e^{\frac{|x-z|^2}{2\sigma}}$, where σ > 0; (2.26)
- sigmoid: $K(x,z) = \tanh(\beta\langle x,z\rangle - \lambda) = \frac{1}{1+e^{-\beta\langle x,z\rangle-\lambda}}$, $\beta,\gamma \in R$ (2.27)

One of the biggest challenges in successful SVM application is the selection of SVM parameters. Unfortunately, this is complicated – no guidelines are given in literature for this step. Steinwart and Chrismann state that "there is currently no practical method known that chooses the hyperparameters of SVMs in an optimal manner for all data sets and is applicable for sample sizes of any size" [202]. Thus various heuristic and metaheuristic search techniques are often employed to solve this task; however, their application often results in higher computational cost.

2.2.2. SVM algorithms

Various improvements for SVM technique have been developed since its initial release. There are many different implementations and algorithms which were developed while extending and improving C-SVM technique using different numerical methods and computational tricks which allowed to obtain better results or to redure computational complexity and time needed for training SVM classifier. This section describes some of the improvements which were made after the original SVM.

ν–SVC (Scholkopf et al., 2000)**.** First described in [192]**,** this technique replaces the use of cost parameter *C* in C-SVC with $\nu \in [0,1]$ parameter, which is used to control the umber of support vectors and training error. This technique is formulated as in [36]: given training vectors $X_i \in R, i = 1,..,l$ in two classes and a vector $y \in R^l$ such that $y_i \in [-1;1]$; the primal form is

$$\min_{\mathbf{w},b,\xi,\rho} \frac{1}{2}\mathbf{w}^T\mathbf{w} - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \quad (2.28)$$
$$s.t.\ \ y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq \rho - \xi_i$$
$$\xi_i \geq 0, i = 1...l, \rho \geq 0$$

Its dual formulation is [36]:

$$\min_{\alpha} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \quad (2.29)$$
$$s.t. 0 \leq \alpha_i \leq \frac{1}{l}, i = 1..l, \mathbf{e}^T\boldsymbol{\alpha} \geq \nu, \mathbf{y}^T\boldsymbol{\alpha} = 0$$

where **e** is the vector of all ones, *Q* is *l x l* positive semidefinite matrix, $Q_{ij} \equiv$

$Y_iY_jK(x_i,x_j)$ and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel function. The decision function is the same as in case of C-SVC. As Chang and Lin point out in [36], Crisp ir Burges have proved that $e^T\alpha \geq \nu$ can be changed to $e^T\alpha = \nu$ .After computing α / ρ the decision function becomes:

$$\operatorname{sgn}(\sum_{i=1}^{l} y_i \frac{\alpha_i}{\rho} K(\mathbf{x}_i, \mathbf{x}) + b) \tag{2.30}$$

The final decision function is $y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) = \pm 1$, which is the same as for C-SVC case.

Sequential Minimal Optimization (Platt, 1999). Sequential Minimal Optimization, abbr. as SMO, breaks large quadratic programming problems into a series of smallest possible quadratic programming problems which are solved analytically, thus avoiding using a time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, and, as matrix computation is avoided, SMO scales somewhere between linear and quadratic in the training set size for various test problems, while the standard chunking SVM algorithm scales somewhere between linear and cubic in the training set size [175]. This technique or it's principles is implemented in many modern SVM solvers.

Least Squares SVM (Suykens, Vandevalle, 1999). Least Squares SVM, abbr. as LS-SVM, aims to solve a set of linear equations instead of convex quadratic programming performed in case of standard SVM. LS-SVM is also referred as kernel Fisher discriminant analysis [223]. The problem can be formulated as [198]:

$$\begin{aligned} &\min_{w,b,e} f(w,b,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \\ &\text{s.t.} \quad y_k[w^T \varphi(x_k) + b] = 1 - e_k, \; k = 1,..., N \end{aligned} \tag{2.31}$$

Lagrangian for LS-SVM is defined as

$$L(w,b,e;\alpha) = f(w,b,e) - \sum_{k=1}^{N} \alpha_k \{y_k[w^T \varphi(x_k) + b] - 1 + e_k\} \tag{2.32}$$

with α_k as Lagrangian multipliers.

SVM$^{\text{Light}}$ (Joachims, 2001). Similarly to SMO, this SVM implementation uses dual representation of SVM optimization problem, but, according to its author it can be twice as faster than SMO. This technique is developed for large scale SVM learning and generates much smaller number of support vectors, compared to training set [119]. Most of its generated support vectors have α near upper *C* boundary.

Standard SVM is reformulated as a solution of subproblem, which uses dataset *B*, when *B* is selected by [119]

$$\begin{aligned} &\min \nabla f(\alpha_k)^T d \\ &-1 \le d \le 1, \\ &d_i \ge 0, if\ (\alpha_k)_i = 0, \\ &d_i \le 0, if\ (\alpha_k)_i = C, \\ &|\{d_i \mid d_i \ne 0\}| \le q \end{aligned} \tag{2.33}$$

BSVM (Hsu, Lin, 2002). This can be viewed as an enhanced SVM$^{\text{Light}}$ version. The following problem, where *B* is a data subset used to formulate a subproblem, is solved [108]:

$$\begin{aligned} &\min_{w,b,\zeta} \frac{1}{2} w^T w + \frac{1}{2} b^2 + C \sum_{i=1}^{l} \zeta_i \\ &y_i (w^T \phi(x_i) + b) \ge 1 - \sum_{j \in N} Qi_j \alpha_j - \zeta_i, \\ &\zeta_i \ge 0, i \in B \end{aligned} \tag{2.34}$$

B is chosen by $0 \le \alpha_k + d \le C, |\{d_i \mid d_i \ne 0\}| = q$, which, according to the authors of BSVM, is more consistent with solution condition $0 \le \alpha_k + d \le C$ and guarantees convergence.

Pegasos (Primal Estimated sub-GrAdient SOlver for SVM) (Shalev-Shwartz et al., 2007). Stochastic gradient descent (abbr. as SGD) is usually used to minimize objective function expressed as a sum of differentiated functions, but it can also be used as SVM type classifier with hinge loss function. Pegasos is a modified SGD where every gradient's next step is performed together with a projection step [195]. Despite the simplicity of this technique, the results obtained are comparable to SMO and SVM$^{\text{Light}}$ [195].

Core Vector Machines (CVM) (Tsang et al., 2005). Core Vector Machines use computational geometry formulations of kernel methods showing that they can be equivalently formulated as minimum enclosing ball to obtain provably approximately optimal solutions with the idea of core sets [216]. It has linear time and space complexity and can be much faster with larger data sets while original SVM algorithm is known to have complexity of $O(n^3)$ [216]. A modification for least-squares classification CVM-LS has also been developed [216]

Least-Squares Core Vector Machines (CVM-LS) (Tsang et al., 2005). Implementation based on Core Vector Machines which solves a set of linear equations instead of quadratic programming [216]**.** It also has insensitive loss for

sparse least-squares classification.

Ball Vector Machines (BVM) (Tsang et al., 2007). It is even faster than CVM as it solves simpler minimum enclosing ball with fixed radius problem. The approximate SVM solution obtained is also close to the truly optimal SVM solution [216]. Its implementation does not require numerical solvers thus it can be easier to implement [216]

Linear Transductive SVM (Keerthi, 2005). This is a family of semi-supervised linear support vector classifiers that are designed to handle partially-labeled sparse datasets with possibly very large number of examples and features. They use modified finite Newton techniques and Deterministic Annealing (DA) algorithm for optimizing semi-supervised SVMs which is designed to alleviate local minima problems [199]. These classifiers feature linearly regularized least squares classification, semisupervised classification, multi-switch linear Transductive L2-SVMs.

Potential SVM (Hochreiter, Obermayer, 2006). Let $\boldsymbol{X}$ be the matrix of data vectors in some high-dimensional feature space φ, $\boldsymbol{w}$ be the normal vector of a separating hyperplane, $\boldsymbol{y}$ the attributes (binary in case of classification, or real valued in case of regression), and $\boldsymbol{K}$ - the kernel matrix. Then the P-SVM "primal" optimization problem has the form [132]

$$\begin{aligned} &\min_{\mathbf{w},\xi^+,\xi^-} \frac{1}{2}\left\|\mathbf{X}^T\mathbf{w}\right\| + C\mathbf{1}^T(\xi^+ + \xi^-) \\ &s.t.\ \ \mathbf{K}^T(\mathbf{X}^T\mathbf{w} - \mathbf{y}) + \xi^+ + \varepsilon\mathbf{1} \geq 0 \\ &\qquad \mathbf{K}^T(\mathbf{X}^T\mathbf{w} - \mathbf{y}) - \xi^+ - \varepsilon\mathbf{1} \leq 0 \\ &\qquad \mathbf{0} \leq \xi^+, \xi^- \end{aligned} \tag{2.35}$$

The parameters C and φ correspond to the two different regularization schemes, where φ-regularization has been proven more useful for feature selection and the C-regularization for classification or regression problems. ξ^+ and ξ^- are the vectors of the slack variables describing violations of the constraints. The parameter C has similar importance as in C-SVC case as it limits the support vector weights α. If it is infinite, no regularization occurs. If it tends to zero, the largest weights of support vectors decrease and the possibly increase of the training error will be compensated through finding similar data vectors and increasing their weights (they become support vectors) [132]. The second parameter φ controls the tolerance level of small training errors; is also closely realated to number of support vectors - if it tends to

infinity, the primal P-SVM problem is solved without support vectors, and if it tends to zero, the tolerance level decreases and the training error decreases too as far as the number of support vectors increases. Thus φ controls the tradeoff between a poor representation of the training data and overfitting [132].

Parallel GPDT (Parallel Gradient Projection-based Decomposition Technique). This technique (Serafini, Zanghirati, Zanni, 2007) uses a technique similar to the ones used in SMO and SVM^{Light} split the SVM quadratic programming problem into a sequence of smaller subproblems. Each one of these subproblems is solved by a suitable gradient projection method (GPM), either Generalized Variable Projection Method (GVPM) or the Dai-Fletcher method (DFGPM). The performance benchmark performed by the authors of Parallel GPDT shows that it can perform comparatively to LIBSVM and SVM^{Light} implementations [258].

Proximal SVM (PSVM) (Fung, Mangasarian, 2001). The points are classified by assigning them to one of two nearest parallel hyperplanes [86].

Active Support Vector Machines (ASVM) (Mangasarian, Musicant, 2000). SVM technique based on active sets strategy for solving bounded quadratic programming problems. The distance between hyperplanes is maximized and margin error is minimized using 2-norm square of distance function [153]

Smooth Support Vector Machine (SSVM) (Lee, 2001). SSVM uses a smooth unconstrained optimization reformulation of the traditional quadratic program. It uses very fast Newton-Armijo algorithm for solving and performance benchmarks showed that it can be comparable or faster to SMO and SVM^{Light} solvers as well as it resulted in lower computational time. It can also be extended to identify nonlinear hyperplanes [142]

Newton Method for LP Support Vector Machine (LPSVM) (Fung, Mangasarian, 2001). LPSVM uses a fast Newton method that suppresses input space features in very high dimensional spaces thus it can be very effective when solving classification problems which use feature selection as a preprocessing step [85].

Lagrangian SVM (LSVM) (Mangasarian, Musicant, 2000). SVM method based Lagrangian reformulation of linear SVM standard quadratic programming [154].

A recent survey of novel SVM algorithms research is also given by Tian et al. [210]. Increasing popularity of SVM also resulted in increase of patents related to

SVM and its applications. Many of algorithms described above are described in them in more detail, such as support vector machine - recursive feature elimination (SVM-RFE) technique [237] and others.

2.2.3. SVM extensions to multiclass classification

As Section 2.2.1 shows, SVM is originally defined as binary classification problem. Various extensions are proposed for multiclass classification, where $y \in [1,2,..,N]$. Three main options are discussed in this section, namely *one-vs-all* (OVA), *one-vs-one* (OVO) and Crammer-Singer multiclass extension. For OVO-SVM, all possible pairwise SVMs are generated, each using training examples from two classes chosen out of *N* classes. The decision function for each pair of classes *i* and *j* is then defined as [68]

$$f^{ij}(x) = \left\langle \phi(\mathbf{x}) \cdot \mathbf{w}^{ij} \right\rangle + b^{ij} \tag{2.36}$$

For a N-class problem *N(N-1)/2* different decision functions are used. The common decision between the generated classifiers can be obtained by using various strategies. Most common of them is majority voting, also known as "max-wins". The decision function then can be defined as [68]

$$\arg\max_i \sum_{j \neq i, j-1}^{k} sign(f^{ij}(\mathbf{x})) \tag{2.37}$$

In case of OVA-SVM, given a *N*-class problem, *N* binary SVM models are constructed, and each *i*-th SVM is trained with all of the training examples in the *i*-th class with positive labels and all other examples with negative labels [68]. The final class is selected according to SVM with the highest output value. i.e., final decision function becomes

$$\arg\max_{i-1,N} (\left\langle \phi(\mathbf{x}^i) \cdot \mathbf{w}^i \right\rangle + b^i) \tag{2.38}$$

Crammer and Singer [53] proposed an approach for multiclass problems by solving a single optimization problem:

$$\min_{w_m, \xi_i} \frac{1}{2} \sum_{m=1}^{k} w_m^T w^m + C \sum_{i=1}^{l} \xi_i \tag{2.39}$$

$$\text{subject to } w_{y_i}^T x_i - w_m^T x_i \geq e_i^m - \xi_i,\ i = 1,..,l$$

$$e_i^m = \begin{matrix} 0, if\ y_i = m \\ 1, if\ y_i \neq m \end{matrix}$$

The decision function is similar to OVA-SVM, except that bias coefficients b^i are missing:

$$\arg\max_{i-1,N}(\langle \phi(\mathbf{x}^i)\cdot \mathbf{w}^i \rangle) \tag{2.40}$$

2.2.4. Linear SVM

These classifiers have several advantages over SVM implementations which use kernel functions; however, they are related to training speed and reduced complexity, as the absence of kernel mapping step simplifies training and reduces the number of calculations required to perform during training phase. However, linear SVM classifiers are not as flexible as SVMs with kernel functions and thus might result in smaller accuracy.

A linear SVM classifier is defined as follows [80]: given training vectors $x_i \in R^n$, i = 1,..,l in two class, and a vector $y \in R^l$ such that y_i = {+1,-1}, a linear classifier generates a weight vector w as the model using a decision function sgn($w^T x$). In some cases, the discriminant function of the classifier includes a bias term, *b*. LIBLINEAR handles this term by augmenting the vector **w** and each instance x_i with an additional dimension using constant *B*, which is specified by the user [80]. According to [80], L1-SVM and L2-SVM are solved using coordinate descent method [108, cited by 80]; for logistic regression and L2-SVM, a trust region Newton method developed by Lin et al.[146, cited by 80] is implemented.

Table 3. Linear SVM classification algorithms and their formulations

Algorithm	**Minimization problem**
L2-regularized logistic regression	$\min_w \frac{1}{2} w^T w + C\sum_{i=1}^{l} \log(1+e^{-y_i w^T x_i})$
L2-regularized L2-loss SVC (dual)	$\min_\alpha \frac{1}{2}\alpha^T Q\alpha - \mathbf{e}^T\alpha,\ 0 \le \alpha_i \le C, i = 1...l$
L2-regularized L2-loss SVC (primal)	$\min_w \frac{1}{2} w^T w + C\sum_{i=1}^{l} (\max(0,1-y_i w^T x_i))^2$
L2-regularized L1-loss SVC	$\min_w \frac{1}{2} w^T w + C\sum_{i=1}^{l} (\max(0,1-y_i w^T x_i))$
L1-regularized L2-loss SVC	$\min_w \|w\|_1 + C\sum_{i=1}^{l} (\max(0,1-y_i w^T x_i))^2$
L1-regularized logistic regression	$\min_w \|w\|_1 + C\sum_{i=1}^{l} \log(1+e^{-y_i w^T x_i})$
L2-regularized logistic regression (dual)	$\min_\alpha \frac{1}{2}\alpha^T Q\alpha + \sum_{i:\alpha_i>0} \alpha_i \log\alpha_i + \sum_{i:\alpha_i<C} (C-\alpha_i)\log(C-\alpha_i) - \sum_{i=1}^{l} C\log C$

Source: created by author using (Fan et al, 2006)

Table 3 gives the main formulations of linear SVM and logistic regression classifiers implemented in LIBLINEAR package; more information and their implementations details can be found on paper by Fan et al. One-vs-all (OVA) strategy is used for multiclass classification problems, discussed in Section 3.2.

2.2.5. SVM software and implementations

As it was discussed in Section 2.2.2, there is a number of various SVM implementations developed mostly under scientific and research purposes. It is useful to analyse the possibilities of SVM packages in terms of functionality and problem solving capabilities by comparing implementations of these packages. SVM software can be viewed as basic framework for further model development]; in order to develop such techniques specialized knowledge from machine learning, data mining, particular domain and programming languages fields is necessary. Detailed analysis of these algorithms is beyond the scope of this work; more information about them can be found in specialized literature or provided references.

Table 4. Comparison of SVM implementations

	LibSVM	**BSVM**	**UniverSVM**	**mySVM**	**TinySVM**	**SVM & KM Toolbox**	**SimpleMKL**	**SVM Light**	**CVM**	**PSVM**	**GPDT**	**LIBLINEAR**	**LS-SVM**	**Lagrangian SVM**	**ASVM**	**SSVM**	**LPSVM**	**Proximal SVM**
Problems solved																		
Classification	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Regression	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		☑	☑					
Ranking								☑		☑								
Clustering													☑					
Feature selection						☑				☑			☑					
Number of classes																		
One-class	☑					☑		☑	☑									
Two-class	☑			☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Multi-class		☑	☑			☑		☑ [1]	☑			☑	☑					
Programming language																		
JAVA	☑			☑	☑							☑						
MATLAB	☑					☑	☑					☑	☑	☑	☑	☑	☑	☑
C/C++	☑	☑	☑	☑	☑			☑	☑	☑	☑	☑						
Python	☑				☑													
Perl	☑				☑													
Kernel function																		
Linear	☑	☑	☑	☑		☑		☑	☑	☑	☑		☑					
Polynomial	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑		☑					

[1] Using SVM Multiclass extension

	LibSVM	BSVM	UniverSVM	mySVM	TinySVM	SVM & KM Toolbox	SimpleMKL	SVM Light	CVM	PSVM	GPDT	LIBLINEAR	LS-SVM	Lagrangian SVM	ASVM	SSVM	LPSVM	Proximal SVM
Sigmoid	☑	☑	☑	☑	☑	☑		☑	☑	☑								
RBF	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑		☑					
ANOVA				☑	☑				☑									
Neural				☑									☑					
Laplas									☑									
User-defined	☑	☑								☑								
Point				☑														
Sum				☑														
Inverted distance									☑									

Source: P. Danenas, G. Garsva. SVM and XBRL based decision support system for credit risk evaluation.

Note that some frameworks or algorithms are more suitable for specific tasks (such as document classification, semi-supervised learning and etc.) or large scale learning; the latter is relevant to research described in this work. Analysis of various works shows that LibSVM, SVM^{Light}, LS-SVM, also LIBLINEAR implementations are the mostly applied for experimenting. Note that commercial SVM implementations (in STATISTICA, SQL Server, Oracle) are not discussed in this work; yet, analysis of such implementations identified that they are mostly, based on C-SVC. Several implementations are available in open source packages, such as Weka [236] or RapidMiner [178] together with integration of popular SVM packages LibSVM and LIBLINEAR; thus these tools are chosen for SVM research.

A comparative analysis of these packages is given in Table 4[2] [89].The comparison includes LibSVM [36], LS-SVM[66], also Lagrangian SVM [154], ASVM [153], SSVM [142], LPSVM [85] and Proximal SVM [86] by Mangasarian et al., SVMLight [120], BSVM [29], UniverSVM [220] SVM&KM Toolbox [33], SimpleMKL [198], mySVM [184], TinySVM [211], Core Vector Machines [51], PSVM [132], GPDT [258] and LIBLINEAR [80]. Currently there is a lack of professional software ar particular toolboxes particularly for SVM and SVM based methods which also integrate at least several SVM implementations. Another difficulty which comes when trying several SVM algorithms is their implementations in different programming languages which makes it difficult to combine them, e.g., to develop ensemble models. Yet there are attempts to develop such tools, such as SVM

[2] Appendix D gives their more extensive list and more detailed analysis

and Kernel Methods Matlab Toolbox, developed by Canu et al. [33], which offers such features as SVM classification using linear and quadratic penalization of misclassified examples, as well as using Nearest Point algorithm, Multiclass SVM using one-vs-all, one-vs-one and M-SVM techniques, regularisation networks, SVM based feature selection, model optimization techniques, tools such as SVM AUC optimization (Ranking SVM, ROC SVM) and SVM bounds (Span estimate, radius/ margin), as well as kernel PCA and kernel discriminant analysis, Kernel Basis Pursuit and Least Angle Regression (LARS), wavelet kernel for classification and regression [33]. Currently this tool is slightly outdated, as the last release dates to 2008. Another interesting solution is Shogun[3] Toolbox, which is written in C++ and constantly updated. It offers several features that are not found in implementations mentioned above, such as dimensionality reduction, data preprocessing, large scale learning, multitask learning, integrated performance measures. It also integrates several different SVM implementations, such as LIBLINEAR, LibSVM, SVM^{Light}, SVMLin, GPDT, and provides implementations of the most common SVM kernels (Eq. 2.25, 2.26, 2.27), as well as a number of recent string kernels which are more relevant for document classification tasks. Notably, it has interfacing to Matlab, R, Octave, Java, C#, Lua, Ruby and Python languages and is also targeted at large scale learning which makes it a good choice for high performance and large scale systems. Yet, complicated compilation process on Windows limits its application for machine learning specialists with less experience in system programming.

2.2.6. SVM advantages and disadvantages

Such advantages of SVM might be identified after its analysis:

- Converting problem to QP problem results in globally optimal solution – local minimas are avoided;
- Control of parameter space by using optimal boundary parameter;
- Good classification performance;
- Avoidance of overtraining, overfitting, architecture selection and testing problems;
- Many algorithms and their modifications, large basis of performed research;

[3] Shogun - A Large Scale Machine Learning Toolbox, http://www.shogun-toolbox.org

- Possibility of scaling and parallelization [37].

Therefore SVM also has its disadvantages:

- Highly complex choice of suitable parameters;
- Slow in testing;
- Complexity of implementation, resulting in complex development of hybrid techniques;
- Demand for computational resources;
- Lack of professional software and heterogeneity of implementations, resulting in more complex tasks of SVM comparison and implementation;
- Orientation to binary classification, which results in increased classification complexity and need of special extensions.

2.2.7. SVM research in credit risk evaluation

This section reviews research made on both SVM-based, hybrid SVM and ensemble techniques. As the number of papers is large only main points are summarized in this section. More detailed analysis can be found in other papers such as surveys of Danenas et al. [63] and Jayanthi et al. [117]; Appendix F also gives an extensive survey of such research. They show that SVM often outperformed other techniques, such as backpropagation neural networks (BPNN), linear discriminant analysis (LDA), decision tree techniques such as CART and C4.5 and etc. Most of the papers analysed in these surveys were based on binary classification (e.g., discrimination between bankrupt and non-bankrupt companies); only few researchers (Chen et al. [47], Chong et al. [49], Hu et al. [110], Huang et al. [112], Kim, Ahn [130]) applied SVM for multiclass classification problems, such as rating analysis or assessment model development. The number of ratios used in such research[4] varied – several papers were based on evaluation of datasets with 6 ratios and less (Chen et al. [46], Gao et al. [88], Yun et al. [256], Ravikumar et al. [179]), however prediction accuracy was still high, over 87%. Other authors used datasets with higher dimensionality – Ahn et al. [3] used 39 ratios, Min et al. [164] developed model using 50 ratios, Chen et al. [47] - with 72 ratios, Wang [234] – with 52 ratios, Wei et al. [235] used 65 attributes. However, such number of ratios didn't result high accuracy – only Wang obtained accuracy higher that 90%. This might contradict to the fact that

[4] Only the number of ratios used in training procedure is considered (for e.g., after feature selection or manual selection). This is often different from initial number of features

SVM shows best performance with high-dimensional datasets, although this cannot be viewed as exact proof of this fact as data used in research varied.

Yu et. al also describe a series of SVM based algorithms for credit risk evaluation and present most of their wotk in their book [253]. These results are summarized in Appendix F. Most of their experiments were performed on German, Australian, and Japanese datasets, as well as England corporate credit dataset. This makes it esasier to compare results of different classifiers and classification paradigms. It can be easily seen that SVM based hybrid classifiers (especially combining fuzzy logic or rough sets integration and/or ensemble techniques) outperformed other, yet the difference in terms of accuracy was not always very significant.

Obtaining data for model development in credit risk field is known as one of the biggest challenges as such data is rarely available online; therefore different authors use datasets that they obtain from various sources. This makes it difficult to perform benchmarking of different techniques. Therefore, the size of datasets used in such research is also variable: some researchers used large datasets, e.g., Kou et al. [136] used 6000 instances for training and 5720 instances for testing, Wang et al. [231] used 2000 instances, Wang et al. [232] used 18960 instances, Yoon et al. [252] and Ribeiro et al. [182] reported on working with more than 400.000 instances. Ribeiro et al. [181] obtained a dataset consisting of 60000 instances from 2005-2006 period and performed training and testing procedures on different time periods using S-isomap, k-NN, SVM and RVM techniques; SVM also outperformed other techniques. However, only several datasets used in such research are available online such as German[5] (consists of 1000 instances with 20 attributes), Australian credit approval dataset[6] (690 instances with 14 attributes) at UCI Machine learning repository; their specifications are given in Appendix J. They were used in research by Ghodselahi [90], Huang et al. [113], Yun et al. [256], Li et al. [145], Liu et al. [147], Peng et al. [173], Zhou et al. [264, 265]. SVM was used as a standalone technique by Chen et al. [45], Chen et al. [46], Chen et al. [47], Yang et al. [249], Yoon et al. [252] and others, as well as in combination with various other soft

[5] German Credit Data Set,
http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29
[6] Australian Credit Approval Data Set,
http://archive.ics.uci.edu/ml/datasets/Statlog+%28Australian+Credit+Approval%29

computing approaches. Integration of fuzzy logic (Chaudhuri et al. [39], Chong et al. [49], Hao et al. [100], Hao et al. [101], Zhou et al. [266]) also proved to show better classification results than original SVM; all of these papers, except [266], reported accuracy above 85%; aplication of rough sets and SVM hybrid techniques was also successful (Lv et al. [150], Ping et al. [173], Wang et al. [232], Wang et al. [233]). Zhou et al. [262] reported that rough sets based classifier performed even better than fuzzy SVM and GA optimized SVM. Other authors used genetic algorithm and swarm intelligence techniques for classifier selection; several reported that PSO based SVM performed better than genetic algorithm (Chen et al. [42], Chen [44], Yun et al. [256], Jiang et al. [118]).Relevance Vector Machines was also applied for classification (Li et al.[145], Ribeiro et al. [181], Ribeiro et al. [182]); RVM ensemble based technique also outperformed SVM as well as canonical RVM (Li et al.[145]). SVM based ensembles also showed better performance than individual SVM or fuzzy SVM models (Ravikumar et al. [179], Yu et al. [253]).

Besides SVM application to typical bankruptcy identification or ratings analysis problems, it can be used to solve other credit risk related techniques – Galkus et al. [87] applied SVM and ensemble techniques for identification of recovering after bankruptcy procedure using data from UCLA-LoPucki Bankruptcy Research database, Bose et al. [27] – to forecast future of dotcoms. Härdle et al. [103] proposed an approach to estimate probabilities of default using SVM. The achieved results proved SVM to be promising, very efficient and therefore widely, developed and applied technique.

2.3. The advantages and disadvantages of computational intelligence methods

L. Yu et al. [253] marked main advantages of the most popular machine learning methods used for credit risk evaluation research and ranked them (Table 5). Various authors give main advantages and disadvantages of various methods which perfectly explain the rankings given by L. Yu et al. These aspects are summarized in Table 6. It can be seen that the SVM is among the most efficient ones, as it can be benchmarked with neural networks or hybrid model for its accuracy and flexibility, yet it also among most complex and hardly interpretable, which makes development of hybrid techniques much more sophisticated.

Table 5. Advantages and disadvantages of machine learning methods according to Yu et al.

Method	Accuracy	Interpretability	Simplicity	Flexibility
MDA, logistic regression	★★	★★★	★★★	★
Decision Tree	★★	★★★	★★	★
k-Nearest Neighbour (KNN)	★	★★★	★★★	★
Linear programming	★	★★★	★★	★★★
Neural networks	★★★	★	★	★
Evolutionary computing	★★	★	★	★
Rough sets	★★	★	★★	★
SVM	★★★	★★	★	★★★
Ensemble models	★★★	★	★	★★

Source: L. Yu, K. K. Lai, S. Wang, L. Zhou. Bio-Inspired Credit Risk Analysis

The following techniques are compared: neural networks (NN), evolutionary computing (EC), mainly genetic algorithm (GA), fuzzy logic (FL), Support Vector Machines (SVM), case-based reasoning (CBR), rough sets (RS), decision trees (DT), Bayesian method, associative rules (AR) and swarm intelligence (SI). Note that hybrid techniques are not considered in this analysis. Each of these techniques has disadvantages that can be eliminated or reduced if combined with other techniques.

Table 6. Comparison of machine learning techniques

	NN	EC	FL	SVM	CBR	RS	DT	Bayes	AR	SI
Main purpose (problems solved)										
Classification	☑			☑	☑	☑	☑	☑		
Regression				☑	☑		☑			
Clustering	☑			☑[7]						
Forecasting	☑			☑	☑		☑	☑		
Rule extraction	☑								☑	
Expert knowledge integration			☑			☑				
Optimization		☑								☑
Feature selection		☑				☑		☑		☑
Best suitable for										
Small datasets				☑	☑					
Large datasets							☑		☑	
Both	☑	☑	☑			☑		☑		☑
Special characteristics										
"if-then" rules			☑			☑	☑		☑	
Complex configuration	☑	☑	☑	☑						
Local minimas	☑			☑						
Complexity in interpretability	☑	☑		☑						

Source: created by the author.

[7] Ben-Hur et al. [24] also proposed SVM clustering technique; however, it is not discussed in this work.

2.4. Credit risk evaluation – main concepts and techniques

This section reviews credit risk problem and its context for analysis. It gives a survey of various risk types related to business and financial domain which can be used to evaluate performance or financial capabilities of particular subject. It also shortly describes main concepts, definitions and techniques used in credit risk evaluation.

2.4.1. Financial risk definition and classification

Financial risk evaluation is one of the most important issues for both financial analyst and investor whoh seeks to invest his money into particular company or its securities. There are two measures which are referred to risk evaluation – risk amount (the size of possible loss) and quality (probability of obtaining possible loss). Risk quantity can be limited by setting limits to the risks taken; the quality can be evaluated using credit ratings.

Banks or financial institutions are often used as case studies as they are highly influenced by various risks, such as market risk, operational risk which is minimized or at least limited using official regulations (standards), credit risk experienced by issuing credits and manipulating large flows of money. Smaller ratio of share capital to liabilities means that any loss or income will highly influence the value of bank shareholders' property, thus precise credit risk evaluation is necessary to evaluate large amount of liabilities which have to be accepted to increase bank shareholder's profit. Thus risk management task can be viewed as profit maximization and risk minimization using acceptable risk constraints given by the shareholders. The place of this problem in functional hierarchy of financial institution is illustrated in Figure 3 which visualizes relations of functions and processes related to the analysed problem and credit risk management. The hierarchical connection between functions and subfunctions is represented by an aggregation relationship. The diagram identifies that management of processes and problems related to credit risk are influential in both loan giving and bank policy formation activities.

Credit risk is one of the most important risks in banking activities as loan portfolio usually contains the largest part of bank assets, and potential loss on this portfolio results in large potential loss for the whole bank. Therefore, credit risk can be defined as a loss suffered by a bank when a counterparty cannot fulfil its' obligations [125].

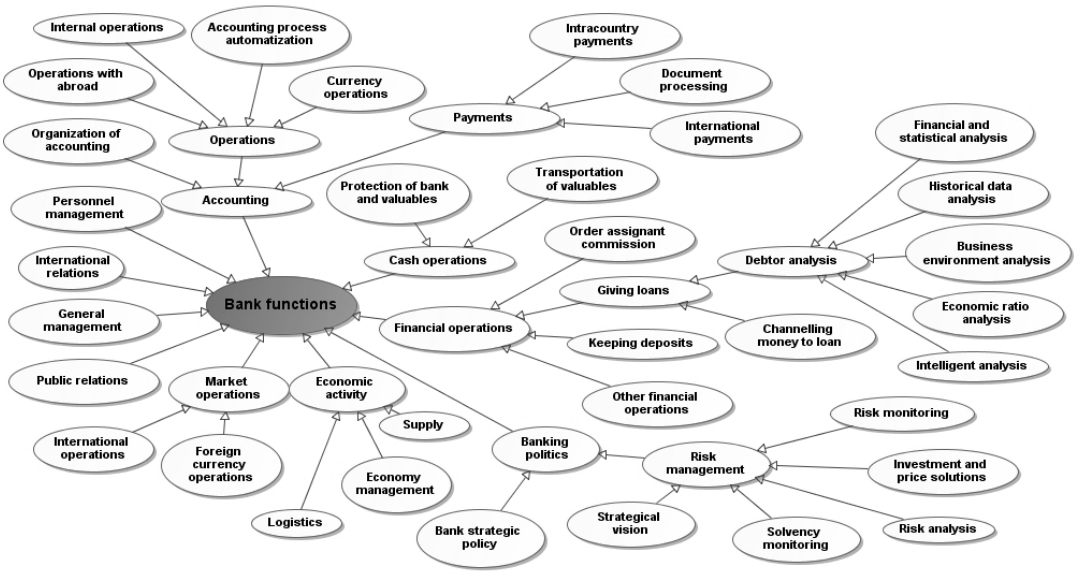


Figure 3. Financial institution function diagram

Credit risk can occur in such fields:

- return of loans (debts);
- return of nominal value of debt instruments (e.g. obligations);
- collecting loans interest or obligation coupon payments (for particular period of time);
- accepting/issuing deposits;
- financial derivatives (e.g., options, futures and etc.)

It is necessary to define and precisely evaluate every risk related to the debtor as well as macroeconomic, legal and other environments to properly assess the risk which has to be taken by the creditor. Evaluation of such risks uses different evaluation methodologies and techniques, both quantitative and qualitative, or even using expert techniques. Appendix G summarizes these risks together with their mostly frequent evaluation techniques. As these risks might be highly correlated among themselves it is also important to evaluate all the possible factors, although limited access to data or its availability is a serious constraint or this task. Such groups of risks can be marked as important:

- Legal and economic environment risks which are related to the legal, economic and money policies performed by government institutions. It is important to properly become prepared for implementation of new legislations to minimize loss, reduce the possibility of danger for stability and trust to minimum, evaluate posible

loss arising from policy changes and reduce them as much as possible.

- Competition and reputation risks – risks related to competitiveness and prestige. They include possible paper profit which was not obtained by inadequate reaction to actions of competitors in the market, loss of reputation because of illegal financial situations or privacy infringements.
- Technology-related risks – risks, related to personnel, hardware, technology and infrastructure performance (errors, inefficiency, incapability to cope with increased load of services, defects and level of technology assimilation), also lack of regulations of uncommon or critical situations or lack of preparing to cope with them.
- Management and strategy related risks, arising from inadequate business strategy, development direction, inappropriate level of evaluation complexity of economical environment..
- Expense and liquidity related – risks which are related to changes to financial assets, fluctuations of assets and liabilities structure, which can result to reduction in profit and/or current value of assets and liabilities. Capital risk is directly related to capital as primary tool to redeem loss, insufficient amount of capital might mean loss of trust in financial institution or performance stability. In case of liquidity risk there is danger that bank will not have sufficient amount of liquid funds to cover its liabilities with the smallest amount of expense. These risks influence financial performance of institution more than any others and directly influence risk limiting constraints.
- Risks which are related to financial income – interest rate risk appears when, during the change of interest rates, income from interest reduces or expenses for interest become larger than income. Management of such risk is especially important for banks as income and expenses from interest are the largest part of their income and expenses. Currency risk means that bank may experience loss from foreign currency exchange rates; this is especially important for banks which perform arbitrage. Trade risk may be influenced by portfolio value reduced by market prices and change of exchange rates.

Every risk contains two aspects: incredibility with danger to lose invested money and uncertainty. Credit quality evaluation process is usually referred as *credit analysis* and covers both quantitative and less formal techniques; people who perform

this process are referred as *credit analysts*. As credit risk measurement is influenced by different factors, there are different techniques to analyse each components. According to 5-C rule [13, 188] the analyst analyzes five key factors (*Character, Capital, Capacity, Collateral, Cycle (or Economic) Conditions*), gives them subjective weights and makes a credit decision. Internal ratings-based (IRB) model [18, 188] is another approach requiring establishing internal ratings model to classify the credit risk exposure of each loan issuing activity. Essential components are [188]:

1. The internal ratings model for classification of the obligation.

2. Risk components - probability of default (PD) and exposure at default (EAD) for the foundation model and PD, EAD, Loss Given Default (LGD) and maturity (M) for the advanced model.

3. Risk weight function that uses the risk components to calculate the risk weights.

4. Requirements for implementation of this model (e.g.., data availability) with supervisory review of compliance with these requirements.

Such risk components are required for implementation of this model:

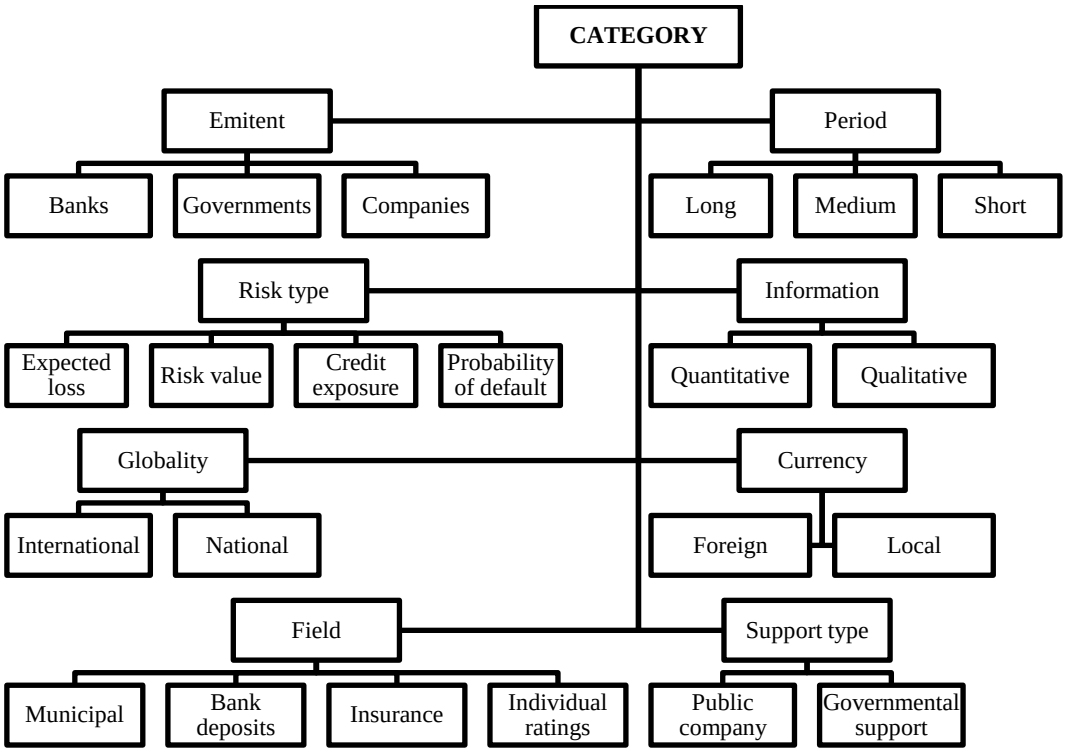
- *Probability Default* (PD), also referred as *default risk* [222] – probability that other counterparty will default during their lifetime or in some particular period (e.g, year) [18].
- *Exposure at Default (EAD),* also referred as *exposure risk* [222] – the expected amount of exposure at the time when a counterparty defaults [18].
- *Recovery Rate* – describes the part of lost credit which can be recovered during default using bankruptcy procedure or other way of payment.
- *Loss Given Default (LGD),* also referred as *loss risk* [222] - determines the amount of loss as a fraction of the exposure in the case of counterparty default [18, 222]. A negative LGD indicates a profit (e.g., due to penalty fees and interest rate) [222].
- *Maturity (M)* - the average maturity of the exposure [18].

Credit ratings can be developed internally or they can be given by external worldwide entities which confirm their reputation and financial capabilities. *Fitch ratings* [83]*, Moody's* [167], *Standard & Poor's* [186], *Dunn & Bradstreet* [55] are examples of such institutions; they also provide data services, together with financial reports. These agencies use ratings as a future reference to evaluate probability of the

emitent to be capable and willing to perform full and timely payments of nominal value and interest.

2.4.2. Scoring and rating

Scoring refers to the use of a numerical tool to rank order cases (people, companies, bonds, countries) according to some real or perceived quality (performance, desirability, sales ability, risk) in order to discriminate between them, and ensure objective and consistent decisions when available data is integrated into a single value that implies some quality, usually related to desirability or suitability [13].



Source: created by the author using (van Gestel, Baesens, 2009)

Figure 4. Credit risk taxonomy, according to van Gestel et al.

According to van Gestel et al., "scoring is related to automatic processing using statistical scoring systems of large customer databases, and ratings result from a manual process that may take days to weeks to complete" [222]. Therefore, scoring can be viewed as decision support automation problem which applies statistical techniques such as classification and/or clustering to form homogenous groups (ratings) according to the level of credit risk value.Note that there are diverse ratings used in various situations and for rating of different subjects. This is illustrated by taxonomy based on van Gestel and Baesens book in Figure 4. It also shows that both

debtors and instruments can be rated. Evaluation in rating process is performed in stages during which separate evaluations are obtained which can be used not only during credit issuing stage but also in monitoring and evaluation during the whole lifecycle of the loan – from initial evaluation to returning money. According to van Gestel et al., There are 4 main stages of loan lifecycle - pre-application (aimed at marketing in order to attract new clients), application, performance and collection [222]. *D*uring performance stage, client's risk, ratios which deflect the behaviour of the client as well as his financial condition, are monitored and evaluated to evaluate changes in his business and realistic possibilities to get money back. Such scores are developed in this stage:

- Performance score to evaluate clients risk during the performance stage and monitor loan portfolio changes and loss;
- Behavioural score to evaluate risk according to changes in client's behaviour;
- Early warning score, used to identify early symptoms of client's critical situation;
- Retention score, which seeks to identify possibility to close accounts or limit their usage as well as avoid machinations;

Collection is final stage of the whole process; if the loan is successfuly refunded, the client can be evaluated as solvent which might possibly influence his rating; or his credit rating might be reduced, and the debt might be exacted. This is also highly influential to future credit rating.

Most of companies and municipal obligations have ratings, except some specific nonrated bonds (e.g., bank obligations). An obligation is defined as a bond with a condition annexed, and a penalty for nonfulfillment. In a larger sense, it is an acknowledgment of a duty to pay a certain sum of money in the future (*Merriam-Webster* dictionary). Therefore, an obligation is an agreement by which an investor gives money at the moment to a debtor who agrees to pay the debt in the future at once or by several distributed payments. This term is more generic and more related to the problem discussed in this work therefore it will be used in this work. Diverse amount of variables can be used for credit risk evaluation – most of scoring and rating models described in this section use up to 10 variables, although modern models can contain up to 30 variables and more. The survey of Velido et al. [227] shows that

most neural network researches used 15 to 25 ratios, while Lee et al. used 41 variables. Larger number of variables enables development of more complex risk evaluation model which might result in more precise credit risk evaluation; therefore, the possibility to access and obtain all necessary data needed reduces as the needed dataset becomes larger in terms of variables. Variable selection performed in various ways, either automatically, using statistical feature selection techniques, or manually, based on experience of analysts. On the whole, according to van Gestel et al., there can be 20 groups of financial ratios used in credit risk research, with 10 groups directly related to credit risk of companies [222]. These groups are:

- *Profitability (earnings)* – these ratios are one of the main financial groups of ratios which describe performance effectiveness as well its capabilities to generate income and keep financial stability. Such ratios, as well as leverage ratios, are important to organizational entities (companies, insurance and financial institutions);
- *Financial leverage* – these ratios describe the liability proportion; the bigger are the liabilities, the higher is the risk.
- *Debt ratios* – important for analysis of all debtors, which directly describes its financial situation and possibility to cover the liabilities in the future. Negative ratios negatively influence rating of the debtor.
- *Growth* – this is more important to companies and insurance companies which have goals to diversify, develop and produce new products and services, expand themselves and obtain profit. As profit and positive balance is priority for banking institutions such ratios can be also used in their evaluation.
- *Liquidity* – directly influences the possibilities to get money back in case the activities of a company are terminated.
- *Activity* – these ratios, like profitability and leverage ratios, are related to the activities bet are more oriented at management and relations with suppliers, e.g., large amount of supply might mean small realization rate as well as inefficient sales management or satiated market.
- *Size* – subject size can be described in diverse ways; the size of personnel, amount of sales or assets and etc. Large companies are considered as more stable as they produce more diverse set of products or services which makes it easier for to react to changes in the market. The number of citizens or amount of taxes might be used as size measures for sovereign ratings.

- *Purpose* – enables analysis of credit dependence from their given pupose and identify how often and which debtors overestimated their financial capabilities to cover their liabilities. Can be used in various levels (individual, company, government, country) if these entities provide such data.
- *Debt history* – one of the most important indicators showing how reliable the debtor; can be used in all levels.
- *Management* – such ratios are important for all organizations, but especially for financial institutions as they reflect capabilities to manage costs and possible loss, as well as efficient usage of resources.

There are two types of variables by their nature:

- *Quantitative* (also referred as *numerical* or *interval*) – variables which can be measured in a fixed measurement scale. Most of primary and derivative financial ratios belong to this group. These type of ratios are less subjective because of their nature and possibility to be measured. Kan defines two scales (interval and ratio) which can be used to represent numerical variables. Ratio scale has an arbitrary initial point (e.g., zero value) which can represent the lowest point. An interval scale indicates the exact differences between measurement points and can use addition and subtraction operations while all mathematical operations can be applied to ratio scale, including division and multiplication [124]. Financial relative ratios are a good example of ratio scale; therefore, they are often used in classification tasks, as it may help to avoid large outlier values which might appear in interval scales.
- *Qualitative* (also referred as *nominal* or *categorical*) – such variables have only fixed, previously defined values and are usually used to express expert evaluations thus sometimes their values can be subjective. Some examples of such variables are marketing strategy, planning level, personnel qualification, information confidence level and etc. According to definition of Kan (2004), they are used to sort attributes into categories with respect to a certain attribute and have two key requirements - jointly exhaustive and mutually exclusive. Mutually exclusive means a subject can be classified into one and only one category. Jointly exhaustive means that all categories together should cover all possible categories of the attribute [124]. Kan (2004) also describes ordinal scale used in software quality measurement which can naturally also be used to express financial categorical variables for risk measurement (e.g., 1 = “completely insolvent”, 2 = partially insolvent”, 3 = “neutral”,

4 = “solvent”, and 5 = “completely solvent”). Similar scale is often used to map categorical variables to numeric; therefore, it is convenient for classification problem formulations.

Similar criterias are used by other rating companies such as *Standard & Poor’s* and etc. As well as legal entities they also rate countries (*sovereign ratings*) which are used to evaluate debt emissions of their governments in local and foreign currency and may highly influence ratings of legal entities in these countries.

Table 7. Company size vs data used for evaluation

Company size	Market prices	Judgemental assessment	Financial statements	Payment history	Personal assessment
Very large	✓	✓	✓		
Large		✓	✓		
Middle		✓	✓	✓	
Small			✓	✓	✓
Very small				✓	✓

Source: R. Anderson. The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation.

Anderson [13] provided a table of data compulsory for assessment of various companies by their size. He also states that for small and very small companies, financial statements may be either unavailable or unreliable (out of date, poor accounting/auditing, or a lie factor). Instead, focus shifts heavily towards obligors’ payment histories. Primary financial data is important for middle companies while judgemental evaluation of large and very large companies usually is dominating factor (although it does not mean that this data is not important). Economic situation in these countries, economic cycle and other external factors influence absolute risk level, for example, during financial crisis the probability of default increases even for companies with highest ratings. Also, as mentioned before, external factors such as political, economic, commodity prices also influence general evaluation. Multistage rating systems described in different sources such as [222] can be used to combine several rating systems as well as refine it with additional rating information, such as country ratings, parental company ratings, branch ratings, expert information that can override final company evaluation. Thus other type of information (economic, social, political) cane be included in the final model as well. Examples of such information are:

- *Macroeconomic ratios and level of social development* which represent overall economic structure and performance of a country such as GDP (Gross Domestic Product), production, revenues, consumption, investment, savings, economic growth, balance of payments, gross capital formation, level of taxes and payments (for sovereigns), tax or price level such as average tax rate (for governments). Examples of social development level are ratios of demographics, revenues distribution, health and education, indicators such as GDP per capita, unemployment rate, Gini coefficient, human development index, life expectancy, health expenditure, adult literacy, level of poverty. They are important for sovereign ratings as well as governments as they are the main criterias which describe their financial situation. They can also be used rating foreign companies if financial situation of the foreign country might directly or indirectly influence their performance. For e.g., a company in a country with unstable political or economical situation might suddenly become insolvent because of high inflation which would reduce the actual value of its assets.
- *Markets and their management policy* – policy of foreign countries directly influences financial performance of these countries and institutions which are situated there.
- *Stability* – the abscence or low risk of conflicts as well as general stability also stabilizes financial performance of entities which are situated in that country. Opposite results might indicate high possibility of financial situation change.
- *Politics* – evaluation of this domain is quite subjective and usually performed by various experts; therefore, its factors, such as political regime, level of liberty and freedom, political environiment, level of corruption directly influence general financial and economic environment and thus it may influence the performance of various entities from that country. However, these factors are especially important for evaluation of government and foreign countries.

Therefore, this work is limited to financial performance and application evaluation ratios.

2.4.3. Credit scoring models

During credit scoring process, credit score is calculated from information obtained about debtor using a particular model. Many techniques for scoring have been developed using various techniques; latest research based on machine learning

techniques is summarized and reviewed in previous sections. Therefore, the most popular and widely adopted models are based on multiple discriminant analysis (MDA) and logit analysis. The earliest works in credit risk research date to 1968 when Altman applied multiple discriminant analysis to develop his Z-Score model [7], obtaining 96% and 79% accuracy in two different samples. Altman continued to use MDA in his further works, developing ZETA model [8], which, according to its later work [9], demonstrates improved accuracy over his original Z-Score model and has demonstrated higher accuracy over a longer period of time. MDA was applied by other researchers as well to develop their own models [67,207] or to improve or discuss existing ones [92,93]. Springate [200] also developed his model using step-wise MDA and 4 ratios selecting them from 19 ratios and obtaining an accuracy rate of 92.5% using the 40 companies; later 83.3% and 88% accuracy rates were reported after testing it with other samples [187].

Other statistical techniques such as logit analysis, probit analysis, hazard models later were also applied credit risk scoring and bankruptcy prediction domain. Ohlson [171] applied logit analysis reporting accuracy of 96.12%, 95.55% and 92.84% for prediction within one year, two years and one or two years respectively. Begley et al. [21] showed that Ohlson's model might perform better than Altman's original and improved models; however, some researchers argue that they did not evaluate the model's sensitivity to industry classification, perform inferential tests of the model's classification accuracies, and evaluate the model's sensitivity to financial distress situations [92]. Shumway [197] developed his model using hazard analysis using the same predictors as in original Altman model. However, Shumway assumes that risk for bankruptcy changes through time and its health depends on the latest financial data of the company and its age thus his model includes an additional component representing the time spent by a firm in the healthy group. Zmijewski [267] used two samples of 840 companies (40 of them were bankrupt companies) for training and prediction purpose using probit and maximum likelihood techniques and obtained 72% accuracy.

Despite many recent techniques, MDA is still widely used as evaluators in various institutions. The models developed using these techniques might be expressed in a general regression form

$$Z = w_0 + \sum_{i=1}^{n} w_i x_i \tag{2.41}$$

Altman's Z-Score (Altman, 1968) is a linear model originally targeted at publicly held manufacturers which may be used to predict the probability that a firm will go into bankruptcy within two years using ratios from balance and income statements:

$$\begin{aligned} . Z = \; & 1{,}2 * (\text{Working capital/Total assets}) + \\ & 1{,}4 * (\text{Retained earnings/Total assets}) + \\ & 3{,}3 * (\text{Earnings before interest and taxes/Total assets}) + \\ & 0{,}6 * (\text{Book value of Equity/Book value of total liabilities}) + \\ & 0{,}999 * (\text{Net sales/Total assets}) \end{aligned} \tag{2.42}$$

If Z > 3 then company is considered as healthy, 2.7 < Z < 2.99 – as non-bankrupt ("gray" zone), Z < 1.79 – bankrupt.

Altman's Z-Score for private companies (Altman, 2000) is a modification of original Z-Score model adapted for companies which do not trade their stock in the market:

$$\begin{aligned} . Z = \; & 0{,}717* (\text{Working capital/Total assets}) + \\ & 0{,}847 * (\text{Retained earnings/Total assets}) + \\ & 3{,}107* (\text{Earnings before interest and taxes/Total assets}) + \\ & 0{,}420* (\text{Book value of Equity/Book value of total liabilities}) + \\ & 0{,}998 * (\text{Net sales/Total assets}) \end{aligned} \tag{2.43}$$

Altman's Z-Score for non-manufacturing companies (Altman, 2000) is a modification of original Z-Score model. The ratio of net sales and total assets is excluded as service oriented companies generate large sales using relatively small amount of assets:

$$\begin{aligned} Z = \; & 6{,}56* (\text{Working capital/Total assets}) + \\ & 3{,}26 * (\text{Retained earnings/Total assets}) + \\ & 6{,}72* (\text{Earnings before interest and taxes/Total assets}) + \\ & 1{,}05* (\text{Book value of Equity/Book value of total liabilities}) \end{aligned} \tag{2.44}$$

Springate's model (Springate, 1978) is also used to evaluate the company's probability of bankruptcy and is very similar in it's form to Altman's model. The company is classified as "failed" if Z < 0.862.

$$\begin{aligned} Z = \; & 1{,}03* (\text{Working capital/Total assets}) + \\ & 3{,}07 * (\text{Earnings before interest and taxes/Total assets}) + \\ & 0{,}66* (\text{Earnings before interest and taxes/Current liabilities}) + \\ & 0{,}4* (\text{Net sales/Total assets}) \end{aligned} \tag{2.45}$$

Zmijewski's model (Zmijewski, 1984), developed using probit analysis, is also used to evaluate the company's probability of bankruptcy.

$$Z = -4{,}336 - 4{,}513 * (\text{Net Income} / \text{Total Assets}) + 5{,}679 * (\text{Total Debt} / \text{Total Assets}) + 0{,}004 * (\text{Current Assets} / \text{Current Liabilities}) \quad (2.46)$$

Shumway's model (Shumway, 2001) is a discrete-time hazard model with a logit model estimation program that uses combination of accounting ratios and market-driven variables to produce more accurate out-of-sample forecasts than alternative models [197]. Note that the same variables as in Zmijewski's model are used. The company is classified as "failed" if s*core* < 0.

$$. Z = -7.811 - 6.307 * (\text{Net Income} / \text{Total Assets}) + 4.068 * (\text{Total Debt} / \text{Total Assets}) - 0{,}158 * (\text{Current Assets} / \text{Current Liabilities}) \quad (2.47)$$

Some of these models are summarized in Table 8; this is useful in their comparison.

Table 8. MDA analysis based models

	Altman	**Altman B**	**Springate**	**Zmijewski**	**Shumway**
w0				-4.336	-7.811
w1	1.2	6.56	1.03	-4.513	-6.307
x1	Working capital/ Total assets	Working capital/ Total assets	Working Capital/ Total Assets	Net Income / Total Assets	Net Income/ Total Assets
w2	1.4	3.26	3.07	5.679	4,068
x2	Retained earnings/ Total assets	Retained earnings/ Total assets	Net Profit before Interest and Taxes/ Total Assets	Total Debt / Total Assets	Total Debt / Total Assets
w3	3.3	6.72	0.66	0.004	-0.158
x3	Earnings before interest and taxes/ Total assets	Earnings before interest and taxes/ Total assets	Net Profit before Taxes/ Current Liabilities	Current Assets / Current Liabilities	Current Assets / Current Liabilities
w4	0.6	1.05	0.4		
x4	Book value of Equity/ Book value of total liabilities	Book value of Equity/ Book value of total liabilities	Sales/ Total Assets		
w5	0.999				
x5	Net sales/ Total assets				
Eval	Z>3 – healthy 2.7<Z<2.99 – non-bankrupt Z <1.79 – bankrupt		Z < 0.862 – bankrupt	Z < 0 - bankrupt	Z < 0 - bankrupt
Type	MDA		MDA	Probit	logit

Source: created by author.

Ohlson's model (Ohlson, 1980), referred as O-score, is developed using a technique based on logistic transfotrmations and used to estimate of the probability of failure [171]:

$$O = -1{,}32 - 0{,}407 * (\text{MASSET} / \text{Consumer Price Index}) + 6{,}03 * \text{leverage ratio} - 1{,}43 *(\text{working capital} / \text{market assets}) + 0{,}076 * (\text{Current Liabilities} / \text{Current Assets}) - \quad (2.48)$$

1,72 (Total Liabilities > Total Assets ? 1 : Total Liabilities / Total Assets) –*
2,37 (Net Income /Total Assets) –*
1,83 (Cashflow from Operations/ Total Liabilities) +*
0,285 (net income for the last two years < 0 ? 1:0) -*
$0{,}521 * \frac{NI_t - NI_{t-1}}{|NI_t| + |NI_{t-1}|}$

where *MASSET* is market assets defined as book asset with book equity replaced by market equity. It can be calculated as *total liabilities + Market Equity.*Leverage ratio is defined as as the book value of debt divided by *MASSET*. NI_t is net icome for current quarter/year, NI_{t-1} – the previous quarter/year. The final probability evaluation is obtained using logistic transformation $\frac{e^{O-Score}}{1+e^{O-Score}}$; if the resulting probabilty is over 0.5 then company is classified as „failed".

Fulmer's model (Fulmer, 1984) is also based on step-wise multiple discriminate analysis:

$$
\begin{aligned}
Z = \; & 5{,}528 * (\text{Retained Earning/Total Assets}) + \\
& 0{,}212 * (\text{Sales / Total Assets}) + \\
& 0{,}073 * (\text{Earnings before interest and taxes/Equity}) + \\
& 1{,}270 * (\text{Cash Flow/Total Debt}) - \\
& 0{,}120 * (\text{Debt/Total Assets}) + \\
& 2{,}335 * (\text{Current Liabilities/Total Assets}) + \\
& 0{,}575 * (\text{Log Tangible Total Assets}) - \\
& 1{,}083 * (\text{Working Capital/Total Debt}) + \\
& 0{,}894 * (\text{Log Earnings before interest and taxes/Interest}) - 6{,}075
\end{aligned}
\tag{2.49}
$$

If Z < 0 then company is considered as unhealthy.

However there are many other models used in other countries developed using MDA, logit analysis, neural networks and other techniques. Such research is summarized by Bellovary et al. [22]. Altman and Narayanan also performed a survey of the works by academics and practitioners in 21 countries, including developed countries such as Japan, Switzerland, Germany, England, France, Canada, The, Netherlands, Spain, Italy, Australia and Greece, as well as countries which they listed as developing [10].

2.4.4. Modern models for credit risk evaluation

Besides discriminatory credit risk evaluation techniques based on classification there are other modern techniques aimed at default modeling. These techniques are outside of the scope of this work, thus only main concepts and examples are provided in this work together with references. According to Allen [5],

Elizalde [76,77] and etc., such techniques can be grouped to structural (which try to determine the time of the default), reduced form models which model the intensity of default as a jump-process, and the time of its jump is the time of default [76], Value at Risk (VaR) and mortality rate models. Merton's model (Merton, 1974) was the first first structural model, based on geometric Brownian motion driven modeling asset value; this technique is based on Black and Scholes (1973) option pricing technique [76]. Commercial solutions such as KMW/Moody's (Merton OPM), KMV's Credit Manager and Moody's RiscCalc, are based on this approach [5]. The former, is based on three step evaluation used to calculate Expected Default Frequency (EDF). Black and Cox (1976) developed first passage models (FPM) assuming that default might take any time after it reaches lower barrier. Elizalde stated that its largerst drawback is its analytical complexity increased even more if stochastic interest rates or endogenous default thresholds are considered [76]. Other referred structural models are Longstaff and Schwartz (1995) and Collin-Dufresne and Goldstein (2001) models; Huang et al. give their specification analysis together with Merton's and Black and Cox approaches [115].

Table 9. Comparison of scoring techniques

Criteria	**Statistical techniques**	**Machine learning techniques**	**Structural models**
Techniques	Univariate analysis MDA Probit/logit analysis	Neural networks and etc. Expert systems Hybrid models	Merton model Gambler's ruin
Applicability	+	+	- (Limited to companies)
Empirical validation	+	+	+
Statistical validation	+	- (No weights that can be statistically tested)	N/A (Parameters are derived from financial theory and cannot be tested statistically)
Economic validation	+ (Expected by experts and obtained weights can be compared)	+ (The impact can be estimated using sensitivity analysis)	+ (These models are derived from financial theory)
Market reference	+ (Riskcalc used by Moody's, etc.)	+ (Many researches applied on real data)	+ (KMV model)

Source: adopted from Balthazar L. From Basel 1 to Basel 3: The Integration of State-of-the-Art Risk Modeling in Banking Regulation

Balthazar compared statistical, inductive (machine learning) and structural models. Table 9 gives this comparison. It can be seen that all these models can be validated empirically (using out-of-sample, out-of-time tests); the main disadvantage of machine learning techniques is lack of their capabilities of statistical validation.

Reduced form models use intensity based techniques to calculate stochastic risk levels. According to Elizalde [77], in structural risk models, predictability of default implies zero short-term credit spreads for the firrm's debt, inconsistent with the short-term spreads seen in practice. Reduced form models overcome this limitation specifying an exogenous default intensity which makes default an unpredictable event. However, occurance of default is not directly related to firm's credit quality [77]. KPMG's Loan Analysis System and Kamakura's Risk Manager are based on this approach [5].

Value at Risk (VaR) approach is widely used approach for market risk modeling and measurement. The main principle of VaR is valuation of financial instrument (position) using pricing model and simulation of underlying risk parameters such as interest rates, exchange rates, equities values, implied volatilities and etc. using statistical distributions and correlation between the different risk factors to generate correlated pseudo-random outcomes (parametric VaR) or using historical time series and selecting randomly observations in the datasets collected (historical VAR). Then these outcomes of the risk drivers are injected in the pricing models and all positions are re-evaluated; these simulations are performed particular (usually) number of times to form a distribution of potential future values [18]. CreditMetrics and Algorithmics Mark-to-Future are widely known examples of modern models based on such techniques [5]. Allen also describes mortality rate models, with Credit Risk Plus model as an example, as well as gives comparative analysis of these approaches according to data requirements, correlation and volatility of credit events, interest rates constant or stochastic, risk classification and etc. [5].

2.5. Generic framework for hybrid model development

The development process of hybrid technique includes various machine learning, statistical and mathematical techniques and algorithms, discussed in Section 2. It is known and shown in various papers that combination of several techniques enables minimization or expose of their drawbacks. Therefore, development of efficient intelligent technique can take several steps which can be formalized in a general structure.

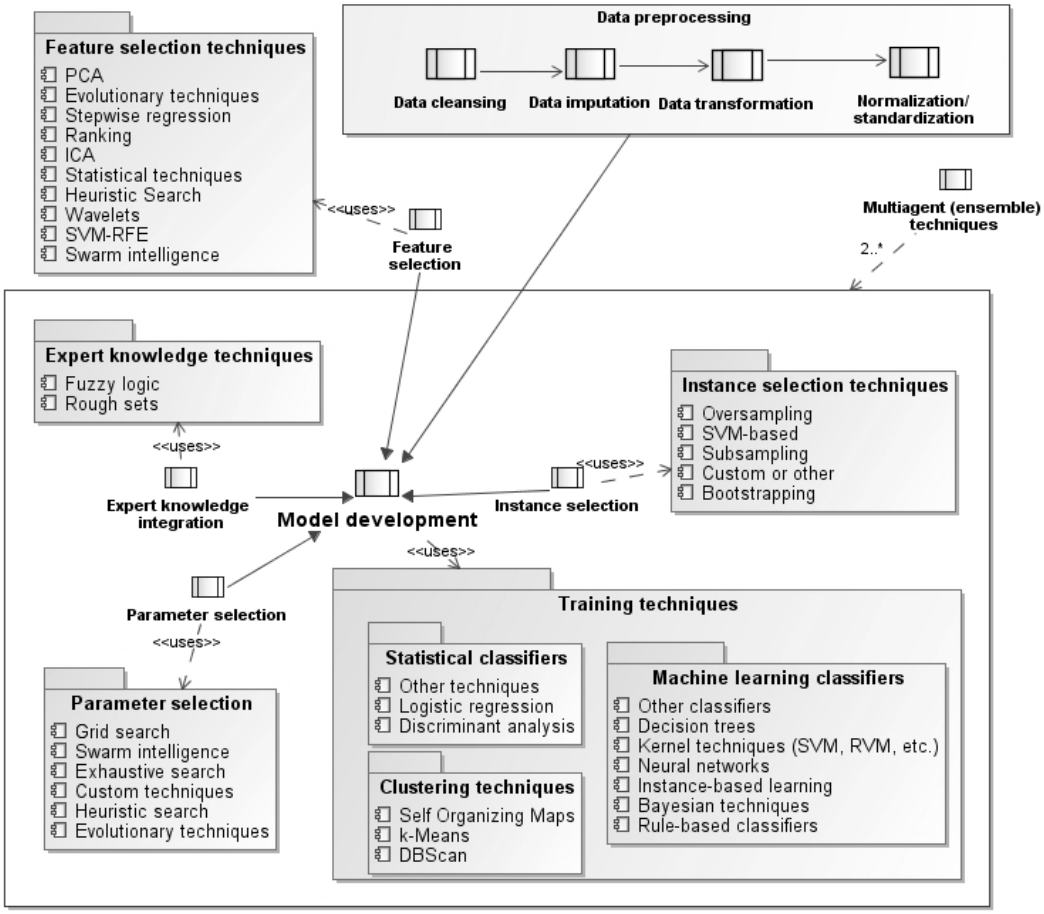


Figure 5. Generalized hybrid model development framework

Such framework developed according to research discussed in Section 2 is shown in Figure 5; although it is oriented at classification model development, it can be extended or adapted for clustering tasks as well. Model development process can be viewed as a composition of several processes (tasks):

$$Model_{dev} = <Data\ preprocessing, Feature\ selection, Instance\ selection, Parameter\ selection, Expert\ knowledge\ integration> \quad (2.50)$$

This structure can be extended for ensemble model development (often viewed as multiagent learning) as well; this relation is modelled as 2..*, i.e., ensemble model must consist of at least 2 different classifiers (agents). Note that data preprocessing and feature selection task are excluded from the range of general model development domain mainly to make this framework suitable for ensemble learning as each agent in the ensemble has to use the same data for training although it may use different strategies for instance and parameter selection. This framework is also consistent with the six-step scheme for scoring model development described by Balthazar [18] and will be used as basis for further development of classification

methods.

Therefore such steps are described in the framework:

- Data processing – preparing data for models. It can include such processes as data normalization, data standardization or data cleansing, imputation, transformation and/or normalization. These tasks are not always necessary but might improve performance and therefore are considered as important, as they might reduce "noise", identify important relations or trends. The attributes might be transformed using logarithmic, log-linear or other similar operations. Differences between data which are not very significant therefore might have influence although removing such data might result in improved performance as trained model might be targeted at larger and more significant changes. However, the opposite is also possible – insignificant changes might be the results of forecasting. Thus it is important for the analyst to know exact purpose and possible outcome of the developed model.
- Feature selection – selection of essential attributes that have the biggest influence (in case of credit risk – financial ratios) by using statistical, mathematical or machine learning methods or selection of principal components by using PCA or ICA.
- Optimization of models parameters by using heuristic techniques described in Section 1.
- Training hybrid model using machine learning and artificial intelligence methods including fuzzy logic and rough sets and/or ensemble techniques. Trained model should be capable to recognize new data structures and instances by classifying them to one of the set of classes.
- Evaluation of this model by benchmarking it with similar methods or hybrid models to evaluate its precision, accuracy and performance. Testing is also needed to evaluate generalization abilities
- Development of intelligent agent which will be using the model created in first four steps. This step is optional and related to implementation of model after it is validated that the developed system has fulfilled its tasks and obtained desirable results.

2.6. Decision support systems and expert systems for credit risk evaluation

First artificial intelligence based systems were developed in the 6^{th} decade;

they allowed integrating expert knowledge from various domains to solve various tasks. First artificial intelligence systems were developed in 1950s more on scientific basis, to solve such tasks as proving theorems (*Logic Theorist* by Newell et al., *Geometry Theorem Prover* by Gelernter), general problem solving (*General Problem Solver* by Newell and Simon, *ANALOGY* by Evans), communication and question answering (*ELIZA* by Weizenbaum)[85]. Expert systems emerged in the 70s and 80s, with *DENDRAL*, solving task of mapping the structure of complex organic chemicals from data gathered by mass spectrometers (Feigenbaum, Lindsey) and *MYCIN* for infection diagnosis (Shortliffe, 1976) as pioneers of such systems[85]. DEC developed first commercial expert system *XCON/R1* for composition of computer systems from various components. As this proved to lead to financial gain, other systems, such as *PROSPECTOR* for mineral monitoring, *WILLARD* to forecast thunderstorms, *FOLIO* to analyse investment portfolios, were developed [85], thus enabling expert-based system usage in solution of real world problems. This is also important in financial domain – Shao et al [196] report on usage of expert systems in UK banking sector in 1983-1985, namely Barclay, Midland banks (over 13 banks, most of which were kept anoynymous). Development of intelligent DSS and related research is still relevant, as a lot of research is still performed; the scope in this work is limited mainly to financial risk domain.

2.6.1. The definition of decision support systems and expert systems

Both of these decision support and expert systems can be defined as systems oriented at decision support at individual, organizational or government levels, using knowledge base with models, patterns and structured knowledge, together with external data. Although such systems also share similar architectural structure and goals, they differ in some particular aspects; expert systems are mostly referred to rule-based systems, while DSS definition usually comprises much broader spectrum of various systems. Expert systems are defined in various sources, including dictionaries; some of these definitions are given below:

- “Information systems which consider particular criterias, constraints and possible conclusions, collect information of these criterias and propose the best possible sequence for further actions. Expert systems never forget relevant details and give more precise results than human“ [1];

- "A system that uses human knowledge captured in a computer to solve a problem that ordinarily needs human expertise" (Aronson, Turban 2001; cited by [248]);
- "Typically autonomous problem solving systems used in situations where there is a well-defined problem and expertise needs to be applied to find the appropriate solution" (Aronson, Turban 2001; cited by [20]);
- "A program that uses available information, heuristics, and inference to suggest solutions to problems in a particular discipline" (*American Heritage Dictionary*);
- "A computer program that contains a knowledge base and a set of algorithms or rules that infer new facts from knowledge and from incoming data. The expert system derives its answers by running the knowledge base through an inference engine, a software program that interacts with the user and processes the results from the rules and data in the knowledge base" (*Free On-Line Dictionary of Computing*);
- "An artificial intelligence application that uses a knowledge base of human expertise to aid in solving problems. The degree of problem solving is based on the quality of the data and rules obtained from the human expert. Expert systems are designed to perform at a human expert level. In practice, they will perform both well below and well above that of an individual expert" (*Free On-Line Dictionary of Computing*).
- "A computer system containing a lot of information about one particular subject, so that it can help someone find an answer to a problem" (*Longman Dictionary of Contemporary English*);
- "A computer system which asks questions and gives answers that have been thought of by a human expert" (*Cambridge Advanced Learner's Dictionary*).

These definitions highlight the most important aspects of expert system:

1. It is a computer system – a system developed on a computing platform, although Aronson and Turban do not highlight ir their definitions;

2. It uses human knowledge and expertise which is collected in knowledge base in various forms (rules, formulas, models and etc.);

3. It aims to gives answers at least in the same level of correctness and validity as a human expert, as well as aims to eliminate the possibility of human error.

4. It has such components as knowledge base, inference engine and a set of models (rules, etc.) which form basis of expert system. The structure of expert systems is reviewed in Appendix A.

Usage of first expert systems also helped to identify their main disadvantages – lack of flexibility, resulting in inflexibility for decision making, absence of history experience, complex intercommunication and sophisticated support, complicated development, resulting in thousands or tens of thousands of rules. Besides of these disadvantages, Luger identifies difficulty in capturing the deep knowledge of the problem domain and lack of detail in explanations, especially in solutions logic, problems in solution verification [20]. These problems were fully or partially eliminated after applying machine learning techniques, such as neural networks, which enabled “learning from the past” using historical data, whereas new technologies, such as Web Services, integrated tools and environments to support expert system development were introduced.

Decision support systems (abbr. as DSS) are also widely discussed and defined in various sources. Raynor defines DSS as “data modeling and reporting system that has been structured to answer specific ongoing business questions or issues. It is usually distinguished from classical IS systems by its emphasis on "real-time", or interactive analysis, where the business analyst can use multiple tools on the data to provide answers "now."[177]. Hamilton describes DSS as “a computer system providing both problem solving and communications capabilities for semistructured/unstructured problems” [98]. Thus DSS are systems designed to support tasks and decisions, which may not be easily specified because of large amounts of related information, complex process of deducing such decision and/or rapid change of situation.

Keen describes such goals of DSS [101]:

1. To help to solve structured or semistructured problems for management level (but not to replace the manager himself).

2. To model various alternatives or strategies for solving a particular problem and forecast possible consequences without any implications on operational level;

3. To contribute to the effectiveness of the solution (but not to the productivity) by increasing the possibility of possible positive result. The

concentrated or derived information enables to accept this solution faster, more precisely and more objectively.

4. Such kind of software is important in solving various business problems, especially in financial domain, such as real-time investment, forecasting of stock prices and indices, stock and securities selection and etc. Credit risk evaluation is one of such problems which involve bankruptcy prediction, selection of possible subjects to give a credit to. A lot of work has been done in this field; it is reviewed in further sections.

2.6.2. DSS conformance banking regulation standards

Basel standards are one of the most widely known banking regulation standards; originally developed for G10 countries it also became a standard for banking regulations in other countries. First standard, usually referred to as Basel I, was developed by Basel Committee on Banking Supervision in 1988 with two main objectives: soundness and stability of the international banking system and minimization of competitive inequality between them [18]. It was generally a set of rules designed to apply at international level although domestic banks could apply it in domestic level as well, including minimum capital level equal to 8 percent. This standard defined two classes of capital as well as risk weight of assets, as well as credit conversion factors and risk decomposition. Therefore, as Balthazar states in [18], it had several drawbacks and limitations which could allow bank manipulations such as lack of diversification to various sectors and regions, lack of requirements' flexibility according to type of loans, activities or banks and etc. Most important, it was focused only on credit risk without covering importance of other risks described in Section 2.4.1. Therefore, more complex and flexible regulatory framework had to be implemented.

Basel II standard which initial proposal was released in 2004 was designed to address these issues and support financial stability and integrate better risk management practices. It consists of three main pillars which describe different aspects of financial risk management [18]:

- Pillar 1 – describes solvency ratio derivation and improves weighting of assets from rough estimates to explicit derivation from a standard simplified credit risk model.

- Pillar 2 – describes internal controls and supervisory review, such as requirement for banks to posess internal systems and models for evaluation of their capital requirements in parallel to the regulatory framework. It also denotes integration of other types of risks which are not covered by the Accord (they are described in Section 2.4.1). As Balthazar notes, although this pillar is too flexible, it should oblige regulators and banks to cooperate closely on the evaluation of internal models.
- Pillar 3 – describes disclosure of the risk management reports to the market place with a large set of elements to be published. This gives opportunity for credit analysts, investors or other related marker stakeholders to evaluate bank's financial situation themselves.

Therefore, it can be seen that Basel II standard promotes risk integration (notable that forthcoming Basel III standard provides more strict guidelines for their integration), thus integrated approach for credit risk evaluation IS is necessary to support this view. Design and development of centralized information architecture and warehouse enabling advanced analytics and reporting capabilities as well as integration with information standards becomes important.

Balthazar describes such key requirements of rating systems in conformance to Basel II standard [18]:

- PD and LGD calculation;
- At least seven rating grades for non-defaulted companies (and one for defaulted);
- Consistency across subsidiaries, locations, businesses;
- Transparency to auditors and external parties (risk description and clasification);
- Proved and documented effectiveness, supervision, auditing and correctness of used scoring model and the model itself;
- Integration of all available information, including external ratings by rating agencies;
- Integration of the debtor's solvency despite adverse economic conditions;
- Regular model development validation and performance monitoring performed by an independent unit;

- Documented and justified overrides (cases where credit analysts give another rating than the one issued by a scoring model);
- Constant recording of the data used for rating and default history.

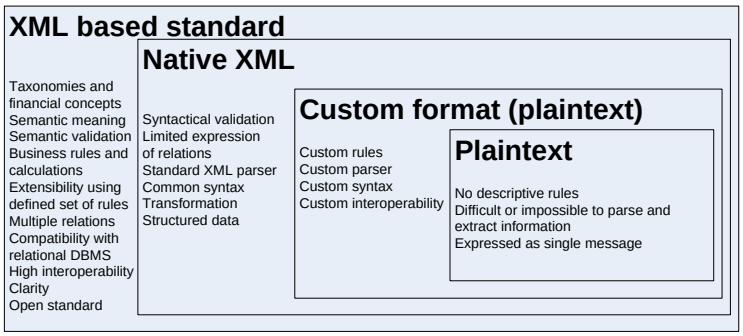
According to Merkevicius [159], there are such requirements for Basel II compatible IS:

- Proper data warehousing infrastructure which contains data from debtors (sector, debtor type. History of internal and external ratings, financial data, risk events), loans or positions (loans type, terms, currency, interest rates and payments, deviations, etc.), financial mortgages (mortgage type, nominal value, market value, currency expression, net value, frequency of reappraisal, terms, other dta, etc.), physical mortgages (ownership references, value, frequency of reappraisal, etc.), details of guarantees and derivative credits (currency data, terms, data of guarantee providers), risk evalutions by various dimensions;
- System integrity and flexibility;
- Supervisory access;
- Corporate positions, which are evaluated according to historical and planned monetary flows, equity structure, income quality, level of leverages and their influence for profitability and cash flows, financial flexibility and access to catipal markets for additional reserves, management experience and competences, company position in sector.
- Consumer positions, realized by implemented credit scoring system;
- Historical data of ratings;
- Segmentation;
- Calculation of PD, EAD, LGD components, risk weights and capital adequacy and monitoring of calculated values.

2.6.3. A review of information exchange standards in financial domain

Financial data standardization has become one of the most important technical criterias to solve financial reporting and exchange issues. Electronic Data Interchange (EDI) standards and technologies such as Extensible Markup Language (XML), XML Schema, created by Microsoft Corporation, querying standards XQuery, XPath, transformation technologies XSL/XSLT/XSL-FO and etc. are often applied to solve such problems. They can be used to implement custom vendor-dependent standards,

as well as to develop widely agreed frameworks which can be used to implement exchange between different systems, using various Enterprise Application Integration (EAI) frameworks and patterns, such as Enterprise Service Bus (ESB), Service Oriented Architecture (SOA) and others. Transitioning to application of these standards is often also a difficult task as it may involve reengineering, restructuring and and redesign of current information systems, business processes or even of the whole IT infrastructure to ensure full compatibility of with these standards, although many vendors adopt their software to comply with such standards enabling fully compatible business systems which use best practices from other companies [89].



Source: Garsva G., Danenas P. XBRL Integration Into Intelligent System For Credit Risk Evaluation

Figure 6. Advantages of XML based standards compared to other representation formats

The paper of Garsva and Danenas [89] gives a detailed survey of such standards used in financial sector, together with their classification compatible to Basel II standard as it involves all three risks – credit risk, market risk and operational field. It identifies the need of compliance and integration capabilities in decision support processes, as well as describes advantages of XML-driven standardization (Figure 6). Yet, the main focus is on eXtensible Business Reporting Language (XBRL) technology which adoption at the moment in financial and banking domain is widely growing. XBRL proposes capabilities of financial information exchange as well as integration of metadata in form business rules to ensure integrity and validation, as well as exploit additional data to extend the dimensionality of information that can be used in modeling. Other XBRL benefits such as reduced number of submissions, better accounting quality and analytical options, integration of best practices, business rules and formulas, format independent representation, automatic access capabilities are also discussed in [89]. As it is stated in this paper interactive data option which offers abilities for real-time analysis can be supported

by regulatory authorities such as SEC, which choose to make data publicly available, e.g., SEC's EDGAR online database that can be accessed using RSS protocol [241].

XBRL is defined by two primary concepts: taxonomy and instance. Taxonomy defines all financial concepts that are used by a particular entity, their inner relationships and internal or external resources; instance can be defined as the list of facts structured as defined in taxonomy [89]. The core of this standard consists of the taxonomy itself and linkbase that defines relationships between elements in order to properly organize the taxonomy content; several types of linkbases are defined which define rules for presentation (taxonomy content organization), calculation (basic validation rules), definition of relationships, multilanguage and references to external documents with additional information of the concepts [89]. XBRL has a modularized structure, i.e., it can be extended with additional modules (, , and modules at the time of analysis in [89]) or custom extensions of taxonomies.

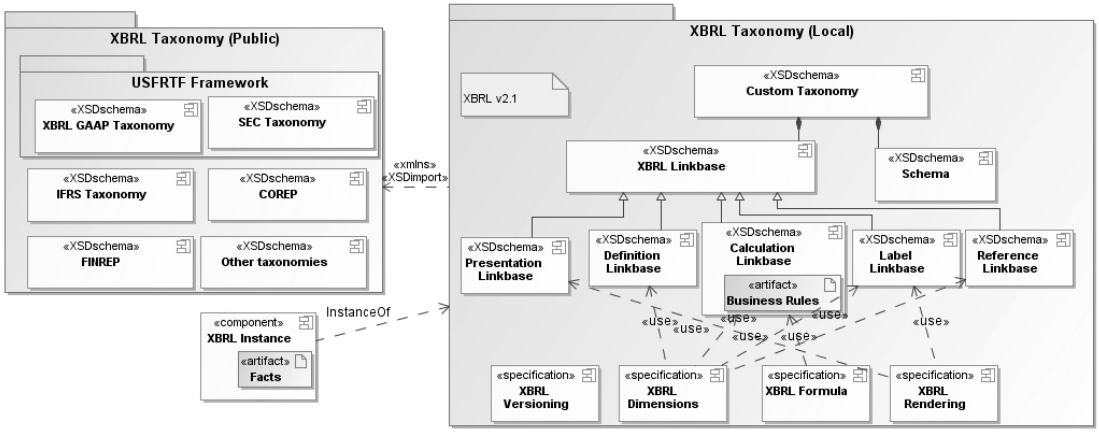
XBRL taxonomy consists of such components:

- XBRL schema, which stores information about taxonomy elements such as an unstructured list of elements and references to linkbase files;
- Linkbase files - provide information about relationships between elements and link them with specified external resources. Five types of linkbase files are commonly referred in various sources; three of them are referred as relationship linkbases (calculations, presentations, and definitions) which provide semantic validation property:
  - Presentation linkbase - provides structured relationships between elements in order to properly organize the taxonomy content (e.g., to arrange hierarchical data);
  - Calculation linkbase - contains definitions of basic instance validation rules.
  - Definition linkbase, which provides different kinds of relations between elements;
  - Label linkbase, which defines labels for different languages.
  - Reference linkbase – provides pointers to external documents describing the concepts defined in taxonomy.

Several extensions are defined, the most important of them are:

- XBRL Dimensions 1.0 [244], defining additional structured contextual information for business facts in a manner similar to „dimension“ concept in OLAP analysis [89];
- XBRL Formula 1.0 [245], providing advanced validation capabilities for instance documents, such as value, existence and consistency assertions using XPath syntax. It also allows definition of new facts (e.g., secondary financial attributes);
- XBRL Rendering (Inline XBRL) [246] specification, which define rendering of XBRL documents. Inline XBRL specifies how XBRL fragments can be embedded in an HTML document, using XBRL tags.
- XBRL Versioning [247] – the main objective of this specification was to provide means to develop versioning reports with sufficient and comprehensive coverage of changes between different versions of taxonomies.

XBRL structure is given as UML package diagram, with inner relationships between specifications modeled as inheritance or aggregation concepts and module connections with particular linkbases presented as <<*use*>> relations (Figure 7). As some taxonomies are used as standards provided by authority regulators (such as US GAAP, used for USA financial reporting, COREP and FINREP created by The Committee of European Banking Supervisors, XBRL-GL) they can be selected as basis for reporting, with custom extensions provided by the reporting entity. Therefore their modeling is consistent with object modeling concepts, such as generalization and inclusion (usage).



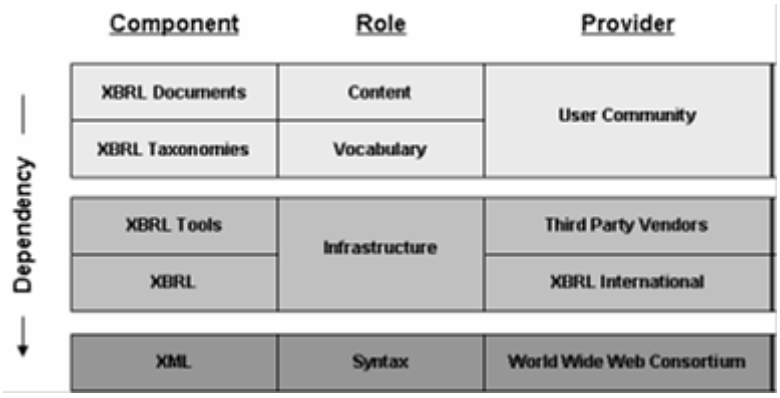
Source: Garsva G., Danenas P. XBRL Integration Into Intelligent System For Credit Risk Evaluation.

Figure 7. XBRL modular structure

Garsva and Danenas also describe SEC’s taxonomy for Nationally Recognized Statistical Rating Organizations (NRSRO) in terms of UML component

diagram which can be used to implement exchange of ratings data; this is an interesting and important option in credit risk modeling tool development.

Different taxonomies are usually defined for different types of filers, e.g., SEC defines different taxonomies for corporative filers (US GAAP taxonomy), investment companies (US Mutual Fund Risk/Return Taxonomy), entities that prepare their financial statements in accordance with Article 6 of Regulation S-X (US Schedule of Investments Taxonomy) or Nationally Recognized Statistical Rating Organizations (RATINGS taxonomy) to comply with its requirements for reporting and data collection in EDGAR system (SEC.gov, cited by [64]).



Source: Financial Information Sharing (FIS) Subcommittee. Transforming Financial Information – Use of XBRL in Federal Financial Management

Figure 8. Layers of XBRL Components

Each regulatory authority may define its own taxonomy for XBRL-based reporting which is more compliant to the needs and financial regulations of the country that it resides in. According to [89, 194], XBRL at the time of writing is or has been implemented or adopted as alternative for financial reporting in Australia, Netherlands, Great Britain, Korea, Belgium, Japan, Singapore, Spain and etc. Another important artefacts, which support XBRL integration into software and process engineering processes, are abstract XBRL Abstract Model 1.0 [242] and XBRL Abstract Model 2.0 [243] specifications (released in 2011 and 2012, respectively) which are consistent with OMG metamodeling guidelines and are referred as abstract XBRL metamodels. XBRL Abstract Model 1.0 is divided into 4 packages: Instance model, Fact model, Concept model and Typing model [242]. Version 2.0 of this model is far more complex than Abstract Model 1.0 and is divided into primary and secondary models. Primary model, besides Instances and Typing packages, also defines Cross Model Elements, Data Dictionary Model, Valid Combinations Model,

Data Points Semantic Grounding, Table Model and Document Model. Secondary model provides a mapping for the primary model to the concrete realization technology and describes models for Instances and Inline instances, Dimensions, Formula and Versioning models (packages), which provide support for respective XBRL 2.1 additional specifications [243]. Early works also considered metamodels of separate taxonomies such as COREP [183]; this approach is outdated.

An important criteria of XBRL evaluation as technology alternative is the amount and quality of mature tools or existing solutions which are necessary for implementation, development of XBRL taxonomies, filling instances as well as various API and SDK for development of custom tools. The XBRL component layers view is used to illustrate the place of these implementations in XBRL component context (Figure 8).It shows that user community or third party counterparts play an important role as consumers of XBRL providing both vocabulary, content as well as infrastructure; therefore they have to be provided with tools necessary to develop and extend XBRL based solutions. Again, [89] showed that some vendors (Fujitsu, Reporting Standard, UBMatrix) offer full support most or all XBRL-realetd aspects and features such as taxonomy development, processing and validation, analysis, storage and API, while others (Hitachi, CoreFilling) mostly provide developer-oriented solutions such as XBRL engines. Currently there are a few open-source XBRL solutions; an evaluation of them was made according to ([35, 168]; cited by [89]). Most of open source XBRL tools are just Java API for developers for XBRL processing without GUI-based tools. One of the most reliable and promising open source solutions seem to be xBreeze Open Source Edition provided by UBMatrix as well as Batavia XBRL Java Library (BXJL) and Inteco solution; it was stated that more of 50 ERP solution vendors used Inteco XBRL API to implement their XBRL functionality ([168]; cited by [89]). Another framework, which became quite mature at the time of writing, is Arielle framework, written in Python, which implements MVC design pattern based architecture. A useful criteria is that it’s data retrieval and processing functionality is based on SEC EDGAR model. Another important criteria is its support for additional XBRL specifications, such as XBRL Formula using XPath, Versioning modules as well as thorough XBRL 2.1 specification support. The authors of Arielle even describe its possibilities for XBRL based data mining application, although no specific algorithms ar machine learning techniques are

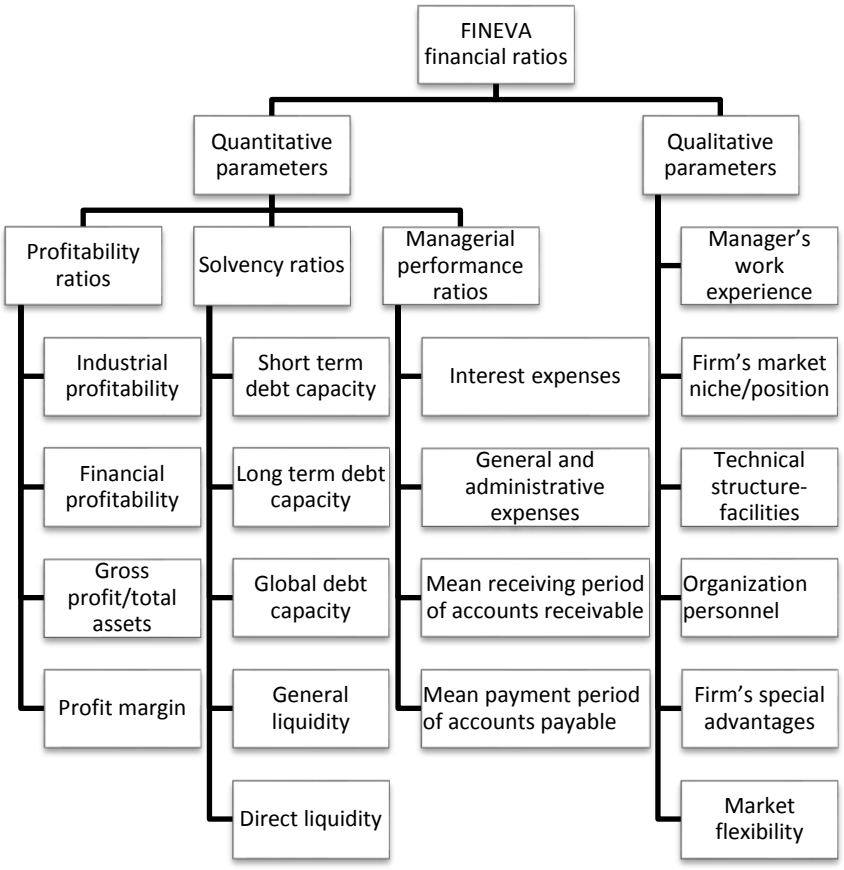
described in their paper [82].

To conclude this section, it can be seen that XBRL standard together with modern Semantic Web technologies offers many possibilities, e.g., automation of information retrieval and model updating in real time thus making evaluation even more precise. Another important extension can be direxts integration of rules defined in XBRL Calculation and Formula Linkbases which might be leveraged to ensure the integrity and validity of data as well as define additional secondary financial ratios.

2.6.4. Examples and developments of DSS in financial domain

A large amount of DSS structures and frameworks has been considered in different research: DSS for investment decision support [201],

Early expert systems for financial modelling were developed using rule-based approach (*if..then* rules); they use both quantitative and qualitative parameters.



Source: adopted from L. Nedović, V. Devedžić. Expert systems in finance – a cross-section of the field.

Figure 9. FINEVA financial ratios

To illustrate this approach, FINEVA (FINancial EVAluation) system (Matsatsinis, Doumpos & Zopounidis 1997; cited by [170]) is presented as an

example of such system. This expert system is developed for financial performance evaluation based on multiple criteria evaluation. It has been developed using M4 development environment for expert systems in Technical University on Crete and knowledge acquired from both literature and financial experts. The knowledge has been elicited using decision tables, represented using decision trees and production rules. Both qualitative and quantitative evaluations can be used; equal weights are assigned according to the proposal of the experts. The output is firm's ranking according to the risk level computed using acquired knowledge [170].

The taxonomy of FINEVA financial ratios is given in Figure 9; the ratios used in this system are grouped to profitability ratios, solvency ratios and management ratios. According to Nedović and Devedžić, FINEVA knowledge base at the time of writing of [170] consisted of 1693 rules which described more than 13000 possible combinations. Some of the modeling rules used in FINEVA are given in Table 10.

Table 10. Modeling rules for quantitative financial ratios in FINEVA

Industrial profitability A1: A1 < 10% not satisfactory 10% < A1 <= 20% medium 20% < A1 <= 30% satisfactory A1 > 30% very satisfactory	Financial profitability A2: A2 <= 17.5% not satisfactory 17.5% < A2 <= 20% medium 20% < A2 <= 23% satisfactory 23% < A2 very satisfactory	Gross profit/Total assets A3: A3 <= 0% not satisfactory 0% < A3 <= 50% medium 50% < A3 <= 75% satisfactory A3 > 75% very satisfactory
Profit margin A4: A4 <= 0% not satisfactory 0% < A4 <= 50% medium 50% < A4 <= 100% satisfactory A4 > 100% very satisfactory	Short-term debt capacity B1; B1 < 25% not satisfactory 25% < B1<= 50% medium 50% < B1 <= 75% satisfactory 75% < B1 <= 100% very satisfactory	Global debt capacity B2: B2 > 80 % not satisfactory 60% < B2 <= 80% medium 40% < B2 <= 60% satisfactory B2 <= 40% very satisfactory
Long-term debt capacity B3: B3 <= 0.5 satisfactory B3 > 0.5 not satisfactory	General liquidity B4: B4 >= 2 satisfactory B4 < 2 not satisfactory	Direct liquidity B5: B5 <= 1 not satisfactory 1 < B5 < 1.5 satisfactory B5 >= 1.5 very satisfactory
Financial expenses C1: C1 > 5% not satisfactory 3% < C1 <= 5% medium 2% < C1 <= 3% satisfactory C1 <= 2% very satisfactory	General and administrative expenses C2: C2 > 8% not satisfactory 6% < C2 <= 8% medium 4% < C2 <= 6% satisfactory 2% <C2 <= 4% very satisfactory C2 <= 2% perfect	medium period of accounts payable C3, medium period of accounts receivable C4: C3 > C4 not satisfactory C3 <= C4 satisfactory
Circulation of inventories C5: C5 increasing not satisfactory C5 reducing or stable satisfactory		

Source: Ljubica Nedović, Vladan Devedžić. Expert systems in finance – a cross-section of the field.

Various types of DSS for credit risk domain are described in scientific literature. Some researchers implement multi-agent based solutions as a network of problem solvers that perform together to solve problems [13], other focus on decision support system architecture with application of modern machine learning techniques

such as SOM [159] or SVM [64]. Multiple criteria decision aid is also applied [16], as financial statements and various financial ratios are used as main source of information in most of similar research.

Table 11. Currently developed structures for financial and credit risk DSS

Author	Cheng et al. [48]	Huai [111]	Mahmoud et al. [151]	Tsaih et al [215]	Zhang et al [260]
Name	Financial knowledge management system	Enterprises Group Financial DSS	Expert System for Banking Credit Decision	Credit scoring system for small business loans	Framework for financial DSS
General purpose	Data management, ETL support for modeling	Financial decision support	To facilitate banking credit decision support	Credit scoring DSS with embedded models	
Supported processes					
Acquisition	☑	FIS, ERP, SCM, HR, other	☑ Document analysis		☑
Transformation	☑	☑			☑
Data management	☑	☑			☑
Model management		☑		☑	☑
Knowledge management		☑			☑
Metadata management	☑				
Inference engine		☑	☑	☑	
Loan processing				☑	
Storage					
Data base	☑	☑	☑	☑	☑
Model base		☑	☑	☑	☑
Knowledge base		☑	☑		☑
Explanation knowledge base			☑		
Decision support					
Analysis		☑		☑	☑
Forecast		☑			
Decision		☑		☑	☑
Planning		☑			
Other					
Techniques used	N/A	Sensitivity analysis, simple and multiple regression, non-linear regression	N/A	Probit regression	N/A
Rules		☑	☑		
Metadata		☑ Definitions	☑ Classified domain concepts		
Ontologies	☑		☑		
Web services			☑		☑

Other papers also propose similar architectures; yet often only high level structure and main components are described [48, 95, 111]. Zhang et al. [260] present their framework of DSS structured as multilayer system, consisting of information integrated platform layer, utilization layer and information representation layer. Mahmoud et al. [151] developed a banking credit expert system, using data from periodicals, references and books, banks reports and publications, research, working papers and banking studies which is updated by domain experts. The knowledge base for their tool consists of five main components, namely Economic Feasibility Study, Financial Feasibility Study, Marketing Feasibility Study, and Collaterals; therefore, it is compatible with Basel II regulation which encourages integration of several risks. Tsaih et al. [215] proposed N-tier architecture with internal credit scoring model transformation into XML document. It consists of thin client layer (representing GUI in Web browser), middle tiers include the web server, Management Application Server (MAS), which provides interfaces to manage TDV, Database and XML repository and perform other management tasks, Loan Processing Subsystem (LPS) with Case Processing Application Server (CPAS) and Evaluation Module (EM) with XML parser and model engine sub-modules, and Model Installing Subsystem (MIS) which consists Model Defining Application Server (MDAS) and Model Recording Module (MRM).

Table 11 compares several proposed frameworks by identifying their components and functionality. Note that commercial systems for financial and credit risk management are not described in this work; their comparison can be found in PhD thesis by Merkevicius [159]. Earlier research is based on model-driven DSS development [131, 138] or rule-based principles, resulting in knowledge-driven expert systems [134, 170]. Other proposed architecture for DSS to support specific tasks in credit risk, e.g., credit card assessment [158].. An interesting solution is proposed by Kotsiantis et. al. [135] who developed a distributed ontology-based credit evaluation system with an application of C4.5 algorithm for scoring and intelligent search and reasoning possibilities; although they referred to XBRL as one of the options which would enable analytical possibilities offered by Semantic Web technologies they chose their own developed ontology to represent financial statements. Both of proposed system prototype solutions were engineered using JAVA technologies which prove to be a good choice for implementation of such

system

However, data access and automated collection and processing capabilities were not described in these works (although it can be viewed as default Extract-Transform-Load approach). This is an important aspect in data-driven and model-driven DSS as integration with various sources enrich these applications with new capabilities as well as with an increased quality of decisions, especially if the most recent data is continously provided by these sources. Although this subject is more explored in investment systems (especially high-frequency trading) such capability might help to indicate trends or caveats of particular sectors as soon as possible especially if financial reporting is made on different moments in each time period. Standardized data also reduces human error possibilities and might provide additional indicators which were not even considered at the time of modeling. As stated in Section 2.6.2, modern financial standards offer many benefits for each counterparty, including data aggregators; therefore, such integration is necessary for modern decision support. Data accessibility problems also have to be considered as access to financial data in risk sector is often limited and third counterparties such as financial data aggregation services have to be used. However, financial institutions such as SEC provide open access to XBRL data [241], therefore interdisciplinary research to utilize it for DSS functionality becomes an important and useful task.

2.7. Conclusions

1. Intelligent (based on artificial intelligence and machine-learning techniques) methods propose a variety of opportunities to develop modern self-adaptive and natural computing based models which incorporate both mathematical/statistical and decision making foundations found in nature or human cognition. Their advantages over similar standard statistical modelling techniques have been shown and discussed in numerous papers.

2. Support Vector Machines are often applied as efficient solution for classification problems. Analysis of currently developed SVM algorithms and their modifications showed that hybrid models combining SVM and other machine learning techniques often outperformed similar models based on other classification techniques such as ANN or decision trees. Similarly to other technqies such as ANN it can be parallelized which is useful in high-performance computing. However, lack of well-developed common framework combining all or most of these techniques

might slow down its adoption as solution for various problems, as well as comparative research/benchmarking or development of hybrid models which might try to combine several techniques.

3. SVM has several advantages over other techniques such as avoidance of overtraining, overfitting, architecture selection and testing problems, which have to be tackled in ANN and similar techniques. According to comparative analysis, accuracy and flexibility are its main strenghts (according to analysis of Yu et al., it is one of two most flexible techniques with linear programming), therefore, as most of other machine learning techniques, it is relatively complex and is hard to interpret for non-ML experts (yet not as hard as other „black-box“ models, such as ANN or evolutionary computing).

4. Main concepts and techniques for credit risk evaluation are also analyzed, as well as various kinds of other risks, sample ratios and evaluation techniques used for their evaluation, as well as their possible influence in credit risk evaluation. Similar techniques, often used to compare results, are discriminant and logistic techniques widely used in real world applications; main techniques are also decribed in detail. Basel standard is one of the most widely accepted financial frameworks for regulation of financial institutions which also directs development and management of solvency techniques. Analysis of this standard proved that integration of other risks into general framework is considered as necessary; therefore, this work concentrates mostly on credit risk.

5. Integration of financial standards is relevant as they offer real-time modeling and synchronization capabilities, using standardized data. The research described in further sections is based on data provided by SEC. XML technology enables standardized access to data structure and relations between its elements as well as their description; standards such as XBRL enable definition of validation or derived rules which can be used as rules to derive new data for modeling application. This standard is extensible, flexible and can be adapted for various needs. XBRL standard is described in more detail than other standards, with its advantages and disadvantages as well as implementation possibilities. This is a premise to develop a mapping model for XBRL interfacing.

6. Classification and basic structures of decision support systems and expert systems are are presented in this work. None of these structures considers external

data source integration and validation; therefore a structure defining all of these concepts should be developed. Analysis of existing DSS structures for credit risk evaluation and related tasks described in literature showed that these systems considered financial standard integration as an option, without detalization of particular aspects, or did not describe such option at all. This criteria is considered as important, thus integration of financial standards will also be discussed in DSS design.

3. DEVELOPED TECHNIQUES AND TOOLS FOR EVALUATION

This section describes techniques, methods and tools for their research, insights and evaluation, such as generic framework for intelligent hybrid model, as well as the developed methods. A framework for multidimensional analysis for credit risk evaluation based on intelligent techniques, which enables analysis in different views and levels using intelligent hybrid models, is also given in Appendix C. Main metrics and ratios used in research and evaluation of results are also given in this section.

3.1. Implemented classification techniques

This section describes techniques proposed in this work – feature selection and SVM based technique (further referred as FS-SVM) with discriminatory evaluator (referred as FS-SVM^{DA}), its extension for sliding window testing approach as well as evolutionary techniques (GA and PSO) based approach for optimization (parameter selection) of linear SVM. Note that multiexpert evaluation driven classifier based on hierarchical majority voting and discriminant analysis (GDM-FS-SVM^{DA}, given in Appendix B) might also be considered as an option for implementation in these approaches [61]; however, as it was developed and tested only using tree-based and rule-based classifiers, it is not further discussed.

3.1.1. FS-SVM^{DA} technique

This technique, first applied in [56] and later used in [30,58] is based on feature selection and SVM based classification. Thus every instance is evaluated using discriminant analysis and the outputs are converted to bankruptcy classes, thus enabling problem formulation as classification problem.

FS-SVM^{DA}

Input: Dataset D with a given set of companies CM, D^O – outputs of this dataset, obtained with discriminatory evaluator DA

C_V – set representing possible class values

A – the set of attributes

n_{att} – number of attributes (financial ratios)

l – the number of companies in the dataset

n_C – number of entries for particular company C

k – index of the company in the dataset

D_{Ck} - the subset of dataset D with size n for k-th company, $D = D_{C1} \cup D_{C2} \cup .. \cup D_{Cl,}$

1. Perform evaluation:

for $\forall D_i \in D$:

ev = evaluate(D_i, *DA)*, C_V); (Calculate evaluations and convert to bankruptcy classes)

if *ev* = {} (if instance cannot be evaluated)

D = remove(D_i, D) (Remove i–th entry from the dataset)

else

D_i^O = map(*ev*, C_V);

2. Perform data imputation:

for $C \in CM$

for $i = 1,..., n_C,\ j = 1,...,n_{att}$

(if the value is empty, average value for particular company is assigned)

if $D_{Ck}(x_{i,j}) = \{\}$

$$D_{Ck}(x_{i,j}) = \frac{\sum_{i=1}^{n} D_{Ck}(x_{i,j})}{n}$$

3. Perform data transformation by computing differences (optional):

for $C \in CM$

if $n_C > 1$

(compute the differences)

$$\forall i > 0, \forall j > 0, \forall k > 0 : D_{Ck}(x_{i,j}) = D_{Ck}(x_{i+1,j}) - D_{Ck}(x_{i,j})$$

$$D_{C_k}^O = \begin{cases} 1, when\ D(X_i) > D(X_{i-1}) \\ 0\ when\ D(X_i) = D(X_{i-1}) \\ -1, when\ D(X_i) < D(X_{i-1}) \end{cases}$$

(transform bankruptcy classes using this rule)

else

D = remove(D_C, D) (the single entry for the company is removed)

4. Divide companies to disjoint sets whose data will be used for training and testing

$C = C\_train \cup C\_test$, and $|C\_train| > |C\_test|$

5. Calculate training and testing data split percentage

6. Create disjoint sets as training and testing data by splitting data of selected companies in the sector by a percentage calculated in Step 5 ($C_D = C_D\_train \cup C_D\_test$ and $| C_D\_train| > | C_D\_test|$);

7. Apply feature selection procedure:

A' = select(*A*) (select attributes used in modeling)

8. Perform training, testing and evaluation procedures.

Output: a) a model (a list of support vectors and model parameters) that might be used to forecast, b) the list of selected attributes *A'* which forms this new model

Source: adopted from [56].

Algorithm 5. FS-SVM^{DA} algorithm

Instances with empty outputs (records, which couldn't be evaluated because of lack of data or division by zero) are eliminated and data imputation is performed by filling missing values with average value of corresponding ratio for particular company. The iteration variables are defined as follows: *i* is the index of D_{Ck} instance, *j* is the index of financial atribute in instance *i* of D_{Ck}.

Data transformation step is optional; therefore it might be applied to forecast changes. In its equation $D(X_i)$ – value of current instance evaluation by a particular model, $D(X_{i-}1)$ – value of previous instance evaluation by a the same model (the entries are sorted by balance date) . This expression represents the change in risk evaluation value. The training and testing data split percentage is important if the number of financial entries is different for each company or if some instances were rejected as not evaluated by evaluator. The feature selection procedure not only helps to reduce dimensionality, but it also obtains statistically significant attributes which are then used to develop a new classifier based on other evaluator.

3.1.2. Genetic algorithm and PSO approach for linear SVM optimization

This techniques implemented in this section were proposed in [60,62]. They combine linear SVM and parameter selection using evolutionary GA and PSO techniques and are further referred as GA-LinSVM and PSO-LinSVM respectively.

PSO-LinSVM. As mentioned in Section 2.1.10.4, linear SVM based classifiers, although having different, operate using several common parameters. This gives an option for automatic selection of classifier using metaheuristic techniques such as evolutionary optimization and swarm intelligence. Thus a classification technique based on Particle Swarm Optimization and linear SVM combination, namely PSO-LinSVM is developed.

Each particle $P = <p_1;p_2;p_3>$ is represented as follows:

p_1 –non-negative integer value, that represents the algorithm used for classification

p_2 – real value, cost parameter C

p_3 –real value, which represents bias term

The main objective of this algorithm is to maximize fitness function defined as sum of TPR values for each class:

$$f_{fitness} = \sum_{i=1}^{N_C} TPR_i = \sum_{i=1}^{N_C} \frac{TP_i}{FN_i + TP_i} \tag{3.1}$$

where N_C is the number of classes, TPR_i – TPR value for i-th class. Alternatively, it can be defined as minimization problem where it is aimed to minimize the difference between "ideal" performance (i.e., when TPR value for all classes is equal to 1) and performance obtained by the classifier

$$f_{fitness} = N_C - \sum_{i=1}^{N_C} TPR_i = N_C - \sum_{i=1}^{N_C} \frac{TP_i}{FN_i + TP_i} \tag{3.2}$$

Many authors [234](Wang, Chin et al.) choose accuracy for fitness evaluation, thus seeking to obtain a classifier with best accuracy performance; however, in case of imbalanced learning, accuracy is not the best option (it is possible to obtain high classification accuracy, if the classifier correctly recognizes most of "majority" instances, but fails to identify most of "minority" instances), so sum of TP rate values is selected for this case. These evaluations are obtained by performing a k-fold cross-validation training (as the number of instances increases while training, it's inefficient to choose a large value of k; k=2 or k=3 might be a good choice). As the formula shows, the optimal solution can be obtained only in case of perfect classification; as this happens very rarely, the main goal is to find satisfactory solution. Thus algorithm stops after no improvement in its performance is observed.

Note that p_1 value itself does not play an important role in obtaining position value, as p_1 is initialized randomly in whole search space, and optimization is done according to performance of SVM classifier represented by this particle. However, scattered values may influence particle velocity; therefore, it is required that p_1 values are non-negative successive integers (i.e., given $cl_{min} \leq P_{i1} \leq cl_{max}$, $S(i) = i+h$ for each P_{i1}). Although it is possible that it can be used with other h values, computationally it is not a reasonable approach, as corresponding population initialization and velocity equations would require modifications by replacing *round* operations with operators which ensure that P_{i1} and velocity values stay valid and require additional operations. Thus h = 1 was used in the experiments. The results can depend on implementation of random number generator used in the implementation of this algorithm, as well the number of particles used in optimization - the larger number of particles is used, the better coverage of search space is obtained, but the larger is the demand for computational resources. Such inner encoding is used in further research:

0 -- L2-regularized logistic regression (primal)

1 -- L2-regularized L2-loss support vector classification (dual)

2 -- L2-regularized L2-loss support vector classification (primal)

3 -- L2-regularized L1-loss support vector classification (dual)

4 -- L1-regularized L2-loss support vector classification

5 -- L1-regularized logistic regression

6 -- L2-regularized logistic regression (dual)

An optimized version for linear SVM classifier selection based on these principles is presented as Algorithm 6. Some PSO related issues such as velocity clamping (described in Section 2.1.10.4) are also implemented. The algorithm is presented as a solver for minimization problem.

PSO-LinSVM(n, c1, c2, rangeC, rangeBias, terminate_iterations, max_iterations)
$k \leftarrow 3$ (number of dimensions in particle, representing different SVM classifiers as described above)
perf $\leftarrow$ *[]*
$cl \leftarrow \{i \mid cl_{\min} \le i \le cl_{\max}, cl_{\min} \in Z, i \in Z, cl_{\max} \in Z\}$
global_fitness $\leftarrow$ *0*
term_iterations $\leftarrow$ *0*
$t \leftarrow 0$ *number of iterations*
$P \leftarrow Init(n)$ *Initialize a 3-dimensional swarm*
for $\forall p_x \in P$
  $p_{x1} \leftarrow cl_{min} + round(rand(0,1) * (cl_{max} - cl_{min}))$
  $p_{x2} \leftarrow cl_{min} + rand(0,1) * (C_{max} - C_{min})$
  $p_{x3} \leftarrow b_{min} + rand(0,1) * (b_{max} - b_{min})$
  $y_p \leftarrow p;$
repeat
  if *no_iterations = max_iterations* return SVM(y_p);
  for $\forall p_x \in P$ *set the personal best position*
    $f(x_p) \leftarrow evalSVM(p_{x1}, p_{x2}, p_{x3})$
    if $f(x_p) < \hat{y}(t)$ *set the global best position*
      $y_p \leftarrow x_p$;
      term_iterations $\leftarrow$ *1* *no need to terminate, continue searching*
    else
      term_iterations $\leftarrow$ *term_iterations + 1*
    if $f(y_p) < f(\hat{y})$ $\hat{y} = y_p$
  for $\forall p_x \in P$
    for j=1:k
      $Vmax \leftarrow \delta_j \times (R_{max,j} - R_{min,j})$ *Maximum allowed velocity*
      if (j = 1)
        $Vmax \leftarrow round(Vmax)$;
        $v_{pj}(t+1) = v_{pj}(t) + round(c_1 \times rand(0,1) \times (y_{pj}(t) - x_{pj}(t)) + c_2 \times rand(0,1) \times (\hat{y}_j(t) - x_{pj}(t)))$
        else
        $v_{pj}(t+1) \leftarrow v_{pj}(t) + c_1 \times rand(0,1) \times (y_{pj}(t) - x_{pj}(t)) + c_2 \times rand(0,1) \times (\hat{y}_j(t) - x_{pj}(t))$
      $v_{pj}(t+1) \leftarrow (v_{pj}(t+1) < Vmax\ ?\ v_{pj}(t+1) : Vmax)$
    $x_p(t+1) \leftarrow x_p(t) + v_p(t+1)$
    $y_p(t+1) \leftarrow \begin{cases} y_p(t), if\ f(x_p(t+1)) \le f(y_p(y)) \\ y_p(t+1), if\ f(x_p(t+1)) > f(y_p(y)) \end{cases}$
      if $x_{p1}(t+1) > cl_{mas}$
        $x_{p1}(t+1) \leftarrow cl_{min}$;
      if $x_{p2}(t+1) < C_{min}$)
        $x_{p2}(t+1) \leftarrow C_{min}$;

$\hat{y}(t) \leftarrow \min(f(y_0(t)),..., f(y_n(t)))$
$t \leftarrow t+1$
until (*term_iterations* < *terminate_iterations*)
Output: Optimal linear SVM classifier SVM(y_p)

Algorithm 6. PSO-LinSVM algorithm

Such parameters are defined for the proposed algorithm:

- n – size of swarm;
- c_1 – PSO coefficient for cognitive component;
- c_2 – PSO coefficient for social component;
- *cl* - a set of classifiers, represented by inner encodings;
- *rangeC* = $[C_{min}; C_{max}]$– range of cost parameters which is considered (note that $C >= 0$);
- *rangeBias* = $[b_{min}; b_{max}]$ – range of B (bias term) parameters which is considered in optimization;
- *terminate_iteration* – optional parameter which defines the number of iterations after which PSO optimization should be terminated if no further improvement is observed;
- *max_iterations* - maximum number of iterations for PSO optimization. It is also optional and if it not considered the procedure loops until terminate_iteration criteria is satisfied. This can be considered if a fast convergence to optimal solution is known to occur.

Mainly two most important parameters are cognition coefficient c_1 and social coefficient c_2. As Engelbrecht points out, they stochastically model social (confidence on solutions by its neighbours) and cognitive (confidence of its own solutions) aspects of particle velocity. Unfortunately, these parameters can be selected empirically. Therefore, larger social coefficient is a better solution for search spaces with smooth surface while larger cognitive coefficient is preferred for problem spaces which have many global and local optimas and therefore result in rough search space [78]. This technique also comprises such aspects as velocity clamping, where $V_{max,j}$ represents maximum allowed velocity in dimension j. According to Engelbrecht, large values of $V_{max,j}$ facilitate global exploration, while smaller values encourage local exploitation. It is often computed as a fraction δ of search space and selected empirically, according to the problem which is solved. Therefore, in proposed technique it is calculated as

$$\delta_j = \frac{R_{\max,j}}{|R_{\min,j}| + |R_{\max,j}|} * 0.8 \tag{3.3}$$

where $R_{min,j}$ is denoted as minimum of search space for j dimension, $R_{max,j}$ - as its maximum. To deal with constraints, particle "teleportation" principle (i.e., if the value of the particle is "out of bounds" for particular dimension, it is set to initial value) is employed.

GA-LinSVM. This algorithm uses real-valued GA, although its chromosome contains integer values as well. The chromosome consists of 3 genes (further referred as G_1, G_2 and G_3), they're defined in the same way is in PSO-LinSVM case. Fitness function is defined in the same manner.

Although experimental results seem promising, yet there are several important factors which might improve the performance. The performance of SVM classifiers much depends on the selected parameters; yet, linear SVM has a smaller number of them which makes it simpler. However, the selection procedure might still be improved by selecting different GA recombination procedures or their combinations (mutation and crossover). PSO neighborhood and topology has not been explored in this research, thus it also leaves room for improvement. Another aspect that should be taken in mind while applying the selected procedure is, as already mentioned before, imbalanced learning procedure, especially as classes are computed dynamically by external evaluator. SVM is one of the machine learning techniques which is sensitive to dataset imbalance as "majority" classes tend to outweigh "minority" classes by pushing classification boundaries over them. Many techniques are applied to overcome this barrier such as internally implemented class-weighting, cost-sensitive learning and evaluation, internal classifier enhancements, numerical sampling techniques, such as bootstrap, undersampling, oversampling. Dataset balancing is crucially important in identification of bankrupt companies if they are represented by minority entries, as identification of bankrupt company might cost more to the creditor than the misidentification of it.

3.1.3. FS-SVM and sliding window testing based approach

This method extends the technique described in Section 3.1.1 with sliding window approach for testing; futher it will be referred as FS-SVM^{SWTest}. This approach was used in [59,60,62].

Thus full classifier development and testing methodology used in the

experiment is defined as follows:

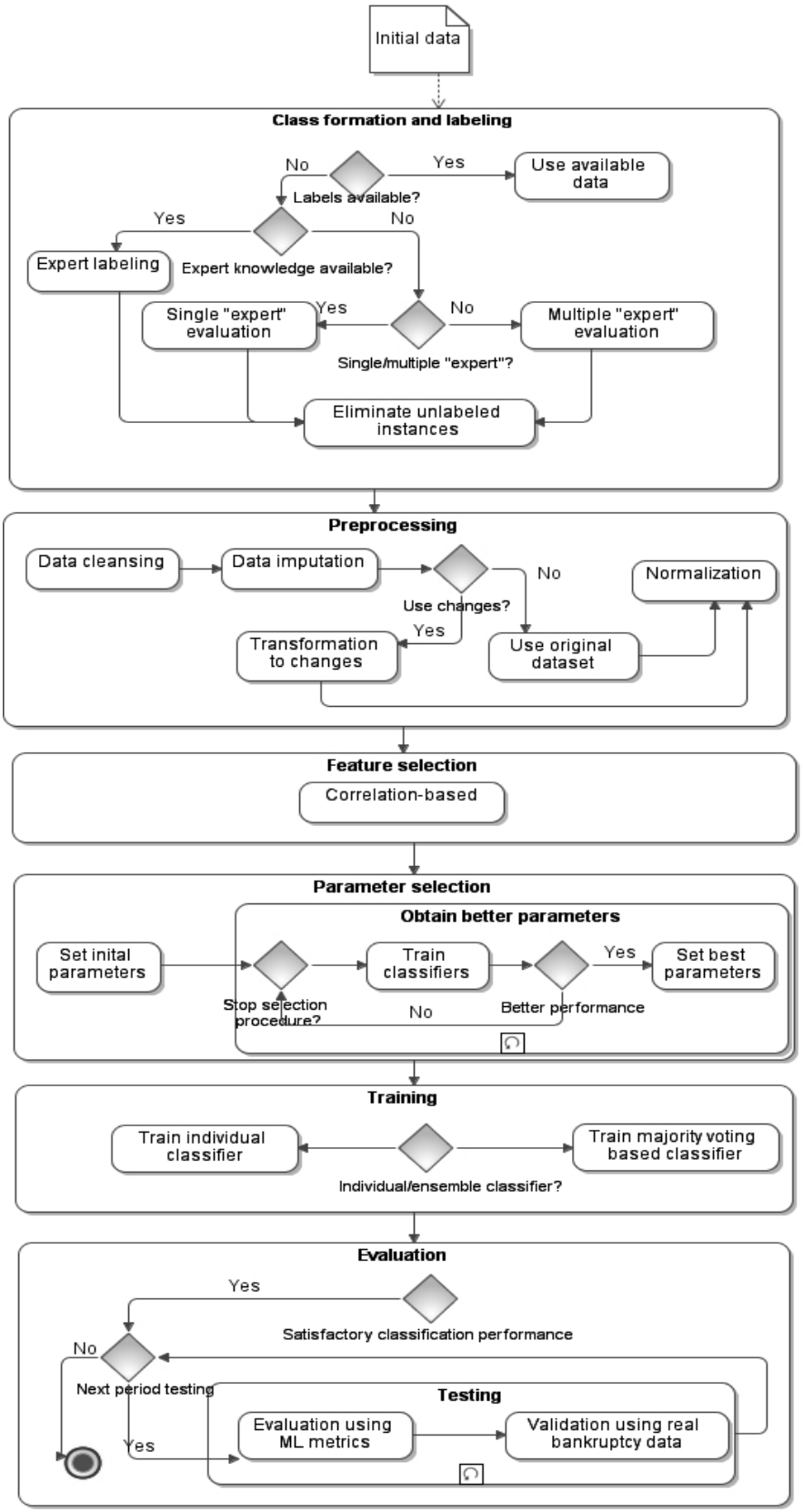


Figure 10. Generalized classification algorithm based on FS-SVM

1. Evaluate each financial entry by using discriminant analysis (or any other expert evaluation method, if possible) and compute bankruptcy classes.

2. Eliminate instances which could not be evaluated in Step 1 because of lack of data or division by zero and thus resulted in empty outputs.

3. Remove attributes from the dataset which have fewer values than specified threshold (70% was considered in this case).

4. Data imputation is performed by filling missing values with average value of particular attribute or by average value of company performance, as described in step 2 of Algorithm 5.

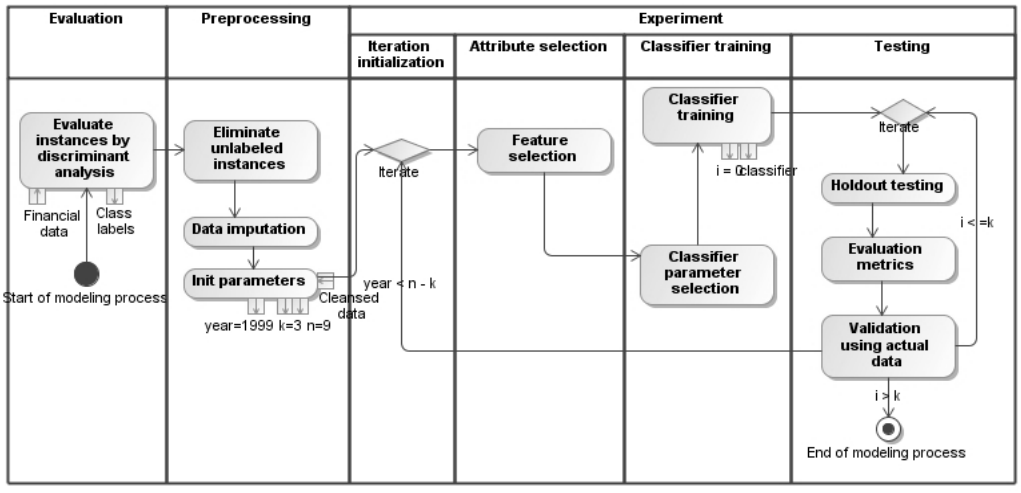
5. Perform the following steps for each $m \in [1, n-k]$, where n is the total number of periods, k is the number of periods which are used for forecasting:

   a. Apply feature selection procedure in order to select the most relevant attributes and reduce number of dataset dimensions;

   b. Perform classifier parameter selection manually or using heuristic procedures. Several techniques, such as described in Section 2.1.10 or in various sources [102, 75], can be employed for this task. An algorithm developed for parameter selection of linear SVM using PSO technique is decribed in the next section.

   c. Train classifier using data from first m periods.

   d. Apply hold-out testing using data from period p, $p \in [m+1, m+k]$; $p \in \mathrm{N}$.



Source: Danenas P., Garsva G. Credit Risk Modeling of USA Manufacturing Companies Using Linear SVM and Sliding Window Testing Approach.

Figure 11. The workflow of FS-SVM^{SWTest} based on discriminant analysis

Feature selection step is important for 2 reasons, like in previous algorithms:

1. To reduce data dimensionality of attribute space, thus forming a new subset of attributes and reducing the complexity of the model (the aspects of quality and complexity);

2. To obtain a set of statistically significant attributes to develop a new classifier based on other evaluator (the aspect of importance).

The output of each iteration in experimental stage is the trained classifier (a set of support vectors in case of SVM) and the set of selected attributes.

Figure 11 presents the workflow which is used for experimenting, while Figure 10 gives extended and more generic structure of this approach which also includes other possible choices such as evaluation using majority voting used in [61] which is not covered in this work.

Such approach can be applied in solving following problems:

1. To develop a new model based on class information of previous model and new set of features i.e., to develop a model based on information of existing model. Formally this can defined as transformation of given model $< X, f : X \to Y >, X \in R^N, Y \in N$ to a new model $< X', f : X' \to Y >, X' \in R^N, Y \in N$ using feature selection $f_s : X_N \xrightarrow{FS} X'$ where X_N is new financial data, X'- new set of features obtained by feature selection procedure f_S.

2. Integrate expert knowledge and express them using a particular model combining this experience and data, especially when his evaluations or knowledge cannot be easily expressed in a form of model or this dependency cannot be described of commercial purposes, i.e., when moving the model to a new environment with the absence of this expert. Having a set of expert evaluations Y, $Y \in N$ a model $< X, f : X \to Y >, X \in R^N, Y \in N$ is developed.

3. The expert cannot evaluate all data instances, either using any mathematical techniques or not. Such problems may arise of missing data, division by zero problems, etc.

4. Mapping external ratings to existing data and identifying most significant financial ratios as well as developing new, more efficient scoring techiques.

3.2. Evaluation metrices

A methodology popular in such researches has been chosen. It uses such measurements:

$$Type\ I\ error = \frac{number\ of\ observed\ "good"\ but\ classified\ as\ "bad"}{number\ of\ observed\ "good"} \quad (3.4)$$

$$Type\ II\ error = \frac{number\ of\ observed\ "bad"\ but\ classified\ as\ "good"}{number\ of\ observed\ "bad"} \quad (3.5)$$

$$Total\ accuracy = \frac{number\ of\ instances\ correctly\ classified}{total\ number\ of\ instances} \quad (3.6)$$

here terms „bad“ and „good“ define companies that have high or low value of risk evaluation. These results are usually expressed in percentage to show precision of correct classifications. Some authors used Type I accuracy and Type II accuracy measures which show accuracy respectively; in this case we calculated corresponding classification errors using

$$Type\ I\ error = 1 - Type\ I\ accuracy \quad (3.7)$$

$$Type\ II\ error = 1 - Type\ II\ accuracy \quad (3.8)$$

Type I accuracy shows the accuracy with which the model identified failed debtors as weak (“bad“); respectively, Type I error will show the accuracy with which the model didn't identify “bad“ debtors. Type II accuracy shows the accuracy with which the model identified “good“ debtors correctly; and Type II error will show the accuracy with which the model identified “good“ debtors as bad. Type I error is considered as more important than type II, as the inability to identify a failing company will cost a lender far more than rejecting a healthy company [33].

To evaluate overall performance, weighted mean was used as following:

$$\overline{Err} = \frac{\sum_{i=1}^{n} r_i Err_i}{\sum_{i=1}^{n} r_i} \quad (3.9)$$

where n is the number of sectors, Err_i is the value of error for sector i, r_i – the number of records in sector i used for testing. Here the weighted mean was evaluated only according to amounts of testing instances; however, if the proportion of training and testing instances varies significantly, it might be useful to evaluate weighted mean with both training and testing instances.

Note that Type I and Type II errors are not preferrable measurements in unbalanced learning evaluation (i.e., if largest part of data consists of instances labeled as one class, and relatively small part of them are labeled as another). Therefore it is often better to use standard technique for classifier performance evaluation, known as confusion matrix, and measures derived from its evaluations.

Confusion matrix is defined[8] as matrix with such values (adopted from [137]):

- a is number of **correct** predictions that instance is **negative**;
- b is number of **incorrect** predictions that instance is **positive**;
- c is number of **incorrect** predictions that instance is **negative**;
- d is number **correct** of predictions that instance is **positive**.

		Prediction	
		Negative	**Positive**
Actual	**Negative**	**a**	**b**
	Positive	**c**	**d**

Such measures for accuracy and efficiency evaluation can be obtained from confusion matrix:

- *Accuracy* (AC) is the ratio of the total number of predictions that were correct and overall number of predictions:

$$AC = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+FP+FN+TN} \tag{3.10}$$

- *Recall* or *True Positive (TP) Rate* is the proportion of positive cases correctly identified:

$$TPR = \frac{d}{c+d} \tag{3.11}$$

- *False Positive (FP) Rate* is the proportion of negative cases incorrectly classified as positive:

$$FPR = \frac{b}{a+b} \tag{3.12}$$

- *True Negative (TN) Rate* is the proportion of negative cases classified correctly:

$$TNR = \frac{a}{a+b} \tag{3.13}$$

- *False Negative (FN) Rate* is the proportion of positives cases incorrectly classified as negative:

$$FNR = \frac{c}{c+d} \tag{3.14}$$

- Precision is the proportion of the predicted positive cases that were correct:

$$prec = \frac{d}{b+d} \tag{3.15}$$

[8] Confusion matrix can be defined for any number of classes; here it is defined only for binary classification problems

Lewis and Gale proposed F-Measure as harmonic mean of accuracy and recall values [144]:

$$F_\beta = \frac{(\beta^2 + 1) * prec * recall}{\beta^2 * prec + recall} \tag{3.16}$$

Often a simplified version of this measure is used in research (β = 1); it will be also further used in the research:

$$F_1 = \frac{2 * prec * recall}{prec + recall} = \frac{2}{\frac{1}{prec} + \frac{1}{recall}} \tag{3.17}$$

3.3. Summary

This chapter presents classification methods which are developed including feature selection and SVM based technique with discriminant analysis used as evaluator (FS-SVM^{DA}), its extension for sliding window testing approach FS-SVM^{SWTest} as well as evolutionary techniques (GA and PSO) based approach for optimization (parameter selection) of linear SVM (PSO-LinSVM and GA-LinSVM correspondingly). These techniques can be implemented as classifiers in proposed classification with sliding window testing approach. Evaluation metrics used for classification evaluation in further research are also described in detail.

4. EXPERIMENTAL RESEARCH

This section presents extensive experimental research of proposed techniques, gives main obtained results, together with comparison of similar machine learning techniques.

4.1. Experimental research of FS-SVM^{DA} based models

4.1.1. Experimental analysis of SVM and neural network classifiers

The experiment was made by using data from EDGAR database of over 8600 companies from year 1999-2006 (Table 12). It consists of yearly financial records with 79 financial ratios and rates used in financial analysis. One of main goals for this research was evaluation of SVM suitability for large scale learning and classification.

Table 12. The sectors used in experiments

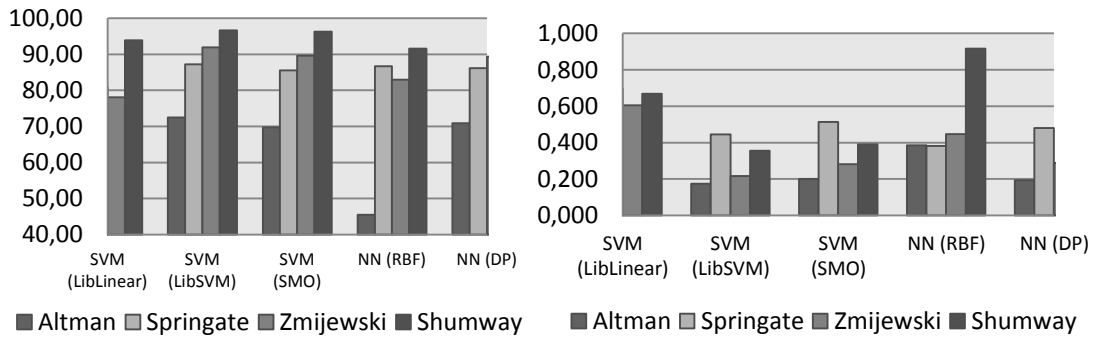
Sector code	**Sector name**	**Total no. of companies**	**No. of companies for training**	**No. of companies for testing**
01-09	Agriculture, Forestry, And Fishing	33	20	13
10-14	Mining	469	281	188
15-17	Construction	83	50	33
20-39	Manufacturing	3027	1816	1211
40-49	Transportation, Communications, Electric, Gas, And Sanitary Services	786	472	314
50-51	Wholesale Trade	287	172	115
52-59	Retail Trade	405	243	162
60-67	Finance, Insurance, And Real Estate	1853	1112	741
70-89	Services	1712	1027	685
All data		**8665**	**5199**	**3466**

This data was split into sectors according to SIC classification by their SIC code. 6:4 split was used in the experiment (60% percent of companies were selected for training). The experiment was run using Weka software with C-SVC, SMO, LIBLINEAR (L2-loss linear SVM). RBF Neural Network and Multilayer Perceptron were used as benchmarking techniques. The companies which data was used were divided into sectors according to their SIC code. The experiment was performed using quarterly and yearly data in each case. It was run using two different approaches. The purpose of the first part of the experiment was to identify the possibilities of classification and bankruptcy prediction by using non-transformed data (data with absolute values). Next step was to identify the possibilities by using changes in data and risk values to predict change dynamics. It was run with the same

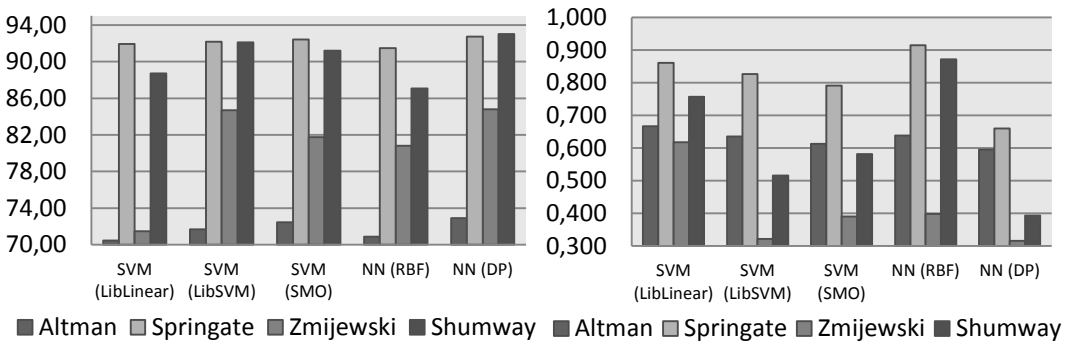
parameters as in the first part in order to compare the results appropriately. FS-SVM^{DA} was applied for every sector, as well the whole dataset for the overall evaluation. The number of all data is different from data in sectors as it also includes entries that weren't labeled as belonging to one of sectors in Table 12 or the number of entries in particular sector was too small for training and evaluating a classifier. Evaluators based on Altman (Eq. 2.41), Springate (2.44), Zmijewski (2.45) and Shumway (2.46) models were used to form bankruptcy classes.

Experimental kernel selection results showed that polynomial ketnel resulted in highest accuracy; however, it also resulted in largest amount of time needed for training classifier. SMO and LibSVM (C-SVC) classifiers, with parameters $C = 4$ for SMO; $C = 7$ and $\gamma = 7$ for C-SVC, were run using a polynomial kernel, although in some particular cases, when the number of training instances was large and the performance became extremely slow, RBF kernel was used for training. LIBLINEAR was run with $C = 20$. Multilayer perceptron was built using *(attributes + classes) / 2* hidden layers with iteration parameter of training epochs set to 500.

Analysis of the results from the first part of the experiment (Figure 12) shows that C-SVC in many cases outperformed other classifiers and it might be compared to the performance of multilayer perceptron. Considering the case of quarterly data, some graphs indicate that in particular cases SMO obtained results similar to LibSVM classifier (especially in case of using Shumway model for evaluation); this might be considered as a good option since SMO training time is significantly shorter than time for LibSVM classifier training. Type I error graphs also prove that LibSVM performance is among the best. Zmijewski evaluation based model might seem the best option in case of quarterly data as it has one of the highest accuracy evaluations and one of the smallest errors, especially in case of LibSVM.



a) Weighted accuracy and weighted Type I error (quarterly data)

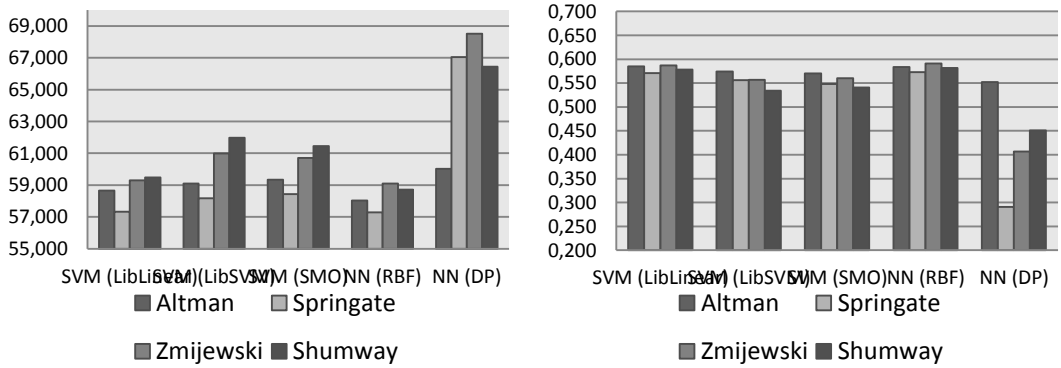


b) Weighted accuracy and weighted Type I error (yearly data)

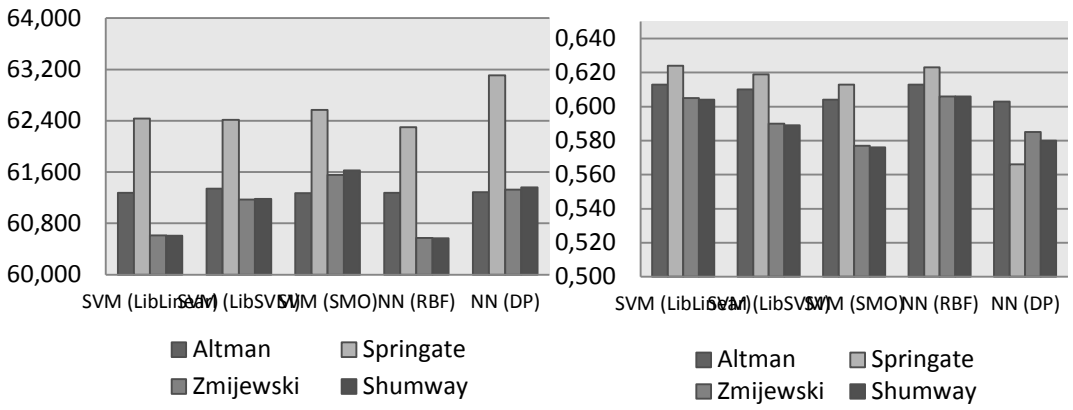
Figure 12. Overall performance of all models (case of primary data)

Considering the case of yearly data, the model based on Springate evaluation obtained best accuracy results, and the results were similar in all cases of classifiers. However, Type I Error is the significantly big (about 0.8 in most cases, and more than 0.6 in case of multilayer perceptron), which shows the inconsistency to use it for bankruptcy identification. Thus the Zmijewski or Shumway evaluation based models might seem the best option here, combining them with LibSVM or NN multilayer perceptron based classifier.

By analyzing the results of change data (Figure 13), it is obvious that multilayer perceptron is the best option here as it produced the highest overall accuracies and smallest errors; C-SVC outperformed other three classifiers, though the difference is small, compared to NN performance. However, the difference is not so significant in case of yearly data; SMO obtained best results among all SVM based classifiers with the highest overall accuracy and smallest error here.



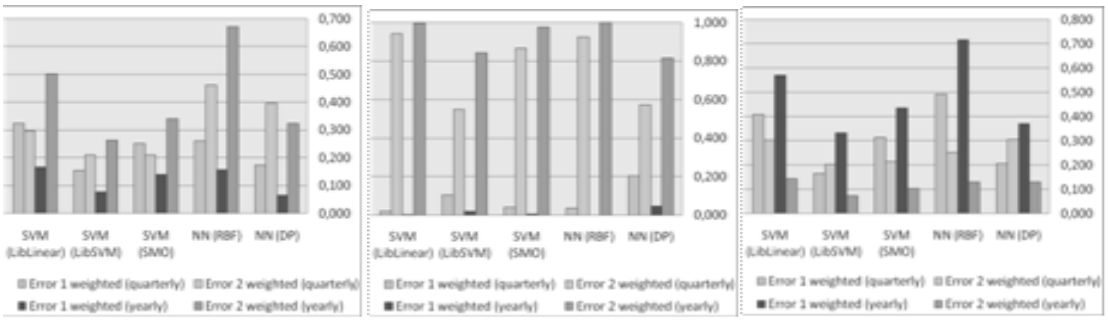
a) Weighted accuracy and weighted Type I error (quarterly data)



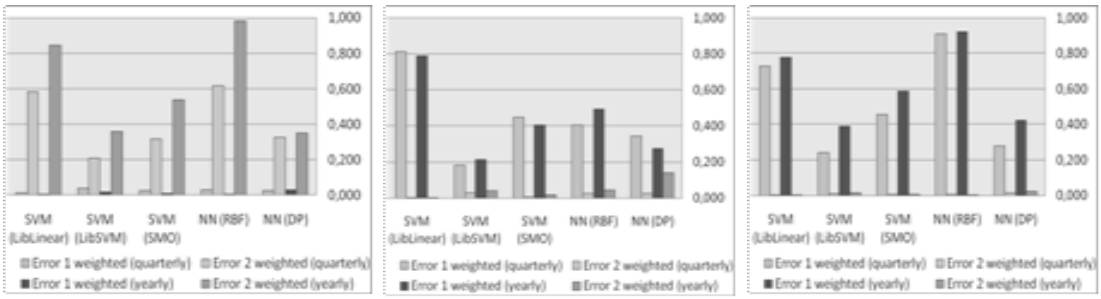
b) Weighted accuracy and weighted Type I error (yearly data)

Figure 13. Overall performance of all models (case of data transformed to differences)

The accuracy and the error in both cases of quarterly and yearly data in most cases respectively is in the ranges of 57-65% and 0.5-0.6 which is not confident enough in bankruptcy prediction, as well as the accuracy. Thus in case of bankruptcy identification using data changes it might be considered using only in some particular sectors where higher results were obtained.



a) Weighted average error values (bankrupt, average, healthy, Altman model)



b) Weighted average error values (bankrupt, Springate, Zmijewski, Shumway)

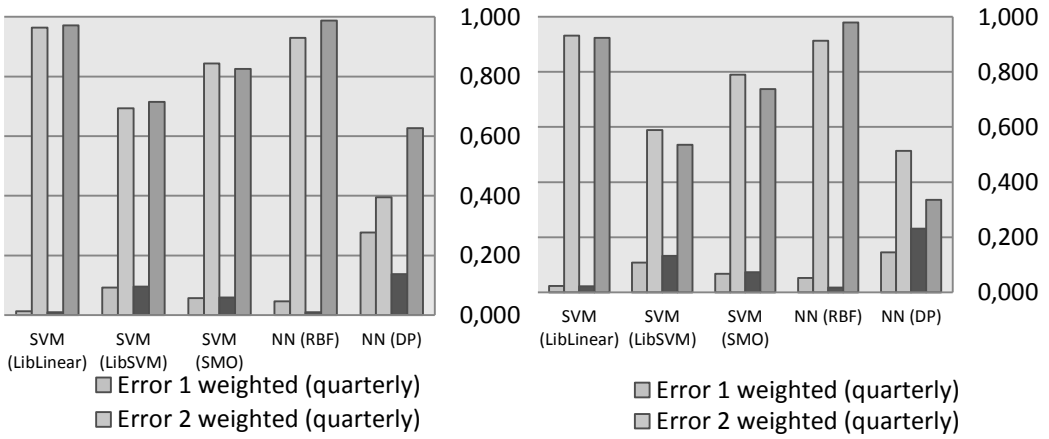
Figure 14. Error values for separate classes in case of data with primary values

No risk evaluation model can be marked as exclusively better here as the results are similar in both cases of yearly and quarterly data. The graphs indicate that slightly better results in quarterly data case were obtained using Shumway evaluation

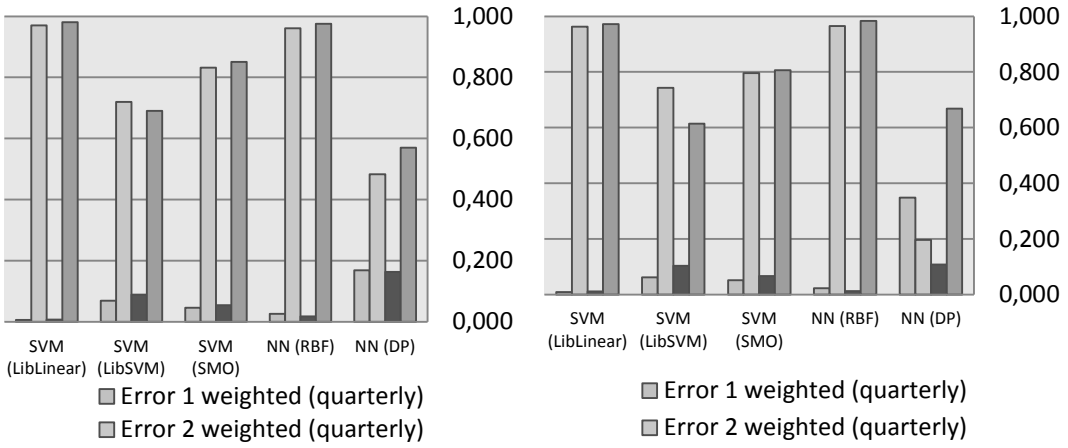
based classifier, and Springate-based model was better for evaluation in case of yearly data.

For more accurate evaluation Type I and Type II errors for separate classes (bankrupt, normal, healthy) were evaluated. Figure 14 represents these results for data with primary values (bankrupt, average, healthy in case of Altman model and bankrupt, healthy – in case of other models). Only cases of bankrupt classes are presented for Springate, Zmijewski and Shumway evaluation based models as they used binary classification, thereby error results for "bankrupt" and 'healthy" companies are inverse.

The graphs show in both cases (quarterly data and yearly data used to train and evaluate models) similar results were obtained. Springate based evaluation model had the smallest Type I error; it was less than 0.05 in cases of all classifiers in cases of both quarterly and yearly data; although these results look promising, they identify that SVM performs poorly on highly imbalanced datasets . In cases of models based on Zmijewski and Shumway evaluators it was less than 0.3 in cases of LibSVM and multilayer perceptron based classifiers; other classifiers showed significantly worse results. Type II error was also among smallest in cases of Zmijewski and Shumway evaluations based models; these models might be more useful to distinguish "healthy" companies. In case of Altman based evaluation model, both Type I and Type II errors are high, compared to the Springate or Zmijewski, and they vary. However, C-SVC and multilayer perceptron showed errors below 0.2 in case of bankrupt companies and in case of "healthy" companies with quarterly data; the results of identifying "healthy" companies using yearly data were slightly worse.



a) Altman *b) Springate*



c) Zmijewski *d) Shumway*

Figure 15. Error values for separate classes, case of differences

As the results of classifiers differ significantly it is obvious that classifier selection performs an important role in model creation process. Figure 15 shows visualization of error values for separate classes in cases of changes (-1 – financial condition gets worse, 1 – financial condition gets better). The results were similar in all cases. LIBLINEAR and RBF network based classifiers showed best results in Type I error; however, Type II error was the biggest in all cases for all the models. Conversely, multilayer perceptron performed with the smallest average Type II error, but Type I error obtained by this classifier was larger than in cases of other classifiers. This leads to conclusion that using data represented as changes between financial ratios for training might be useful only in identifying financial declines and thus help distinguishing "bankrupt" companies, but it will not predict accurately financial growth. Unfortunately, dataset imbalancement leads to low values of errors for classes represented by significantly smaller amount of instances.

4.1.2. Empirical research of various SVM based classifiers[9]

The experiment was run using LibSVM, LIBLINEAR, SMO and genetic search implementations in Weka software. Genetic search used for attribute selection was run using the following parameters: crossover probability equal to 0.6, number of generations equal to 20, mutation probability of 0.033 and population size of 20. SMO and LibSVM (C-SVC) classifiers were run using polynomial kernel with parameters C = 4 for SMO; C = 7 and $\gamma = 7$ for C-SVC. LIBLINEAR was run with C

[9]Based on paper Danenas, P., Garsva, G. Credit risk evaluation using SVM-based classifier.

= 20. These parameters were chosen in an experimental way.

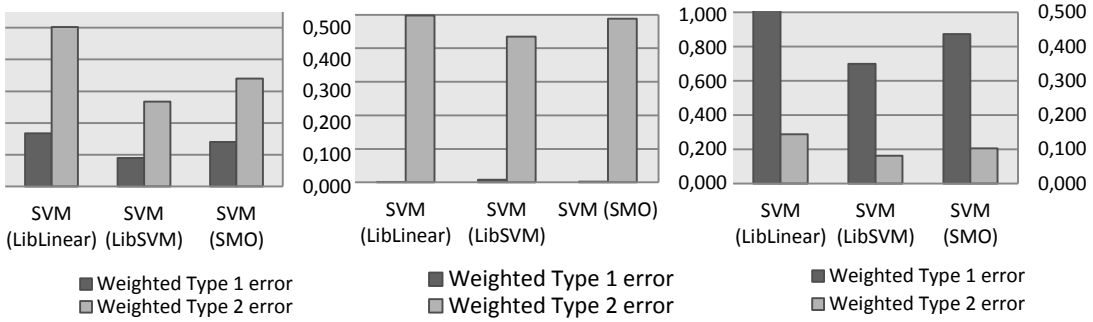
The results of these experiments using data from different sectors as well as all data are presented in Table 13.

Table 13. Experiment results

Sector	Number of companies for training	Number of companies for testing	Training percentage	No of training instances	No of testing instances	SVM (LIBLINEAR)		SVM (LibSVM)		SVM (SMO)	
						Accuracy	Type1 Error	Accuracy	Type1 Error	Accuracy	Type1 Error
Agriculture, Forestry and Fishing	20	13	59,56	134	91	85,71	0,3	75,82	0,561	86,81	0,265
Mining	281	188	59,78	1660	1117	71,17	0,612	77,17	0,408	72,78	0,559
Construction	50	33	60,67	324	210	84,76	0,534	72,38	0,596	88,57	0,349
Manufacturing	1816	1211	60,10	11641	7727	88,92	0,523	90,86	0,336	89,93	0,388
Transportation, Electric, Gas, Communications, Sanitary Services	472	314	59,63	2838	1921	70,85	0,666	81,05	0,294	76,37	0,453
Wholesale Trade	172	115	59,80	1095	736	91,58	0,555	90,08	0,363	92,66	0,473
Retail Trade	243	162	60,20	1570	1038	95,38	0,775	95,09	0,586	94,99	0,706
Finance, Insurance and Real Estate	1112	741	61,18	6413	4069	68,86	0,512	82,13	0,206	74,07	0,389
Services	1027	685	60,22	6241	4122	80,81	0,524	85,30	0,286	84,43	0,382
All data	5199	3466	63,53	36674	21054	70,45	0,667	71,67	0,635	72,46	0,613
Weighted mean						**81,21**	**0,552**	**86,39**	**0,318**	**83,95**	**0,420**

Source: Danenas, P., Garsva, G. Credit risk evaluation using SVM-based classifier.

The classifiers achieved satisfactory results using both small (~1500 and less) and large (from 6000 to 36000) number of instances; the best results were obtained by using LibSVM. Again, as in it was noted in Section 4.1.1, training LibSVM-based classifier using polynomial kernel is very demanding computationally when a large number of instances is used. Therefore, SMO or LIBLINEAR might be considered as an alternative as they obtained similar results.



Source: Danenas, P., Garsva, G. Credit risk evaluation using SVM-based classifier.

Figure 16. Weighted mean error values for all classes (bankrupt, average, healthy)

The table also shows that SMO classifier is a good option when the number of instances and attributes is high as it also performs this task in a significantly shorter

time than LibSVM based classifier used in this research.For overall evaluation weighted mean errors of all these three classes were calculated; they are given in Figure 16. These graphs show that LibSVM produced smallest errors in most of cases; however, there were sectors which resulted in extremely different errors (i.e., the sector of Finance) although the performance in other sectors was significantly better.

4.1.3. Comparison of SVM and Bayesian classifiers[10]

In order to compare performance of SVM and Bayesian method based classifiers an experiment was also performed with WEKA implementations of Naïve Bayes and Bayesian Network (further referred as BayesNet) classification methods. Naïve Bayes classifier was run using supervised discretization. BayesNet was used with the following parameters: estimator – SimpleEstimator algorithm for finding the conditional probability tables of the Bayesian Network with $\alpha = 0.5$. Hill climbing was used as search algorithm for adding, deleting and reversing arcs as the search is not restricted by an order on variables. These parameters were selected experimentally. Altman model based evaluator was selected to form bankruptcy classes.

Table 14. SVM and Bayesian classifier performance results

Sector	Training percentage, %	No of training instances	No of testing instances	SVM (LIBLINEAR)		SVM (LibSVM)		Naïve Bayes		BayesNet	
				Accuracy	Type1 Error	Accuracy	Type1 Error	Accuracy	Type1 Error	Accuracy	Type1 Error
01-09	59,556	134	91	85,71	0,3	75,82	0,561	83,52	0,105	84,62	0,079
10-14	59,777	1660	1117	71,17	0,612	77,17	0,408	77,35	0,174	77,17	0,174
15-17	60,674	324	210	84,76	0,534	72,38	0,596	77,14	0,091	75,24	0,109
20-39	60,104	11641	7727	88,92	0,523	90,86	0,336	68,10	0,097	68,12	0,098
40-49	59,634	2838	1921	70,85	0,666	81,05	0,294	69,91	0,123	69,91	0,124
50-51	59,803	1095	736	91,58	0,555	90,08	0,363	75,00	0,127	75,14	0,127
52-59	60,199	1570	1038	95,38	0,775	95,09	0,586	79,87	0,082	79,77	0,080
60-67	61,181	6413	4069	68,86	0,512	82,13	0,206	77,02	0,088	77,17	0,088
70-89	60,224	6241	4122	80,81	0,524	85,30	0,286	74,31	0,173	74,36	0,173
All data	63,529	36674	21054	70,45	0,667	71,67	0,635	72,26	0,174	72,29	0,174
Weighted mean				**81,21**	**0,552**	**86,39**	**0,318**	**72,68**	**0,117**	**72,70**	**0,117**

Source: Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers.

The results of these experiments (weighted accuracy values together with

[10]Based on paper Buzius, G., Danenas, P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers

weighted Type I errors) are presented in Table 14. LibSVM performed with highest accuracy as well as with highest weighted accuracy but it had higher weighted Type I error than Bayesian classifiers. Very similar results by both weighted accuracy and weighted error were obtained by using Naïve Bayes and BayesNet classifiers; although accuracy in most cases (especially in cases of higher number of instances for training) they obtained significantly lower results than SVM, but weighted errors and weighted mean errors were the smallest. Type I and Type II errors for separate classes - bankrupt" (B),"average" (A), "healthy" (H) were also evaluated.

Table 15 and Table 16 show these results. Type I Error was relatively small for "bankrupt" companies in cases of SVM based classifiers, except Finance, Insurance and Real Estate sector where it was very high (0.811).

Table 15. Type I error values for different classes

	LIBLINEAR			**LibSVM**			**Naïve Bayes**			**BayesNet**		
	B	**A**	**H**	**B**	**A**	**H**	**B**	**A**	**H**	**B**	**A**	**H**
01-09	0,026	0,036	0,381	0,064	0,024	0,714	0,105	0,036	0,182	0,079	0,048	0,182
10-14	0,012	0,000	0,889	0,050	0,006	0,580	0,174	0,081	0,048	0,174	0,081	0,054
15-17	0,010	0,051	0,625	0,021	0,164	0,688	0,091	0,155	0,090	0,109	0,155	0,100
20-39	0,015	0,000	0,618	0,033	0,001	0,394	0,097	0,235	0,136	0,098	0,234	0,136
40-49	0,008	0,002	0,948	0,032	0,051	0,401	0,123	0,215	0,092	0,124	0,214	0,091
50-51	0,009	0,003	0,628	0,031	0,024	0,407	0,127	0,160	0,050	0,127	0,158	0,050
52-59	0,003	0,000	0,818	0,007	0,010	0,618	0,082	0,136	0,052	0,080	0,137	0,055
60-67	0,811	0,000	0,012	0,303	0,030	0,052	0,088	0,148	0,092	0,088	0,147	0,090
70-89	0,014	0,000	0,702	0,059	0,005	0,369	0,173	0,118	0,095	0,173	0,118	0,094
All data	0,001	0,000	0,962	0,003	0,000	0,914	0,174	0,153	0,064	0,174	0,153	0,064

Source: Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers.

Naïve Bayesian and BayesNet performed with errors under 0.1; similar result was also in the Finance, Insurance and Real Estate sector. LIBLINEAR and LibSVM performed with smaller Type I errors for "bankrupt" and "average" companies than Naïve Bayes and BayesNet; however, the latter two showed better results for "healthy" companies' recognition. Usage of Bayesian method based resulted in much more balanced Type I and Type II errors. These results indicate that SVM based classifier might effectively predict "bankrupt" and "average" companies but Bayesian classifiers might be more suitable for general classification as they better separate instances on all classes.

Table 16 shows that the possibility to misidentify "healthy" debtors as "bankrupt" is significantly high for SVM based classifiers compared to the opposite (except Finance, Insurance and Financial Services sector). Bayesian classifiers performed with this error respectively below 0.25 for both "bankrupt" and "healthy" company incorrect identification. As in Type I error cases, their results were more

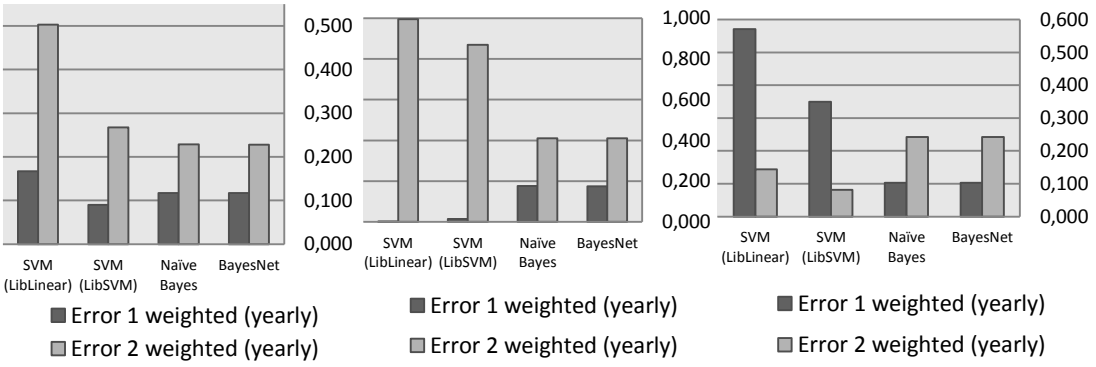
even; the smallest errors were in cases of Mining and Transportation, Communications, Electric, Gas and Sanitary Services sectors.

Table 16. Type II error values for different classes

	LIBLINEAR			LibSVM			Naïve Bayes			BayesNet		
	B	**A**	**H**	**B**	**A**	**H**	**B**	**A**	**H**	**B**	**A**	**H**
01-09	0,154	0,875	0,057	0,615	1,000	0,086	0,000	0,857	0,130	0,000	0,857	0,116
10-14	0,822	1,000	0,010	0,376	0,966	0,047	0,175	0,474	0,217	0,179	0,487	0,217
15-17	0,647	0,800	0,051	0,882	0,800	0,174	0,422	0,255	0,136	0,422	0,309	0,145
20-39	0,530	1,000	0,012	0,290	1,000	0,024	0,269	0,430	0,247	0,269	0,429	0,248
40-49	0,899	0,991	0,011	0,335	0,594	0,065	0,164	0,295	0,377	0,164	0,298	0,376
50-51	0,517	1,000	0,008	0,400	0,962	0,037	0,249	0,308	0,235	0,243	0,308	0,235
52-59	0,756	1,000	0,003	0,561	1,000	0,014	0,288	0,258	0,166	0,283	0,266	0,168
60-67	0,005	1,000	0,695	0,041	0,606	0,262	0,208	0,360	0,204	0,206	0,360	0,203
70-89	0,606	0,993	0,012	0,252	0,934	0,046	0,195	0,517	0,244	0,195	0,511	0,245
All data	0,949	1,000	0,001	0,886	1,000	0,003	0,203	0,416	0,291	0,202	0,416	0,291

Source: Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers.

For overall evaluation weighted mean errors of all these three classes were calculated; they are given in Figure 17. They illustrate that SVM based classifiers (especially LibSVM) might better identify "bankrupt" and "average" companies, but "healthy" companies might be better recognized by Bayesian based classifiers.



Source: Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers.

Figure 17. Weighted mean error values for all classes (bankrupt, average, healthy)

However, weighted mean does not precisely reflect the situation because of the "outstanding" Finance, Insurance and Real Estate sector thus weighted mean errors excluding this sector were also calculated. These values, together with "original" (including errors of Finance, Insurance and Real Estate sector) are given in Table 17. It shows that after excluding the "outlying" Finance sector weighed mean Type I error in case of "bankrupt" class is reduced from 0.167 to 0.013 for LIBLINEAR; LibSVM error is reduced from 0.09 to 0.039 which means that LIBLINEAR classifier might be a good choice for "bankrupt" company identification. However, weighted mean Type I error significantly increased for

"healthy" class for the latter classifiers; this exclusion didn't affect Bayesian-based classifiers as their performance was almost even for all sectors. After excluding Finance sector results Weighted Type I Error increased for SVM based classifiers but these results didn't change significantly for Bayesian classifiers.

Table 17. Weighted mean error values for different classes ("bankrupt", "average", "healthy")

	LIBLINEAR			*LibSVM*			*Naïve Bayes*			*BayesNet*		
	B	*A*	*H*	*B*	*A*	*H*	*B*	*A*	*H*	*B*	*A*	*H*
Including Finance, Insurance and Real Estate sector (60-67) results												
Error 1 weighted	0,167	0,001	0,571	0,090	0,015	0,349	0,117	0,176	0,103	0,117	0,176	0,103
Error 2 weighted	0,503	0,995	0,144	0,268	0,869	0,081	0,229	0,411	0,242	0,229	0,411	0,242
Excluding Finance, Insurance and Real Estate sector (60-67) results												
Error 1 weighted	0,013	0,001	0,705	0,039	0,012	0,421	0,124	0,183	0,106	0,124	0,182	0,106
Error 2 weighted	0,622	0,994	0,012	0,322	0,932	0,038	0,234	0,423	0,252	0,233	0,423	0,252

Source: Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers.

Proposed approach of "slicing" data to sectors and usual training using full dataset with all data was also compared to do overall evaluation; the results of classifiers trained using all data are presented in Table III and Table IV. They show that weighted accuracy achieved while training as separate classifier for each sector after division to sectors was significantly higher; Type I errors also differ. Table 4 shows that "bankrupt" companies were identified with Type I Error in range 0.001-0.003 by using SVM classifiers; Bayesian based methods performed with Type I error of 0.174 respectively.

The experiment results show that LibSVM (C-SVC) in many cases outperformed other classifiers; however, LIBLINEAR performed best while identifying particularly "bankrupt" companies. The experiment results show that "bankrupt" and "average" companies were identified in most of the cases; however it showed that SVM based classifiers do not seem to perform well when the dataaset is very unbalanced. Naïve Bayes and BayesNet performed with more balanced errors, together with the smallest weighted Type I errors.

4.1.4. Model development using various support vector based classifiers[11]

As it is mentioned in Section 2.2.2, many algorithms and their implementations based on SVM were developed at the time of writing this work. It is also concluded that these implementations are very heterogeneous (i.e., written in different languages and different packages), which makes their comparative research

[11] This section is based on Danenas P., Garsva G., Gudas S. Credit Risk Evaluation Model Development Using Support Vector Based Classifiers.

very complicated. Yet, an empirical research on a subset of data was carried to compare them in terms of accuracy and training time. The dataset used in this research consisted of entries from 1354 USA service companies with their 2005-2007 yearly financial records (balance and income statement) from financial EDGAR database. Each instance had 59 financial attributes. This dataset was also transformed into differences between ratios using Step 4 in Algorithm 5; both datasets with absolute differences and differences in percents were created. This step was applied in order to transform dataset into type of data usually used in financial analysis. Table 23 presents main characteristics of original and transformed datasets.

Table 18. Main characteristics of datasets used in experiments

	Original dataset		*Reduced dataset*	
	No of entries	No of attributes	No of entries	No of attributes
Original dataset	3266	59	3266	13
Dataset with absolute differences	1912	59	1912	10
Dataset with percentage differences	1912	59	1912	11

Source: Danenas P., Garsva G., Gudas S. Credit Risk Evaluation Model Development Using Support Vector Based Classifiers.

To select the most important ratios feature selection was also applied for these datasets by using correlation-based feature subset selection algorithm with tabu search for search in attribute subsets. The subsets of selected features turned out to be different; characteristics of obtained datasets are also presented in Table 18.

The experiment was performed using several SVM implementations: original LibSVM 2.91 package by Chang and Lin, LibLINEAR 1.6 by Lin et. al., WEKA implementations of SMO, SGD and Pegasos, LibCVM 2.2 by Tsang et. al, RapidMiner mySVM implementation and multiclass version of SVM^{Light} by Joachims. 70% of data was selected as training data and the remaining 30% - as testing data. Default parameters were used to train classifiers. For first part (three datasets with 59 attributes each) RBF kernel was selected experimentally for all nonlinear SVM classifiers, yet CVM, CVM-LS and BVM classifiers were also applied with Laplacian, inverse distance and inverse square distance functions as kernels. Discriminant analysis was applied for evaluation of financial instances and dynamic formation of bankruptcy classes. The possibilities of feature selection application were also researched by applying correlation-based feature subset evaluator and Tabu search. As some of the classifiers (such as SGD, Pegasos, mySVM) were implemented to solve only binary problems, “1-vs-1” method to use

for transforming the multi-class problem into several 2-class ones was applied, thus training SVM classifiers as a set of binary classifiers. Several classifiers targeted at binary classification (e.g., mySVM) produced several sets of support vectors, the number of produced support vectors was marked as N/A (not available).

Table 19. Results of full dataset

Classifier	Kernel function/SVM type	Dataset with ratios			Abs. difference dataset			Perc. difference dataset		
		Training time	No SV	Acc	Training time	No SV	Acc	Training time	No SV	Accuracy
LibLINEAR	L2-reg. L1-loss SVC	0,710	-	**83,528**	0,430	-	81,882	0,860	-	79,965
C-SVC	RBF	0,610	570	79,155	0,510	524	82,753	0,510	494	82,753
SMO	polynomial	0,844	-	82,449	1,562	-	82,927	0,719	-	82,753
SMO	Pearson	13,407	898	**83,061**	3,297	1168	82,753	3,781	1217	82,753
Pegasos	-	3,313	-	**84,286**	2,047	-	82,23	1,922	-	81,882
SGD	-	3,812	-	**83,265**	2,234	-	**83,275**	2,172	-	82,753
mySVM	RBF	3,812	N/A	82,959	1,328	N/A	82,927	1,203	N/A	**83,972**
CVM	RBF	0,875	607	78,571	1,515	1043	82,753	2,172	1112	82,753
CVM	Laplacian	1,094	783	78,571	0,906	1104	82,753	1,125	1175	82,753
CVM	inverse distance	0,672	801	78,571	0,766	1125	82,753	0,969	1183	82,753
CVM	inverse square distance	0,735	620	78,571	1,312	1033	82,753	1,844	1098	82,753
CVM-LS	RBF	2,344	846	78,571	3,031	1106	82,753	4,891	1118	82,753
CVM-LS	Laplacian	13,453	1600	78,571	9,781	1338	82,753	12,844	1338	82,753
CVM-LS	inverse distance	13,437	1600	78,571	8,828	1338	82,753	8,781	1338	82,753
CVM-LS	inverse square distance	2,453	942	78,571	2,828	1089	82,753	3,344	1096	82,753
BVM	RBF	1,891	691	78,571	2,078	1066	82,753	2,734	1121	82,753
BVM	Laplacian	0,750	728	78,571	0,781	982	82,753	1,218	1063	82,753
BVM	inverse distance	0,672	743	78,571	0,688	992	82,753	1,031	1064	82,753
BVM	inverse square distance	1,875	681	78,571	1,890	1026	82,753	2,360	1093	82,753
SVMLight	RBF	22,48	1	78,570	14,80	1	82,93	15,61	1	82,23
Average:				**80,006**			**82,736**			**82,605**

Source: Danenas P., Garsva G., Gudas S. Credit Risk Evaluation Model Development Using Support Vector Based Classifiers

Linear SVM classifiers (LIBLINEAR, SGD, Pegasos) produced a set of weights instead of a set of SV, thus the number of SV was not included in their cases (although weights with more than 0 can be described as support vectors according to SVM definition). Linear SVM classifiers and nonlinear classifiers with linear, RBF and polynomial kernels were applied for reduced datasets. Table 19 gives results in terms of accuracy, training time and number of SV obtained after performing the described procedure to original data, whereas Table 20 presents the results obtained with "reduced" datasets. The test was run on a 2GHz Pentium Dual Core CPU PC with 3 GB of RAM. Metrics, presenting accuracy (classification accuracy), generated model complexity (no of support vectors) and training time, were chosen for evaluation[12].

The results show that feature selection performed an important part in

[12] As classifiers are written in different programming languages (SMO, Pegasos and mySVM are implemented in JAVA, others in C/C++), execution speed should not be viewed as the main factor in evaluation

construction process, as training time decreased from 4 to 10 times or more, yet accuracy was even better than performed with original data, which shows that the eliminated variables were not the most relevant. The results obtained while training original untransformed dataset varied; however, linear and gradient descent SVM classifiers, such as Pegasos, L2-regularized L1-loss SVC and SGD generated results with highest accuracy (84,286, 83,528 and 83,265 respectively). RBF kernel selection with nonlinear SVM was also a good choice as it obtained better results than nonlinear classifiers with other kernels.

Table 20. Results of experiment with reduced data

Classifier	Kernel/type	Dataset with ratios			Abs. difference dataset			Perc. difference dataset		
		Time	#SV	Acc.	Time	#SV	Acc.	Time	#SV	Acc.
LibLINEAR	L2-regularized L2-loss dual SVC	0,170	-	82,755	0,040	-	82,753	0,080	-	82,578
LibLINEAR	L2-regularized L2-loss primal SVC	0,080	-	82,653	0,030	-	82,753	0,030	-	82,753
LibLINEAR	L2-regularized L1-loss SVC	0,190	-	82,041	0,050	-	82,753	0,110	-	82,753
LibLINEAR	Crammer-Singer multi-class SVC	0,220	-	82,653	0,050	-	82,753	0,120	-	82,753
LibLINEAR	L1-regularized L2-loss SVC	0,640	-	82,449	0,130	-	82,753	0,200	-	82,753
C-SVC	Linear	0,420	797	82,245	0,110	472	82,753	0,110	484	82,753
C-SVC	polynomial	0,270	837	81,735	0,110	450	82,753	0,130	491	82,753
SMO	Pearson	32,61	1047	83,061	7,16	743	82,753	6,94	853	82,404
SMO	RBF	25,45	559	81,327	3,09	335	82,753	2,19	330	82,753
Pegasos	-	1,015	-	82,551	0,5	-	83,101	0,515	-	82,578
SGD	-	1,156	-	82,653	0,547	-	83,101	0,656	-	82,753
mySVM	dot	0,266	N/A	81,837	0,078	N/A	83,101	0,094	N/A	82,753
mySVM	anova	0,250	N/A	82,959	0,079	N/A	83,972	0,093	N/A	**83,101**
CVM	linear	19,891	1504	**83,163**	1,438	1338	82,753	6,156	1330	82,578
CVM	RBF	2	1048	82,755	1,984	1260	81,882	1,813	1282	81,533
CVM	normalized polynomial	3,859	1435	82,959	1,609	1331	**83,450**	1,718	1324	82,927
CVM	inverse square distance	1,563	984	**83,163**	1,656	1245	81,707	1,703	1282	81,359
CVM-LS	linear	54,469	1510	**83,061**	2,625	1336	82,753	18,734	1329	82,578
CVM-LS	RBF	4,578	1004	82,653	4,250	1256	82,056	4,406	1291	81,359
CVM-LS	normalized polynomial	8,563	1417	82,959	4	1335	83,275	3,656	1331	82,927
CVM-LS	inverse square distance	3,765	987	**83,061**	3,984	1238	81,882	4,218	1289	81,359
BVM	RBF	4,125	1121	82,959	2,922	1268	81,882	3	1308	81,359
BVM	Laplacian	1,297	980	81,939	0,875	1057	80,836	0,890	1083	82,404
BVM	normalized polynomial	6,187	1466	**83,265**	3,219	1331	83,450	3,234	1324	82,753
BVM	inverse distance	1,094	999	81,939	0,750	1059	81,185	0,765	1088	82,404
BVM	inverse square distance	3,578	1032	82,959	2,438	1244	81,882	2,640	1303	81,185
SVMLight	linear	0,08	2	81,330	0,03	2	82,750	0,03	2	82,750
SVMLight	RBF	73,53	2	81,330	20,19	2	82,750	20,42	2	82,750
Average:				82,515			82,591			82,417

Source: Danenas P., Garsva G., Gudas S. Credit Risk Evaluation Model Development Using Support Vector Based Classifiers.

The results, obtained after training data with differences, show that best results were obtained by mySVM based classifier, using RBF kernel; usage of linear classifiers SGD and Pegasos also resulted in results which are above 83%. Other results were similar which shows that kernel function selection did not have much influence, especially in case of dataset with percentages where similar results were

obtained with all classifiers. This is why most results obtained with transformations of original dataset were above average results of all classifiers.

However, CVM and BVM based classifiers outperformed others when performing classification procedure on reduced dataset; the choice of linear, inverse square distance and inverted polynomial kernels would be the best here although they are not fastest. Selection of linear kernel and linear classifiers resulted in good performance in terms of accuracy and training time in all three cases here as the results were above average results of tested classifiers. The results for the last two datasets (consisting of differences, together with applied feature selection), of almost all classifiers vary; CVM and BVM classifiers with normalized polynomial kernel function outperformed others, although they were not as fast as linear classifiers. Thus CVM and BVM classifiers are a good choice for further model development as they are trained fast enough (although not as fast as linear SVM classifiers) and produce high accuracy. However, they also produce the highest number of SVs (highest complexity); yet, the tests of their authors show, that the number of SV does not tend to grow less, compared to other SVM implementations (particularly LibSVM or SimpleSVM)[217].

4.1.5. Conclusions

Experimental results showed that feature selection often resulted in even better accuracy, i.e., re-selected features helped to obtain better performance. It is important that the number of features selected by feature selection procedure is different than the ones used by original evaluators, as this shows that other features were also considered as important. These results can be improved by applying heuristic or metaheuristic parameter selection techniques which is extensively discussed in Section 2.1.10 and Section 2.2.7. Another possible improvement is data imbalancement which becomes relevant in cases of dynamical class formation. This is crucially important in identification of bankrupt companies if they are represented by minority entries (which is often the case in real-time situations); as mentioned before, identification of bankrupt company might cost more to the creditor than the misidentification of it.SVM is one of the machine learning techniques which is sensitive to dataset imbalance as “majority” classes tend to outweigh “minority” classes by pushing classification boundaries over them. Many techniques are applied to overcome this barrier: internally implemented class-weighting, cost-sensitive

learning and evaluation, also combined with SVM [157, 209], internal classifier enhancements [143, 208], as well numerical sampling techniques, such as bootstrap, undersampling, oversampling and etc. van Hulse et al. [224] showed that minority random oversampling technique could be the best choice for SVM classifier; however, similar improvements were not successful in presented experiments.

4.2. Empirical evaluation of PSO-LinSVM algorithm

An experiment to analyse classification effectiveness of PSO-LinSVM algorithm defined in Algorithm 6 was run using Australian and German datasets' Appendix J gives detailed specification of these datasets. The proposed technique was compared with similar SVM implementations such as LibSVM and LS-SVM. The experiment was run using MATLAB 2010b environment, using LibSVM 3.12, LibLINEAR 1.8 and LS-SVMlab 1.8 toolboxes. Their parameter selection was implemented using simulated annealing algorithm in MATLAB's Optimization toolbox, while PSO optimization was developed using PSO toolbox for MATLAB by Sam Chen[13]. Two approaches for fitness evaluation were applied, in order to obtain a classifier with best classification performance capabilities in both balanced and unbalanced classification conditions:

- Accuracy, obtained using *k-fold* cross-validation (further referred as CV optimization);
- The sum of specificity and sensitivity (identical for sum of TP ratios in case of binary classification), also obtained using k-fold cross-validation (this principle further referred as balanced CV optimization). This is an approach used in [60, 62].

For experiment tasks, $k = 5$ was selected (although if dataset is large, $k = 2$ or $k = 3$ can be considered as a better choice). In order enable comparison of various optimization approaches, a direct parameter search procedure was run, using seven linear SVM classifiers in LibLINEAR 1.8 toolbox. p_1 is represented by value $-s$ of its training command, i.e., the inner encoding of algorithm in LIBLINEAR package, as it is defined in Section 3.1.2.

Figure 18 and Figure 19 give result visualizations obtained after performing classification tasks. These figures show performance results in terms of accuracy, where each point is represented as $(C;\ bias;\ max(acc_i))$, with

[13] Another Particle Swarm Toolbox, http://www.mathworks.com/matlabcentral/fileexchange/25986-another-particle-swarm-toolbox

$0 < C \leq 100, 0 \leq bias \leq 10, 0 \leq i \leq 6$ and acc_i as accuracy obtained by each classifier using particu lar cost and bias values. C parameter change step was set to 5, whereas bias parameter was set to change by 1.

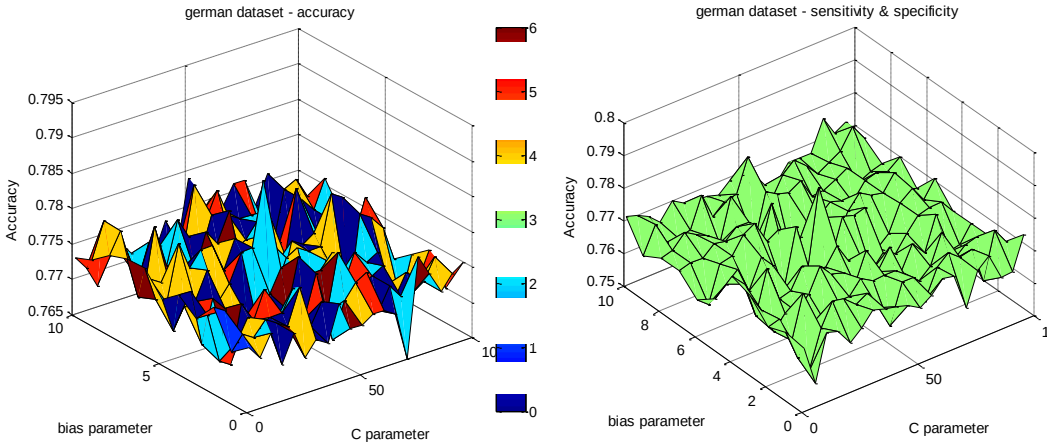


Figure 18. Linear SVM classifier results (German dataset)

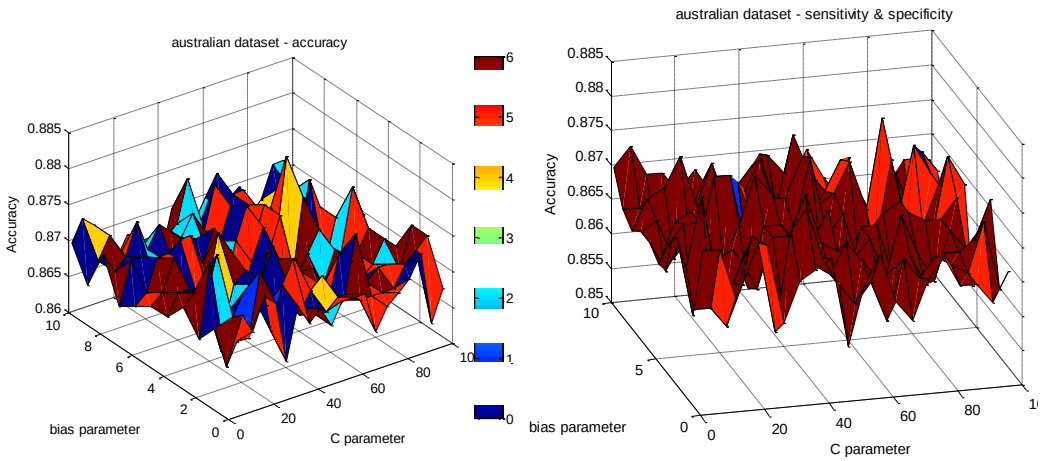


Figure 19. Linear SVM classifier results (Australian dataset)

Such representation visualizes possible search surface and thus helps to identify core parameters for optimization procedure. These figures also show that accuracy selected as evaluation metrics resulted in wide range of classifiers, i.e., none of the classifiers could be identified as the most effective solution whereas in case of balanced cross-validation (using sum of sensitivity and sensitivity) based evaluation single classifiers (dual L2-regularized L1-loss support vector classification and dual L2-regularized logistic regression) dominated. Note that although such results might indicate, that these classifiers might be the best choice for working with imbalanced datasets, optimal classifiers are not necessary among the ones which are given in this visualization.

In order to compare proposed approach with similar classification techniques, similar SVM (particularly C-SVC classifier from LibSVM package) and LS-SVM classifiers were also developed by performing heuristic parameter selection on their kernel functions using Simulated Annealing and Particle Swarm Optimization algorithms previously described in Section 2.1.10.2 and Section 2.1.10.4. C-SVC was run using with RBF (further referred as LibSVM^{RBF}) and sigmoid (further referred as LibSVM^{Sigmoid}) kernel functions (polynomial kernel function was not selected of relatively large parameter space and slow performance) whereas LS-SVM classifiers were based on polynomial (further referred as LS-SVM^{Poly}) and RBF (LS-SVM^{RBF}) kernels. Dataset split 7:3 (i.e., 70% of data was selected for classifier training and optimization procedure, the rest 30% were used for testing) widely used in such research was selected. SA procedure was run using exponential temperature (*temperatureexp*) and *simulannealbnd* functions in 180 iterations whereas PSO implementation was applied with default parameters. c_1 *and* c_2 parameter values were selected experimentally. LIBLINEAR classifier with similar approach to PSO-LinSVM (used in [60, 62]) was also tested; the main difference lies in its design as it is based on real-valued PSO implementation instead of hybrid proposed in Algorithm 6.

Table 21. German dataset results

	Optimization based on accuracy					Optimization based on balanced accuracy				
	Linear classifier code	Cost parameter	Error Rate	TPR 1	TPR 2	Linear classifier code	Cost parameter	Error Rate	TPR_1	TPR_2
LIBLINEAR + DS	0	46	0.214	0.897	0.527	0	46	0.214	0.897	0.527
PSO-LinSVM	3	14.808	**0.187**	0,894	0,634	3	14.808	**0.187**	0,894	0,634
Particle Swarm Optimization										
LIBLINEAR	5	99,274	0,197	0,894	0,602	7	96,112	0,233	0,797	0,699
LibSVM^{RBF}	-	0,014	0,197	0,903	0,591	-	0,016	0,217	0,874	0,581
LibSVM^{Sigmoid}	-	2,885	0,247	0,889	0,581	-	11,406	0,380	0,720	0,613
LS-SVM^{Poly}	-	8,659	0,330	0,763	0,462	-	2,859	0,490	0,536	0,452
LS-SVM^{RBF}	-	4,674	0,217	0,889	0,548	-	3,944	0,240	0,870	0,516
Simulated Annealing										
LIBLINEAR	6	76,788	0,203	0,884	0,602	5	85,577	0,200	0,874	0,634
LibSVM^{RBF}	-	0,013	0,207	0,889	0,581	-	0,012	0,200	0,889	0,602
LibSVM^{Sigmoid}	-	19,520	0,297	0,966	0,140	-	10,932	0,273	0,908	0,387
LS-SVM^{Poly}	-	9,969	0,357	0,720	0,473	-	0,138	0,363	0,705	0,484
LS-SVM^{RBF}	-	2,198	0,240	0,894	0,462	-	5,752	0,230	0,870	0,548

Table 21 shows results obtained with German dataset. Surprisingly, optimized linear SVM classifiers showed best performance, only LibSVM with RBF kernel function showed similar results. These classifiers also showed better performance than LIBLINEAR + DS which proves necessity of such techniques in hybrid models.

PSO-LinSVM, proposed in Algorithm 6, showed even better performance in both cases of accuracy and balanced CV based optimization.

Table 22. Australian dataset results

	Optimization based on accuracy					Optimization based on balanced accuracy				
	Linear classifier	Cost parameter	Error Rate	TPR_1	TPR_2	Linear classifier	Cost parameter	Error Rate	TPR_1	TPR_2
LIBLINEAR+DS	5	6	**0.122**	0.864	0.896	5	6	**0.122**	0.864	0.896
PSO-LinSVM	1	33,606	**0.126**	0,853	0,901	1	33,606	**0.126**	0,853	0,901
Particle Swarm Optimization										
LIBLINEAR	6	15,401	0,164	0,905	0,747	6	64,458	0,169	0,862	0,791
$LibSVM^{RBF}$	-	0,020	0,184	0,914	0,692	-	0	0,150	0,785	0,934
$LibSVM^{Sigmoid}$	-	20	0,169	0,905	0,736	-	9,159	0,159	0,922	0,769
$LS\text{-}SVM^{Poly}$	-	3,850	0,430	0,655	0,462	-	3,201	0,430	0,690	0,418
$LS\text{-}SVM^{RBF}$	-	9,972	0,164	0,879	0,780	-	10,327	0,164	0,879	0,780
Simulated Annealing										
LIBLINEAR	7	0,005	0,159	0,905	0,758	2	45,864	0,155	0,871	0,813
$LibSVM^{RBF}$	-	0,057	0,213	0,888	0,659	-	0	0,150	0,785	0,934
$LibSVM^{Sigmoid}$	-	0,119	0,179	0,897	0,725	-	16,945	0,159	0,871	0,890
$LS\text{-}SVM^{Poly}$	-	0,010	0,164	0,914	0,736	-	0,4	0,193	0,897	0,692
$LS\text{-}SVM^{RBF}$	-	2,656	0,159	0,879	0,791	-	4,363	0,145	0,897	0,802

Table 22 presents experimental results obtained with Australian dataset. Again, linear SVM classifiers outperformed other SVM classifiers. $LS\text{-}SVM^{RBF}$ showed similar results, while it outperformed other classifiers in balanced CV based optimization. PSO-LinSVM with L2-regularized L2-loss support vector classification (dual) classifier again proved to be best choice; however, direct search resulted in highest performance.

Note that in both experiments PSO-LinSVM obtained the same classifiers in both cases of fitness based on accuracy and balanced CV evaluation. The second case also proved to be a reasonable choice in general – both Table 21 and Table 22 show that application of such this approach often resulted in increased accuracy compared to accuracy-based optimization; this is especially seen with classifiers developed using Simulated Annealing approach. This approach identified best parameter sets for classifiers better than or equal to accuracy-based evaluation approach in almost all cases for Australian dataset except $LS\text{-}SVM^{Poly}$ (note that this classifier also did not perform well with PSO optimization technique for this dataset).

It is important to note that the sum of specificity and sensitivity is largest in both cases of proposed classifier. Thus it can be concluded that proposed approach helped to obtain best separation of "positive" and "negative" classes.

4.3. Experiments on classification based on FS-SVM^{SWTest}

This section presents experimental results of FS-SVM^{SWTest} algorithm, given in Section 3.1.3. Various classifiers are applied in this research, including canonical SVM, BPNN and classification technique proposed in Section 3.1.2.

4.3.1. Comparative analysis of BPNN and SVM performance

This section gives results based on BPNN (backpropagation ANN) and canonical (non-linear) SVM, with forecasting on next two periods. The experiments were made by using data of Services sector from EDGAR database from year 1999-2008. It consists of yearly financial records with primary and derived 79 financial ratios and rates used in financial analysis. Table 23 presents main characteristics of data in this sector.

Altman's Z-Score was chosen for evaluation and formation of class variable. Feature selection was applied for each formed dataset using correlation-based feature subset evaluator [96] with Tabu search to select the most relevant financial ratios. The experiment was run using LibSVM [36] and MultilayerPerceptron (Backpropagation Neural Network with sigmoid transfer function) implementations in Weka [236] software. As datasets were unbalanced (there were classes which dominated over others), following procedures were applied in order to improve performance and detection of minority classes:

Table 23. Main characteristics of data used in experiment

Year	Entries labeled as			Total entries	Bankrupt 1 year after	Bankrupt >1 year after
	Bad	Average	Healthy			
1999	754	40	152	946	6	6
2000	884	59	195	1138	6	5
2001	977	62	224	1263	6	2
2002	1017	69	194	1280	7	4
2003	1037	83	265	1385	2	3
2004	1078	102	315	1495	4	9
2005	1149	96	349	1594	9	26
2006	919	71	366	1356	-	8
2007	917	74	356	1347	2	57
2008	113	21	175	309	5	5
Total	8845	677	2591	12113		

- Weights were computed and applied for LibSVM using

$$\forall C_i : w_i = \frac{\max(n_C)}{n_{Ci}}, i = 1..k \tag{4.1}$$

k is the number of classes, C is a set of classes, n_{Ci} –number of instances that belong

to class C.

- SMOTE (Synthetic Minority Oversampling TEchnique) [41] was applied with the sampling percentage calculated using

$$\forall C_i : p_i = \left(\frac{\max(n_C)}{\min(n_C)} - 1 \right) * 100, i = 1..k \tag{4.2}$$

where p_i is the ratio of sampling for minority class.

Model parameters were selected iteratively after running a series of experiments: LibSVM was run using polynomial and RBF kernels with $c \in [10;100]$ using step of 10, degree $d \in [2;5]$ and $\gamma \in [2;4]$ with a step of 1. MLP was run with different numbers of hidden layers $l \in [1;10]$ and using parameter $learningRate \in [0.1;0.9]$ with a step of 0.1. To properly evaluate overall performance, weighted mean was used.

The results of these experiments using data are given in Table 24. It gives results of models which performed best in terms of testing accuracy with data of next two years (e.g., in case of training with data from first *m* periods, testing was performed with *m+1* and *m+2* period data respectively), as well as main parameters of these models, such as kernel function, degree (*-D*), gamma (*-G*), C (cost/complexity) parameters (*-C*) for C-SVC and learning rate (*-L*), momentum (*-M*) and number of hidden layers (*-H*) for BPNN. It is important to note that models that performed better in both periods were chosen more preferably than those which slightly outperformed them in one period but underperformed in another. Thus the main factor for choosing models as best was the sum of their accuracy ratios for both periods.

Table 24. Results of ANN and SVM based classifiers with sliding window testing

Training period	**Multilayer Perceptron**			**SVM (C-SVC)**		
	Parameters	**Year 1**	**Year 2**	**Parameters**	**Year 1**	**Year 2**
2000	-L 0.5 -M 0.2 -H 8	90,427	14,041	Polynomial, -D 3 G 2 -C 60.0	84,889	87,129
2001	-L 0.8 -M 0.8 -H 4	83,541	84,271	Polynomial, -D 2 G 2 -C 30,0	81,045	84,127
2002	-L 0.8 -M 0.8 -H 8	74,892	72,144	Polynomial, -D 3 G 2 -C 20.0	76,912	79,158
2003	-L 0.7 -M 0.7 -H 3	76,420	78,869	Polynomial, -D 2 G 2 -C 10,0	76,286	72,343
2004	-L 0.1 -M 0.1 -H 1	73,897	70,818	RBF, -D 3 G 2 -C 80,0	76,134	70,376
2005	-L 0.8 -M 0.8 -H 2	76,713	75,436	RBF, -D 3 G 2 -C 20,0	80,766	89,317
2006	-L 0.4 -M 0.6 -H 6	82,413	80,906	RBF, -D 3 G 2 -C 60,0	84,666	57,282
2007	-L 0.5 -M 0.5 -H 6	83,172	-	RBF, -D 3 G 2 -C 40,0	75,728	-
Weighted average		**79,533**	**68,682**		**79,753**	**76,745**

The classifiers achieved good results in testing, often in 75-80% range, thus showing good prediction abilities. However, backpropagation neural network

performed very poorly while producing models with data of periods from 1999 to 2000. C-SVC based classifier performed better, showing more even results in all periods. Yet there were few cases (especially in last two periods) where BPNN produced better results.

4.3.2. Experiment using linear SVM without parameter selection

The experiments were made by using data of period 1999-2008 from EDGAR database, Manufacturing sector. The initial dataset used in the experiment consists of yearly financial records with 51 financial ratios used in financial analysis computed using original primary financial data from balance and income statement data. To formulate credit risk problem as classification problem, Zmijewski's developed model was selected as means for evaluation and label formation. One of main reasons for this selection was the origin of the data (the data comes from USA companies).

Table 25. Main characteristics of data used in experiment

Year	Entries labeled as		Total entries	No of selected attributes	Bankrupt 1 years after	Bankrupt >1 year after
	Risky (R)	Not risky (NR)				
1999	1312	537	1849	12	-	-
2000	1869	589	2458	15	0	0
2001	1753	672	2425	15	1	0
2002	1709	777	2486	13	3	0
2003	1770	723	2493	14	0	2
2004	1920	637	2557	13	0	1
2005	1964	660	2624	14	3	17
2006	1636	429	2065	14	0	3
2007	1545	393	1938	14	1	13
2008	483	109	592	14	4	1
Total	**15961**	**5527**	**21487**		**12**	**37**

Table 25 presents main characteristics of dataset, including classes formed by evaluation using Zmijewski's score, together with bankruptcy data from UCLA LoPucki bankruptcy database [219] used to validate the results. UCLA LoPucki database contains bankruptcy data and covers about 50 companies from used dataset. The data from 2000 – 2010 was applied for validation; instances which represent last entry in financial history were marked as "risky" and were evaluated by developed classifiers. The code and algorithms for the experiments were implemented using WEKA machine learning framework with LIBLINEAR 1.7. The test was run using 5 classifiers described in Section 3.3. Cost parameter *C* and bias *b* for these algorithms were chosen experimentally, by using linear search in range of $C \in [0;100]$ and $b \in [0;1]$.

Table 26. Results of the experiment

Training period			*2000*	*2001*	*2002*	*2003*	*2004*	*2005*	*2006*	*2007*
Structure (parameters) of selected classifier			CS-SVM	L1-LSVM (dual)	L1-LSVM (dual)	CS-SVM	L1-LSVM (dual)	L1-LSVM (dual)	L2-LSVM (primal)	L2-LSVM (dual)
C			20	20	20	15	20	15	15	5
Bias			0.7	1.0	0.7	1.0	0.4	0.7	0.7	1.0
Year 1	**Accuracy**		**96,702**	**96,344**	**95,471**	**95,504**	**91,604**	**93,085**	**92,008**	**92,295**
	TP	**R**	0,973	0,974	0,970	0,965	0,974	0,977	0,971	0,981
		NR	0,951	0,940	0,917	0,925	0,745	0,756	0,724	0,675
	FMeas	**R**	0,977	0,973	0,968	0,970	0,945	0,957	0,951	0,954
		NR	0,941	0,942	0,922	0,911	0,818	0,820	0,789	0,770
Year 2	**Accuracy**		**96,183**	**94,233**	**95,348**	**96,785**	**92,940**	**91,445**	**91,960**	-
	TP	**R**	0,966	0,966	0,972	0,983	0,977	0,966	0,977	-
		NR	0,953	0,938	0,898	0,923	0,749	0,716	0,675	-
	FMeas	**R**	0,972	0,970	0,969	0,979	0,956	0,947	0,952	-
		NR	0,940	0,928	0,906	0,936	0,816	0,775	0,762	-
Year 3	**Accuracy**		**96,032**	**96,286**	**96,710**	**97,389**	**91,291**	**92,127**	-	-
	TP	**R**	0,962	0,970	0,987	0,987	0,964	0,981	-	-
		NR	0,956	0,940	0,908	0,923	0,716	0,667	-	-
	FMeas	**R**	0,972	0,975	0,978	0,984	0,946	0,953	-	-
		NR	0,933	0,927	0,933	0,936	0,772	0,764	-	-

Feature selection was applied for each formed dataset using correlation-based feature subset evaluation to select the most relevant financial ratios. Table 26 depicts classification performance in terms of classification accuracy together with TPR and F-Measure rates for each class. Classifiers which resulted in best average testing performance are selected as best.

Surprisingly, the accuracy is above 90%, which can be considered as a very good result; F-Measure values also prove that discriminatory performance of these classifiers was also high. Best results were obtained while training classifier sequentially with data from first five years (starting with year 1999) as classification accuracy remained over 95%. Later it decreased to approx. 91-92%, although the number of instances used for training thus the possibility to improve learning and extraction of inner patterns increased; this might indicate change in trends of instances provided to the classifier. It is important to note that classification results are given for the classifiers which showed best performance overall in all three testing periods, i.e., the total classification performance is maximized instead of paying attention to performance in single periods which sometimes was better compared to the performance of classifiers with best overall performance.

No single classifier dominated among these which showed best results - Crammer-Singer multiclass SVM showed best performance for 2 analyzed cases, L1 dual linear SVM – for 4 cases and L2 linear SVM, both primal and dual – for last two

cases (once per each classifier). The obtained TPR values for both "risky" (R) and "non-risky" (NR) classes can be considered as a good result (both were over 0.9 in first four periods, and over 0.7 in next periods); this shows that instances for both of these classes were identified successfully.

Table 27. Results of the PSO-LinSVM classifier

Training period			*2000*	*2001*	*2002*	*2003*	*2004*	*2005*	*2006*	*2007*
Classifier			L2-RLR (primal)	L2-SVM (dual)	L2-SVM (dual)	L2-RLR	L2-SVM (primal)	L2-SVM (dual)	L2-SVM (dual)	L2-SVM (dual)
C parameter			46,5068	9,4532	20,0452	76,0741	1,0000	32,1152	40,2581	20,4178
Bias parameter			-3,5519	9,5337	3,5257	-0,6641	5,2068	6,7547	2,2369	1,3727
	Accuracy		**95,218**	**95,46**	**87,655**	**94,253**	**91,679**	**93,52**	**86,12**	**90,372**
Year 1	**TP**	**R**	0,984	0,977	0,979	0,976	0,967	0,968	0,857	0,990
		NR	0,869	0,905	0,626	0,841	0,769	0,812	0,878	0,523
	F-Measure	**R**	0,967	0,967	0,918	0,962	0,945	0,959	0,908	0,944
		NR	0,91	0,926	0,747	0,879	0,824	0,839	0,719	0,667
Year 2	**Accuracy**		**93,853**	**95,311**	**89,367**	**94,743**	**92,94**	**91,744**	**86,486**	-
	TP	**R**	0,98	0,972	0,984	0,985	0,969	0,955	0,865	-
		NR	0,847	0,906	0,62	0,836	0,777	0,771	0,862	-
	F-Measure	**R**	0,956	0,967	0,933	0,965	0,956	0,949	0,913	-
		NR	0,896	0,918	0,744	0,89	0,821	0,791	0,701	-
Year 3	**Accuracy**		**93,908**	**95,387**	**90,053**	**96,373**	**91,073**	**92,736**	-	-
	TP	**R**	0,967	0,976	0,993	0,99	0,954	0,969	-	-
		NR	0,87	0,889	0,629	0,863	0,74	0,743	-	-
	F-Measure	**R**	0,957	0,969	0,937	0,977	0,945	0,956	-	-
		NR	0,893	0,906	0,762	0,908	0,771	0,790	-	-

High F-Measure values which are more suitable for unbalanced learning evaluation also prove this. This indicates that imbalanced learning techniques were not necessary for this case, although their integration could be considered as an option to improve performance for training with other data. Parameters *C* and *bias* varied; the experiment showed that bias parameter had significant influence and the performance might depend on proper selection of this parameter.

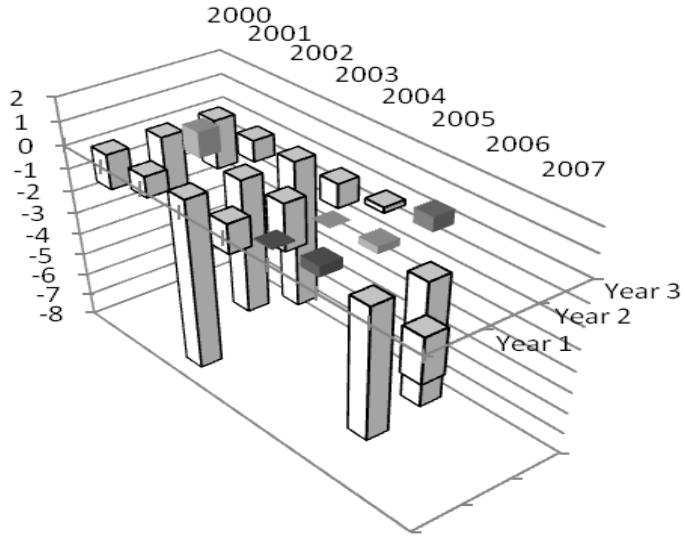


Figure 20. Visual representation of differences between experimental results

Finally, to compare classifier performance with linear parameter selection and PSO-LinSVM classifier, an experiment with the same data was perfomed using PSO-LinSVM classifier. The obtained results show that PSO-LinSVM application for classification resulted in less stable performance; although it is important to note that the latter experiment is conformant to "training-testing-validation" paradigm (i.e., the results presented here are validation results), whereas previous results are selected according to testing results.

Figure 20 depicts differences of results in both experiments - if PSO-LinSVM performance resulted in worse performance than manually selected linear SVM with best performance (i.e., difference beween accuracies of corresponding testing results is less than 0) it is drawn as a transparent bar, otherwise the results is given as filled bar. Results given in Figure 20 indicate that there were several cases when PSO-LinSVM resulted in significantly worse performance when testing accuracy was >5% worse than manual selection. However, these differences were not as significant in other cases; it can also be seen that in some cases accuracy was even better.

4.3.2.1. Identification of actual bankruptcies using proposed techniques

An additional validation step which shows the performance in identification (prediction) of actual bankruptcies was performed to test performance of developed models using real bankruptcy data in Section 4.3.2. The procedure is as follows: if applied dataset was in the period [p_{start}; p_{end}], with year p_{end} as the year of last entry in financial history, bankruptcy is known to be occurred after the financial history, i.e., on year p_{end} +1, p_{end} +2,.., p_{end} +k, with k as the maximum number of years during which the company is officially recognized as bankrupt. Thus the instance in financial history record representing year p_{end} is labelled as "unhealthy", and prediction procedure is performed on the instance. Here it is presumed that bankruptcy might have occurred following the year of the last entry of financial history for particular company, next year or even later (k years after). k = 2 or k = 3 is selected in the experiments; thus bankruptcy fact is evaluated here only if it happens during the next 2 or 3 years after the last entry in financial records of the company.

The results of bankruptcy identification step are given in Table 28. It can be seen that the model developed by the proposed method identified more bankruptcy facts than original Zmijewski model which was used as the evaluator. Table 25 also shows that the number of financial ratios considered relevant by feature selection

procedure is different from the number of features used in original evaluator.

Table 28. Bankruptcy prediction results

Year	Actual bankrupt	Original model (Zmijewski)	Bankruptcies after testing period		
			Year 1	Year 2	Year 3
2002	1	0	0	-	-
2003	3	0	0	0	-
2005	2	0	0	0	0
2006	4	1	1	1	1
2007	1	0	1	0	0
2008	8	6	6	5	5
2009	27	9	18	16	17
2010	3	0	1	1	1
Total:	**49**	**16**	**27**	**23**	**24**

The number of bankruptcies is not large, especially for years from 1999 to 2007, thus the performance results should be viewed carefully; however, the performance in years 2008-2009 shows a reasonable increase in performance and gives more reliability. Empirical results of validation using actual bankruptcy results obtained in [60, 62] proved that such improvements in actual prediction are possible, thus this step is one of main steps in future work, related to this research.

However, such prediction capability should be viewed carefully as classification performance converging to 100% would not result in bankruptcy identification performance increase, compared to original evaluator. Therefore, a trade-off maximization optimization approach for classifier selection can be suggested at this step

$$\max\ w_1 * \sum_{i=1}^{N_C} TPR_i + w_2 * \frac{b^{identified}}{b^{total}} \tag{4.3}$$

$$\text{s.t. } \sum_{i=1}^{N_C} TPR_i >= threshold,\ 0 \le \sum_{i=1}^{N_C} TPR_i \le N_C,\ 0 \le threshold \le N_C,$$

$$b^{identified}, b^{total} \in N$$

where $\sum_{i=1}^{N_C} TPR_i$ is the sum of TPR values obtained by proposed approach, N_C is the number of classes, $b^{identified}$ is the number of bankruptcies identified by the classifier model, b^{total} is the total number of bankruptcies during the period which data is used for training ($b^{identified} \le b^{total}$ as it follows from the definition). Again, the sum of TPR values can be replaced with accuracy, if preferred, yet it should be better suitable for classification of less balanced datasets. Weights w_1 and w_2 are selected experimentally, representing the relevance of either original evaluation accuracy or

actual bankruptcy identification proportion. If $w_1 = w_2$ the problem (4.3) is simplified to

$$\max \sum_{i=1}^{N_C} TPR_i + \frac{b^{identified}}{b^{total}} \tag{4.4}$$

$$\text{s.t. } \sum_{i=1}^{N_C} TPR_i >= threshold,\ 0 \le \sum_{i=1}^{N_C} TPR_i \le N_C, 0 \le threshold \le N_C,$$

$$b^{identified}, b^{total} \in N$$

(4.3) can also be extended for optimization according to testing results in several $n_p < k$ periods

$$\max w_1 * \sum_{i=1}^{n_p} \sum_{j=1}^{N_C} TPR_{ij} + w_2 * \sum_{i=1}^{n_p} \frac{b_i^{identified}}{b_i^{total}} \tag{4.5}$$

$$\text{s.t. } \sum_{i=1}^{N_C} TPR_i >= threshold,\ 0 \le \sum_{i=1}^{N_C} TPR_i \le N_C, 0 \le threshold \le N_C,$$

$$b_i^{identified}, b_i^{total} \in N$$

where TPR_{ij} is testing True Positive Rate value for *j*-th class at *i*-th testing period, $b_i^{identified}$ is the number of bankruptcies identified by the classifier model in *i*-th period, b_i^{total} is the total number of bankruptcies in *i*-th period, n_p is the number of periods selected for testing bankruptcy identification performance. Thus, if $k = n_p + n_v$ periods were selected for modelling in FS-SVM^{SWTest} approach, with n_p as the number of periods applied in optimization problem (4.5), n_v is the number of periods used for classifier testing, the classifier is developed by adopting original evaluator with respect to actual bankruptcy results.

Such approach can viewed as an attempt to improve the original discriminant evaluator, by adapting its discriminant hyperplane with respect to its original dicriminatory abilities and actual bankruptcy instances that were reported. In order to respect the classification abilities of original evaluator, *threshold* constraint parameter such that $0 \le threshold \le N_C$ with higher value, e.g., $0.8 \times N_C$ or $0.9 \times N_C$, can be applied.

4.3.3. Experimental research of PSO-LinSVM and GA-LinSVM[14]

The same algorithm as in Section 4.3.2 but using PSO-LinSVM and GA-LinSVM techniques was applied on a dataset consisting of entries from 785 USA

[14] Results of this research are also presented in: Danenas P., Garsva G. Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach

Transportation, Communications, Electric, Gas, And Sanitary Services companies with their 2005-2007 yearly financial records (balance and income statement) from financial EDGAR database. Each instance has 51 financial attributes (indices used in financial analysis). Main characteristics of the datasets formed for the experiment are presented in Table 29. To select the most important ratios feature selection was also applied for these datasets by using correlation-based feature subset selection [96] algorithm with tabu search for search in attribute subsets. The code and algorithms for the experiments was implemented using Weka framework [236] with LIBLINEAR 1.7 by Lin et. al. JGAP (Java Genetic Algorithms Package)[15] v3.6 and JSwarm 2.08[16] frameworks were used to implement GA and PSO functionality (note, that real valued PSO is used to implement this version of PSO-LinSVM which is different than hybrid version proposed in Algorithm 6).

Table 29. Main characteristics of datasets used in experiments

Year	Entries labeled as		Total entries	No of selected attributes
	Risky (R)	Not risky (NR)		
1999	376	166	542	11
2000	423	192	615	8
2001	383	226	609	13
2002	376	239	615	11
2003	417	220	637	9
2004	460	194	654	9
2005	478	173	651	8
2006	375	118	493	8
2007	367	112	479	11
2008	38	12	50	8
Total	**3693**	**1652**	**5345**	

Source: Danenas P., Garsva G. Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach.

The search space for both GA and PSO was set to $C \in [0;50], bias \in [0;1]$, as well as the number of run iterations was set to 10. The crossover rate for GA was set to 0.7 (70% of the best offsprings were selected after each evaluation iteration to form a new population) and population size was set to 20. PSO was configured to run with 20 particles and inertia rate of 0.8. Minimum velocity for p_2 was set to 3, for p_3 was set to 0.2; maximum velocity for p_2 was set to 3, for p_3 it was set to 0.2.

[15] JGAP: Java Genetic Algorithms Package, http://jgap.sourceforge.net/
[16] JSwarm-PSO: Swarm optimization package, http://jswarm-pso.sourceforge.net/

Table 30. Results of GA-LinSVM

Training period			*2000*	*2001*	*2002*	*2003*	*2004*	*2005*	*2006*	*2007*
Linear classifier			L2-RLR	L2-RLR	L2-RLR	L2-RLR	L2-RLR	L2-RLR	L2-RLR	L2-RLR
C			19,211	36,731	38,904	45,293	48,837	37,752	21,513	9,620
Bias			0,277	0,005	0,820	0,709	0,887	0,006	0,221	0,058
Year 1	Accuracy		**78,431**	**79,870**	**83,046**	**86,280**	**81,098**	**83,603**	**83,711**	**84,000**
	TP	R	0,956	0,957	0,971	0,976	0,946	0,947	0,954	0,974
		NR	0,496	0,550	0,564	0,595	0,446	0,487	0,470	0,417
	F-Measure	R	0,848	0,853	0,882	0,909	0,880	0,898	0,899	0,902
		NR	0,631	0,680	0,697	0,720	0,560	0,589	0,582	0,556
Year 2	Accuracy		**81,656**	**77,865**	**85,823**	**84,146**	**84,008**	**82,887**	**84,000**	-
	TP	R	0,976	0,947	0,978	0,979	0,949	0,946	0,974	-
		NR	0,567	0,459	0,574	0,469	0,496	0,462	0,417	-
	F-Measure	R	0,867	0,849	0,907	0,900	0,900	0,893	0,902	-
		NR	0,706	0,589	0,707	0,615	0,599	0,565	0,556	-
Year 3	Accuracy		**78,336**	**82,165**	**83,537**	**88,664**	**83,505**	**82,000**	-	-
	TP	R	0,971	0,959	0,981	0,984	0,951	0,947	-	-
		NR	0,427	0,497	0,441	0,580	0,470	0,417	-	-
	F-Measure	R	0,854	0,883	0,897	0,929	0,897	0,889	-	-
		NR	0,577	0,624	0,591	0,711	0,579	0,526	-	-

Source: Danenas P., Garsva G. Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach.

Table 30 presents the results obtained by GA-LinSVM classifier – classifier parameters, obtained by Genetic Algorithm, classification accuracy together with True Positive and F-Measure rates for each class. They are satisfiable, although they could be improved – analysis of TP and F-Measure shows that some particular improvements, such as imbalanced learning or classifier search space expansion might be applied in order to improve the performance.

Table 31. Results of real valued PSO-LinSVM implementation

Training period			*2000*	*2001*	*2002*	*2003*	*2004*	*2005*	*2006*	*2007*
Classifier			L1-SVM (dual)	L2-SVM (dual)	L2-SVM (dual)	L2-RLR	L2-SVM (primal)	L2-SVM (dual)	L2-SVM (dual)	L2-SVM (primal)
C			15,316	47,834	24,735	29,049	22,3727	38,086	6,5322	48,0734
Bias			1,000	0,196	0,749	0,797	0,873	0,838	0,436	0,508
Year 1	Accuracy		**77,941**	**78,409**	**80,220**	**83,689**	**80,640**	**83,806**	**82,887**	**82,000**
	TP	R	0,969	0,952	0,981	0,987	0,952	0,957	0,970	0,974
		NR	0,461	0,521	0,464	0,482	0,412	0,462	0,385	0,333
	F-Measure	R	0,846	0,843	0,867	0,895	0,878	0,900	0,896	0,892
		NR	0,609	0,653	0,618	0,637	0,535	0,579	0,520	0,471
Year 2	Accuracy		**80,032**	**77,080**	**84,146**	**83,232**	**83,806**	**84,742**	**82,000**	-
	TP	R	0,979	0,947	0,985	0,990	0,957	0,959	0,974	-
		NR	0,521	0,436	0,503	0,407	0,462	0,496	0,333	-
	F-Measure	R	0,857	0,844	0,897	0,896	0,900	0,905	0,892	-
		NR	0,670	0,568	0,653	0,567	0,579	0,611	0,471	-
Year 3	Accuracy		**77,237**	**80,488**	**83,384**	**86,032**	**84,124**	**84,000**	-	-
	TP	R	0,966	0,952	0,987	0,987	0,967	0,974	-	-
		NR	0,405	0,456	0,418	0,462	0,444	0,417	-	-
	F-Measure	R	0,848	0,873	0,897	0,915	0,902	0,902	-	-
		NR	0,551	0,582	0,576	0,615	0,575	0,556	-	-

Source: Danenas P., Garsva G. Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach.

Notably, selection of much larger C results in slower classifier training. The results show that, surprisingly, L2-regularized logistic regression (L2-RLR) was selected for almost all cases. The analysis shows that classification accuracy increased while providing the classifier with more data each year. However, in 2004 this accuracy slightly decreased although it was relatively stable (testing results of Year 1 show an increasing accuracy trend and testing results for next two years remain above 82%). Table 31 gives the results obtained by PSO optimized linear SVM classifier; however, these results were obtained by classifier developed using PSO implementation described in Algorithm 4. Results obtained with PSO implementation proposed in Algorithm 6 are presented in Table 32. It resulted in similar or even improved results; however, it resulted in very poor performance at initial stage when amount of data for training was considerably small.

Table 32. Results of proposed PSO-LinSVM

Training period			***2000***	***2001***	***2002***	***2003***	***2004***	***2005***	***2006***
Classifier			L2-reg L2-loss SVC	L2-RLR (dual)	L2-RLR	L2-RLR (primal)	L2-reg L2-loss SVC	L2-RLR	L2-reg L2-loss SVC
C			59,0500	36,8314	27,3827	59,8318	16,4779	15,4329	11,0138
Bias			6,6934	1,9119	-2,0739	-2,3890	4,2076	5,9212	0,4691
Year 1	Accuracy		**36,275**	**78,084**	**81,319**	**83,689**	**78,659**	**83,603**	**84,124**
	TP	0	0,003	0,963	0,957	0,948	0,975	0,952	0,973
		1	0,969	0,496	0,541	0,574	0,277	0,471	0,427
	F-Measure	0	0,005	0,843	0,87	0,891	0,87	0,898	0,903
		1	0,531	0,638	0,667	0,677	0,412	0,58	0,565
Year 2	Accuracy		**39,448**	**76,609**	**84,299**	**83,079**	**82,389**	**83,918**	-
	TP	0	0,011	0,962	0,961	0,956	0,989	0,954	-
		1	0,996	0,395	0,564	0,492	0,303	0,479	-
	F-Measure	0	0,021	0,843	0,896	0,892	0,895	0,9	-
		1	0,562	0,539	0,681	0,611	0,453	0,589	-
Year 3	Accuracy		**34,223**	**80,488**	**83,079**	**86,437**	**83,711**	-	-
	TP	0	0,007	0,963	0,965	0,979	1	-	-
		1	0,977	0,431	0,469	0,504	0,325	-	-
	F-Measure	0	0,014	0,874	0,893	0,916	0,903	-	-
		1	0,506	0,568	0,599	0,642	0,49	-	-

Yet, the results also show that there were cases when application of hybrid PSO-LinSVM resulted in slightly better performance than real valued PSO-LinSVM. Still, the results were slightly worse than GA-LinSVM but the choice of classifiers by the algorithm was more diverse than in case of GA-LinSVM. Although it can indicate that L2-RLR was the best choice for the classifier, few cases (in cases of year 2004 and 2005) which resulted in better results than in GA-LinSVM case contradict this fact. This proposes a conclusion that proposed PSO-LinSVM algorithm might still be improved.

4.4. Experiment conclusions

1. Empirical research of SVM classification techniques identified such key findings of Support Vector Machines and their current implementations:

a. Nonlinear SVM often resulted in poor separation and slow performance when large amounts of data (i.e., when there are more than 1000 instances) were used.

b. The choice of proper kernel functions (RBF, sigmoid, polynomial) is complicated and data dependable, especially if larger datasets are used. It can be difficult to apply heuristic selection techniques in real problems as this might need large computational resources.

c. SVM does not perform well on imbalanced datasets; application of additional data balancing techniques can help to improve performance but does not always result in satisfactory results.

d. Linear SVM algorithms do not show satisfactory performance on small datasets; however, their performance on large datasets was better than nonlinear SVM.

e. Optimization of SVM performance according to sum of sensitivity and specificity (or sum of True Positive Rate values in case of multiclass problems) has to be considered if imbalanced datasets are used, as SVM are prone to ignore proper identification of instances which belong to "minority" classes.

f. Empirical ANN and SVM (C-SVC) evaluation proved better performance of C-SVC based classifier.

2. Key findings obtained after empirical research of FS-SVM^{DA} classifier:

a. FS-SVM^{DA} classifier based on nonlinear SVM often obtained overall classification accuracy over 80%; this measure varied in different sectors, in range from 70% to 95%. This can be evaluated as satisfactory result.

b. Linear SVM implementation proved to be an efficient alternative to nonlinear SVM classifiers as it often showed comparative or better results.

c. Research on different SVM classifiers (CVM, CVM-LS, BVM and etc.) showed that other classifiers such as CVM and BVM classifiers with normalized polynomial kernel function can be a good alternative for other classifiers and even outperform them.

d. Additional procedures such as imbalanced learning improvements or parameter selection might be used to improve their performance.

3. Classification based on FS-SVM^{SWTest} proved to be an efficient tool. Classifier based on proposed technique and external evaluators also showed better actual bankruptcy identification results than original discriminatory evaluators which is promising for future research. However, current results should be viewed carefully as more actual bankruptcy facts consistent with experimental data should be used for testing to make more exact conclusions.

4. PSO-LinSVM classifier presented in Section 3.1.2 proved to be an efficient solution in classification problems resulting in better performance than similar SVM classifiers optimized using heuristic techniques. It also resulted in highest sum of sensitivity and specificity which is equal to best quality of separation of instances labeled as “positive” and “negative”.

5. Further experimentation was based on GA-LinSVM and PSO-LinSVM classifiers, which also showed satisfactory classification performance (over 80% or even 90%), although it varied for different sectors (as it can be seen from results given in Appendix M)

5. DECISION SUPPORT SYSTEM FRAMEWORK AND ITS IMPLEMENTATION

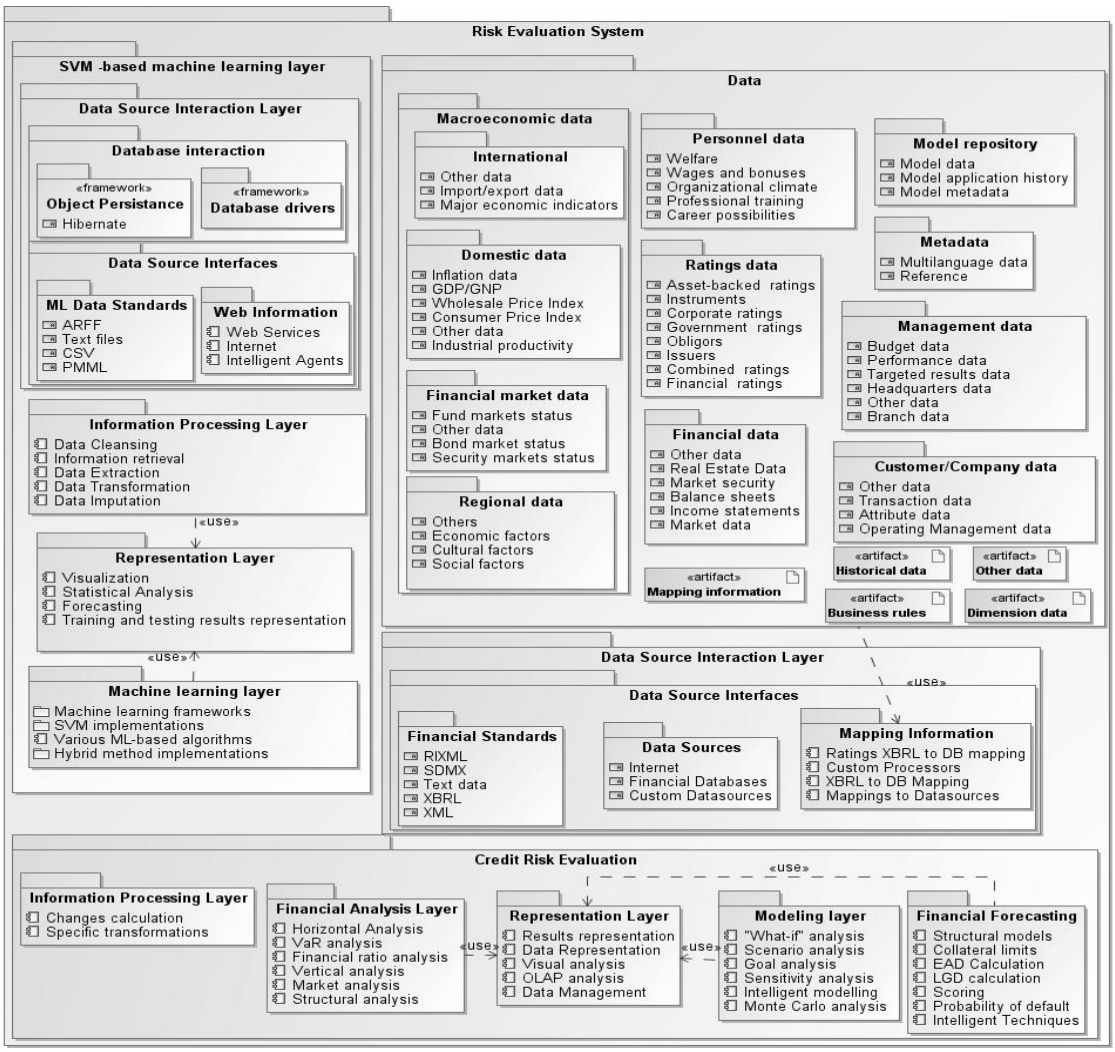
The analysis of DSS structures for credit risk evaluation identified missing integration of complex financial standards. This was one of main incentives to investigate and develop a framework for financial decision support, which implements proposed techniques based on Support Vector Machines (SVM) and eXtensive Business Reporting Language (XBRL)[17]. Possible extensions; design and development methodology based on Domain Driven Design and Feature Driven Development for DSS and implementation scenarios of this system are also discussed.

5.1. The main components of the system

The proposed system can be described as multilayered component model, with each component representing functionality of particular aspect or domain. Such structure enables separation of computational and domain aspects, enables reuse and adoption of developed functionality in other systems or domains. It has main components which were defined in earlier works [65,131,138,159,260] – model repository, data store, reasoning facility and user interface. However, additional capabilities are defined for data retrieval and processing as modern XML-based standards allow creating automatic data import from various sources.

The system's structure is defined as consisting of three main layers – SVM based machine learning layer (further referred as SVM-ML layer) which purpose is to define and implement all the machine learning techniques and algorithms necessary for evaluation as well as other data mining tasks which need to be solved in machine learning process such as information processing, representation. The techniques proposed in Section 3 (FS-SVM, FS-SVM^{SWTest} and PSO-LinSVM) are also included in this layer. Data layer defines data that is available for modeling and stored in data repository facility; and credit risk evaluation layer (further referred as CRE layer) that implements whole analysis, modeling, forecasting and evaluation logic, as well as data visualization. The separation of these aspects gives a possibility to use machine learning techniques implemented in this system to solve other problems by implementing only the logic specific to these problems. The main aspects of this system are also defined as particular layers:

[17] This section is based on Danenas P., Garsva G. SVM And XBRL Based Decision Support System For Credit Risk Evaluation.



Source: P. Danenas, G. Garsva. SVM and XBRL based decision support system for credit risk evaluation.

Figure 21. Layered diagram of designed framework

- Data source interaction layer – it is defined in both SVM-ML and CRE layers. SVM-ML layer interaction sublayer includes database interaction layer with object persistence frameworks and database connection frameworks (drivers), as well as data standards commonly used in machine learning software (such as Weka ARFF, Comma Separated Values (CSV) formats or interoperable Predictive Model Markup Language (PMML) standard). It also defines the interfaces for intelligent information retrieval (not necessarily financial) using Web Services or intelligent agents. CRE layer extends previously described layer with financial standards and data sources specifically for finance or credit risk related tasks. It also has a mapping package that contains the mappings between XBRL (or other standards) and data stored in Data

Layer. The model describing main principles for XBRL integration is given in Section 5.2.

- Information Processing layer – also defined in both SVM-ML and CRE layers. It implements main tasks that are solved during the whole intelligent model creation process before training using one of the algorithms. It defines such standard data mining aspects as information retrieval, data extraction and cleansing, feature selection and data transformation (e.g., PCA, ICA, factor analysis, etc.), normalization/standardization, data imputation. The same layer defined in CRE layer implements specific tasks, e.g., specific transformations, data transformation to absolute or percentage changes between particular ratios during particular period and etc.

- Data Layer – defines all the data that is stored in data store (multidimensional data warehouse, database or other source). The system described here uses company data, financial data (data extracted from financial reports), company management and personnel data, historical records, market data, also macroeconomic and statistical data for macroeconomic environment evaluation (this type of data for analysis is also defined in earlier works [131,138,159]). It also contains metadata, such as reference or multilanguage data, as well as financial ratings and historical informatikon, obtained manually or automatically (e.g., instances of SEC RATINGS taxonomy). The last component is a model repository which contains all intelligent, statistical or other models (including SVM-based or hybrid models), as well as their execution log, evaluation results and their metadata.

- Representation layer - this module includes all methods and operations which are used for representation and visualization of results. It is more generic in case of SVM-ML (defines standard representations of training, testing and prediction results as well as their visualizations). CRE layer defines more sophisticated modules such as OLAP analysis together with representation of financial analysis, simulation/modeling and forecasting as well as data management functionality;

- Financial Analysis, Modeling and Forecasting modules are defined particularly for risk evaluation layer and represent specific business logic for credit risk domain (analytics, simulations, forecasting). The list of techniques is not complete and can be extended using novel recent methods.

Such layer-based functionality can be mapped to use case for intelligent credit

risk DSS given in Figure 34 using UML composition structure diagram; it is given in Figure 22. This enables clear expression of correspondence between tasks and components which have to be developd for their fulfilment.

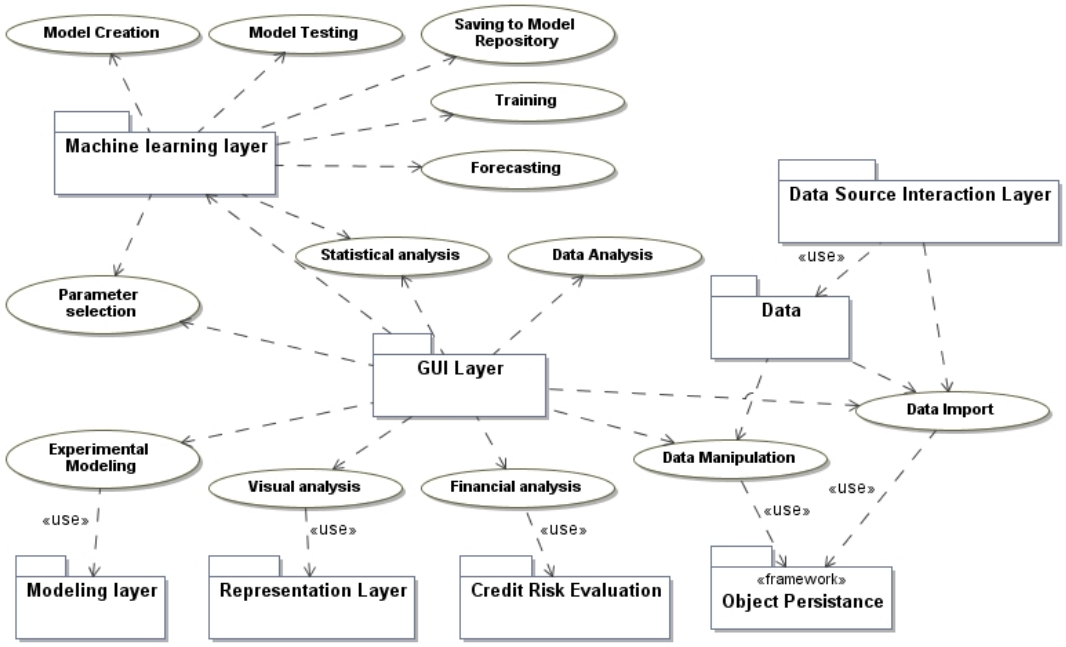


Figure 22. Composition structure diagram for developed framework

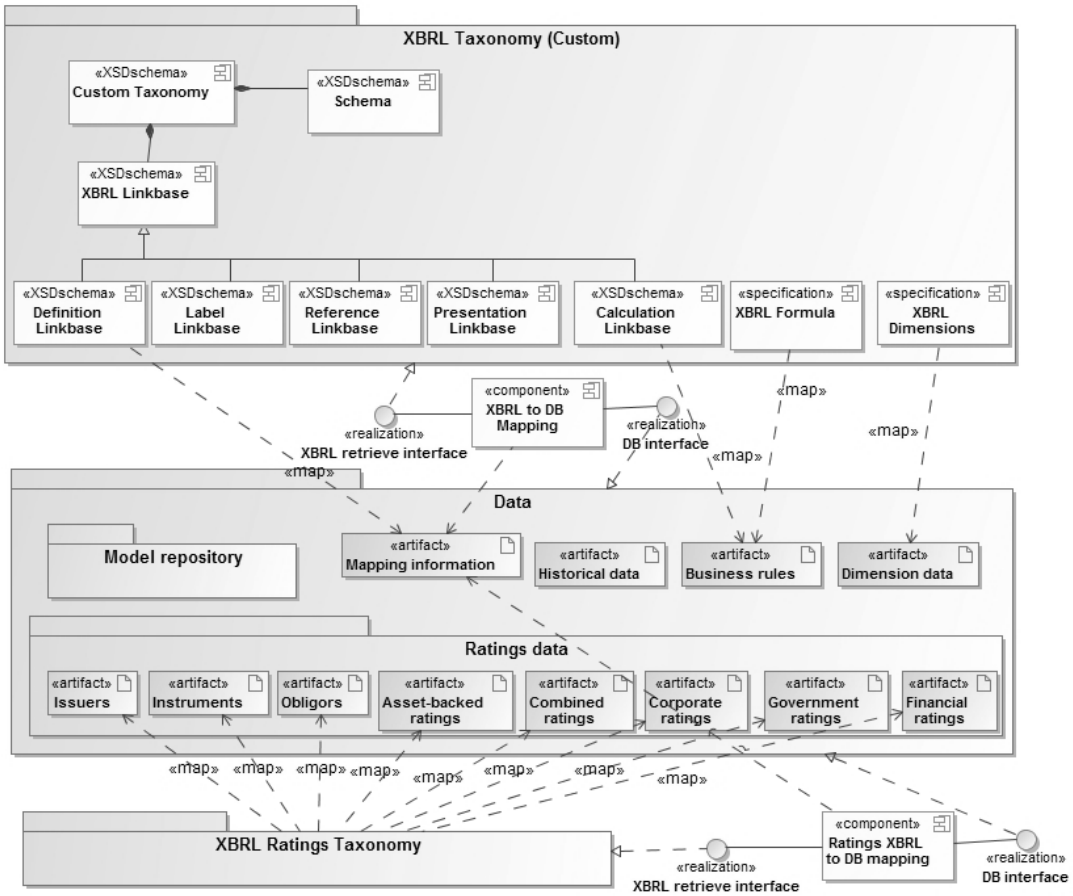
Note that Parameter selection task is linked to both GUI and Machine learning layers; such link represents both manual and automatic (using heuristic or metaheuristic techniques) parameter selection.

5.2. Integration of XBRL financial standard

The process of using various statistical, econometric, machine learning or artificial intelligence techniques uses large datasets of financial data, such as primary and secondary financial ratios, management information, historical data and other entities. Data used for credit risk evaluation model development is usually presented as a two-dimensional *m x n* array of *n* instances of data having *m* attributes. However, formation of such dataset from XBRL is nontrivial as different taxonomies with varying number of financial concepts have to be considered and linked. If public taxonomies are used for this task, only the mappings between these taxonomies and fields in data store model (usually relational model) need to be provided. Total or secondary financial ratios defined as business rules (e.g., *Sales Income = Other Income + Gross Income + Cost of Goods Sold*) in linkbases also need to be

considered; it is proposed to import them into database as business rules by using the same mappings and use them to obtain new features by these rules. They can also provide validation facilities or ensure integrity of existing data. Thus, a proper mapping model for importing should be designed in order to use as much of available information in research as possible.

The basic principles of this mapping model are presented in Figure 23. SEC taxonomies are selected as basis, as they are well developed, applied and consistent with the data used in research. The bindings between taxonomy parts and data model artefacts (they may represent relational tables, etc.) are modelled using <<*map*>> stereotypes.



Source: adopted from Garsva G., Danenas P. XBRL Integration Into Intelligent System For Credit Risk Evaluation.

Figure 23. Mapping model for XBRL data import

The mapping and importing process is done by *XBRL to DB Mapping* component which performs selection of corresponding mapping rules from *Mapping information* and binding them to particular taxonomies and instances. It may also perform the function of dataset formation (if data is stored in relational two-

dimensional model, as it was described before). Note, that model in Figure 23, conversely from original source given in [89], is complemented with XBRL Formula specification which also enables representation of financial business rules.

The mapping operator in this model is represented using ⇔ notation. Therefore, such mappings are given in this model:

- Mappings for reporting periods, financial attributes, such as primary and secondary ratios *Schema* ⇔ *Financial Data*;
- *Schema* ⇔ *Company Data* – mappings for company data (entity name, parent company, various information, etc.);
- *Definition Linkbase* ⇔ *Mapping Information* – metadata representing internal mappings information
- A set of data mappings to enable data dictionary functionality <*Label Linkbase* ⇔ *Multilanguage data, Reference Linkbase* ⇔ *Reference data*>, which represent multilanguage label data and reference data (standards and definitions, links to standards) for every mapped concept;
- A set of mappings to enable business rules functionality <*Calculation Linkbase* ⇔ *Business Rules, XBRL Formula* ⇔ *Business Rules*>;
- *Dimensions* ⇔ *Dimension Data* – dimension data for every instance.

This mapping model is also extended with ratings information; it is defined here as a set of mappings *XBRL Ratings Taxonomy* ⇔ *Ratings Data.* This extension enables mapping of existing XBRL ratings data to financial data, with means to improve existing models or derive new using statistical and machine learning classification techniques. This process is quite straightforward if instances from single taxonomies are used in the modeling process as the number of financial attributes is fixed. This is true for both XBRL open (if a fixed subset of financial ratios is used in modeling) and closed taxonomies

However, in case of open XBRL taxonomies (such as US-SEC), formation of dataset incorporating full XBRL information, including custom extensions by various authorities and companies, is not so straightforward. Given a set of taxonomies $T = \{T_i \mid 0 \le i \leqslant |T|\}$, where $|T|$ is the number of taxonomies in *T*, with $\{F_{T_i}^k \mid N^{\min} \le k \le N^{\max}\}$ concepts, common in every taxonomy (it is obvious that only financial ratios defined in all used taxonomies can be used to form dataset for machine learning task), where the number of common financial ratios, represented by

mappings in taxonomy mappings data store, is defined as minimal number of intercepting concepts F

$$N^{\min} = \min(k) \mid F_{T_i}^k \cap F_{T_j}^k, T_i \in T, 0 \le i \le |T|, 0 \le j \le |T| \tag{5.1}$$

and maximum number of common financial concepts in taxonomy

$$N^{\max} = \max(k) \mid F_{T_i}^k, T_i \in T, 0 \le i \le |T| \tag{5.2}$$

1.

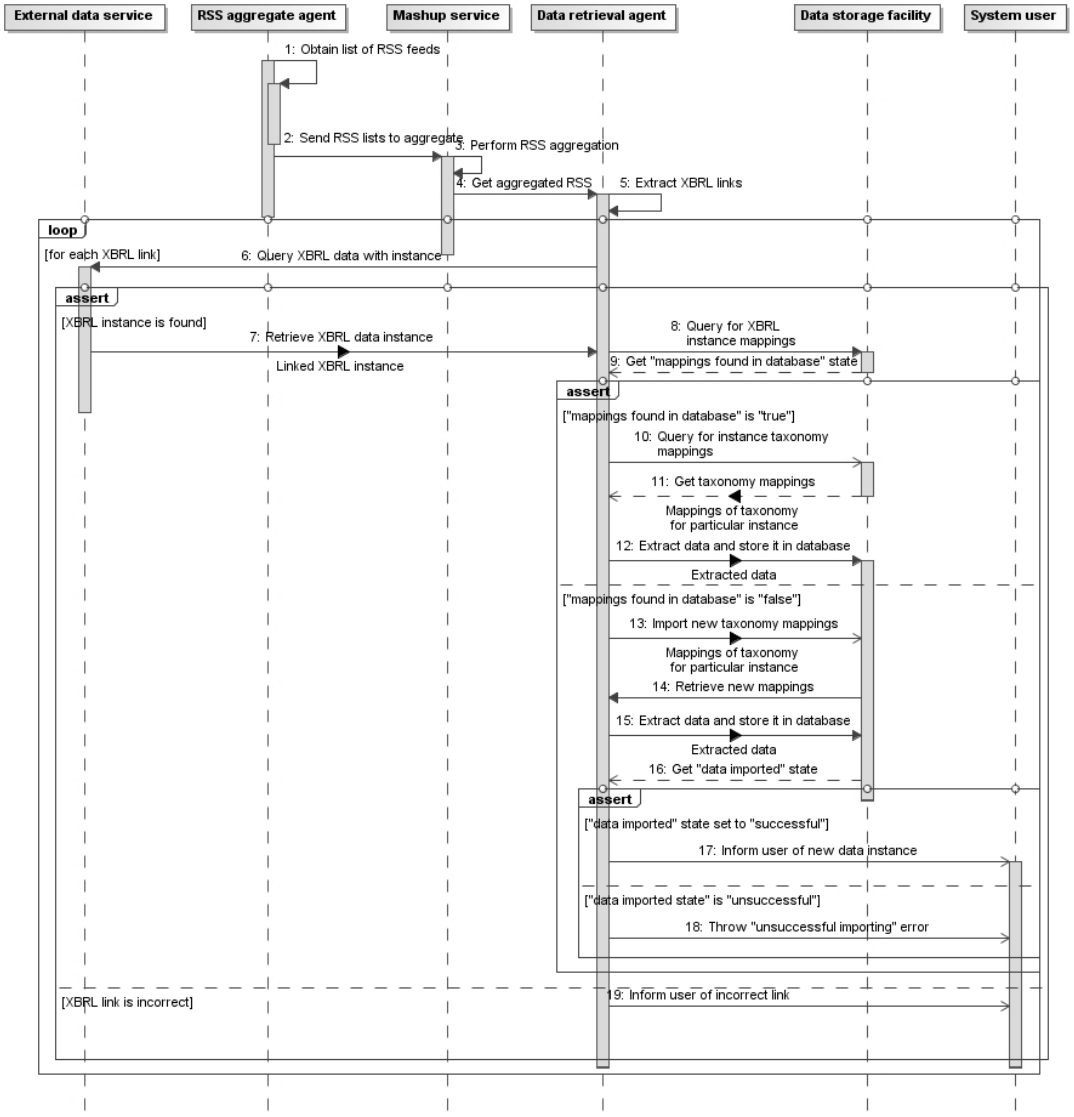


Figure 24. Possible scenario of XBRL importing process

Suppose that dataset represented by m instances from taxonomy set $T = \bigcup_{i=1}^{|M|} T_i$ needs to be defined for classification task. Three possible strategies can be identified:

1. Dataset represented as $m \times N^{min}$ matrix is formed, consisting of financial ratios, common for each taxonomy. Such strategy results in maximum loss of

possible information available for model development.

2. Dataset represented as matrix $m \times k, N^{\min} < k < N^{\max}$ – together with common financial ratios additional ratios is partially integrated. The higher k is selected the more sparse dataset is obtained.

3. Dataset represented as matrix $m \times N^{max}$ is formed, which incorporates maximum amount of available data but results in dataset with largest sparseness.

Two ways of retrieving data can be identified; one of them can be defined as simple data retrieval from remote database with XBRL instances via particular interface. Another scenario is using RSS feeds via mash-up services – composition services, which are used to aggregate RSS links by combining RSS feeds via filters (e.g., date filters, etc.) or mappings, Yahoo Pipes[18] can be named as one of examples for such service. It is assumed that RSS feeds contain links to XBRL documents (instances) with data. RSS Standard is also proposed by SEC as a means of interactive data exchange in *"Ratings Files Publication Guide"* (SEC, 2009). The developers of Arielle XBRL framework also adopt RSS protocol to obtain data from SEC [14], therefore this approach is proved as a reasonable choice. This scenario is modelled as UML 2.0 sequence diagram, using *loop* (iteration) and *assert* (necessity) concepts, and is given in Figure 24. The sequence of activities necessary for first option is also reflected in this diagram, from activity No. 8 "Query for XBRL instance mappings" to No. 18 "Throw unsuccessful import error".

5.3. Implementation scenario

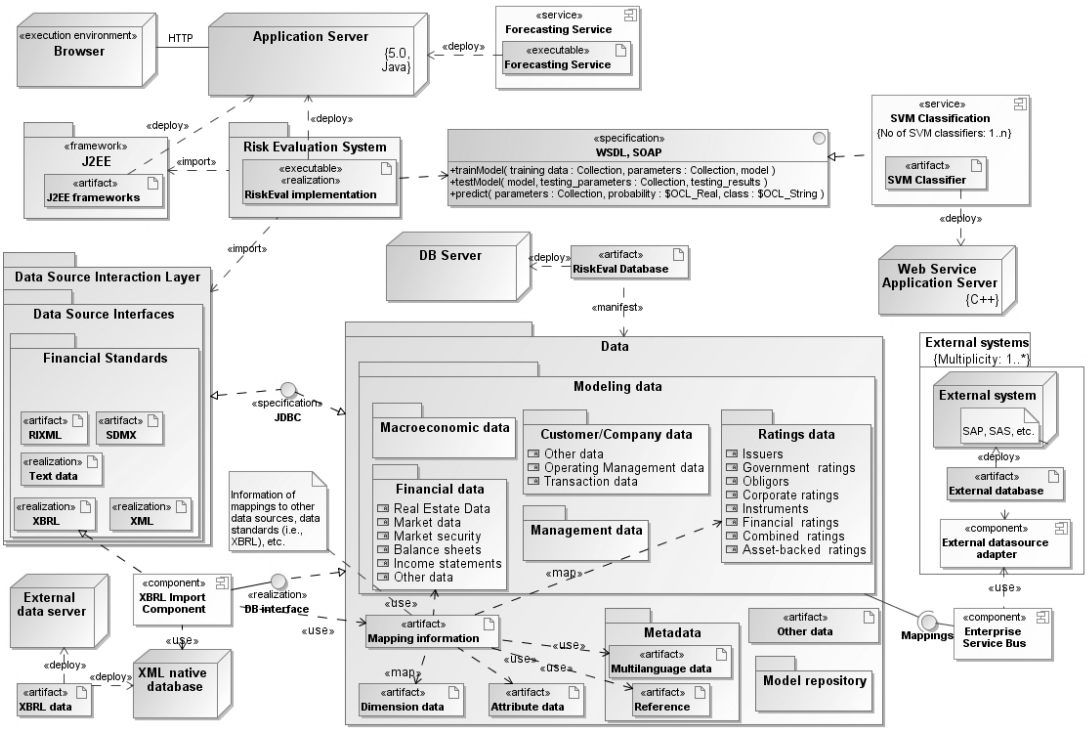
UML implementation diagram for DSS based on proposed framework is given in Figure 25. It describes all server nodes, execution environments as well as possible technologies that can be used to implement the described system. As Table 4 shows, SVM frameworks are implemented in different languages. This makes a difficult task to combine them in hybrid algorithms or use together in single system.

Thus a framework for interoperability, such as CORBA, COM or Web Services, has to be implemented in this system to ensure that as many SVM implementations might be used as possible; as it was mentioned in related work, different SVM algorithms often show different results. Web Services was selected to implement the SVM classifiers as services in this case, as it ensures maximum compatibility and is easier to implement than CORBA because known CORBA open

[18] Yahoo Pipes service, available at http://pipes.yahoo.com/pipes/

source implementations seem difficult to apply (i.e., they do not contain good reference material covering all aspects or are difficult to implement in cross-platform manner), and COM is not cross-platform. Three operations most commonly used in classification tasks are defined in WSDL document: *Training*, *Testing* and *Prediction*. Yet it can be extended with other operations, as some SVM implementations also offer additional functionality, e.g., ranking (SVM^{Light}) or outlier detection using one-class SVM.

JAVA was chosen as an implementation language for the whole system as it offers many possibilities and frameworks needed to implement functionality described here; e.g., WEKA and RapidMiner can be a good choice to implement Machine Learning layer functionality as they contain implementations of mostly referenced SVM algorithms together with many others. There are several known open source cross-platform JAVA XBRL implementation frameworks discussed in Section 2.6.3 (xBreeze Open Source Edition provided by UBMatrix, Batavia XBRL Java Library, XBRLAPI.org) which would allow to develop XBRL Import Component.



Source: P. Danenas, G. Garsva. SVM and XBRL based decision support system for credit risk evaluation (modified)

Figure 25. Implementation diagram for system based on designed framework

A similar component might also be implemented for other similar XML-based formats, e.g., MDDL and SDMX standards might be utilized for macroeconomic data

import, RIXML - for research related data import (depending on the subset defined in this structure); this can also extend the maount of variables in financial analysis. JAVA also offers good possibilities for enterprise-level development and integration with other systems (e.g., using ESB and JMS for messaging-oriented integration development) as well as web interface implementation and development of Web services. Thus, if SVM classifier cannot be implemented in JAVA then C++ can be a good option as Table 4 shows that most of them are written in C/C++ or MATLAB (using its *mex* compiler). These aspects are defined in proposed system framework. Such features of implementation proposed in Figure 25 can be emphasized:

- Cross-platform – XBRL and other XML-based standards do not depend on any system platform; JAVA can be deployed on any Windows/*nix/BSD platform and mature cross-platform frameworks for development in C++ are also known (e.g., Qt framework by Digia[19]);
- Data source-independent – application of object persistence technology and SQL query mapping to corresponding entities in object-oriented manner allows implementing the system almost independently on DBMS or other data sources.

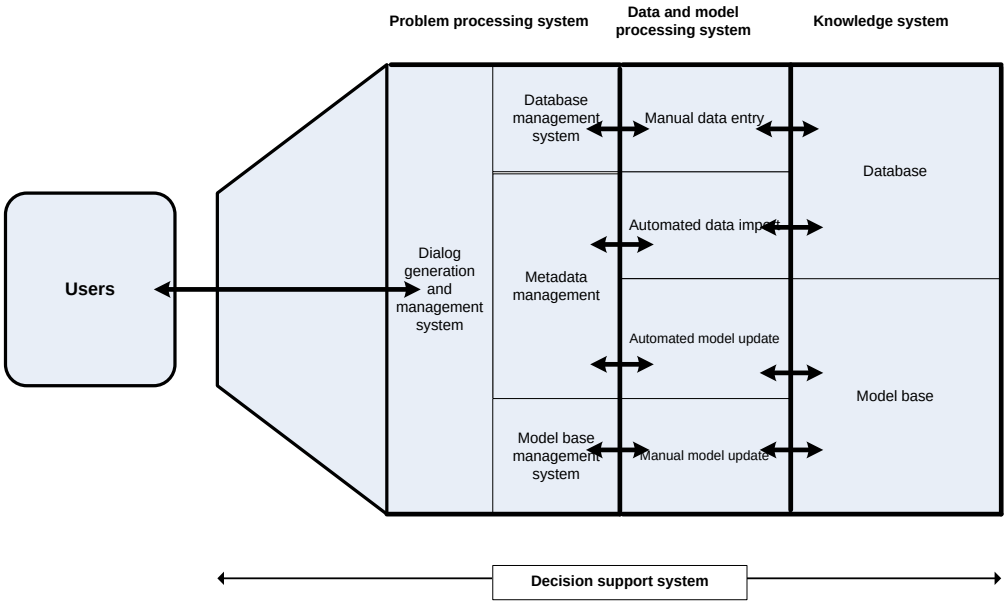


Figure 26. Extension of Holsapple's combined database and solver-driven DSS architecture using data and model processing layers

Therefore, such automation enables important extensions of generic combined database and solver oriented DSS architecture described by Holsapple (Figure 31),

[19] Qt Framework homepage, http://qt.digia.com/

such as management of metadata (taxonomy data, mappings) and automated processing. The modified architecture is presented in Figure 26. This architecture includes new structural components such as automated data capturing and importing, automated updating of existing models using new data instances; in proposed framework extension they can be designed as independent set of agents.

5.4. Design and development methodology for developed DSS

Design and development methodologies are an important issue in both software engineering and DSS domain as proper selection and application of such methodology might reduce time needed to design and develop complex modular systems. Component-based software engineering (abbr. as CBSE) and Model Driven Architecture (abbr. as MDA), based on extensive UML application, are one of most popular methodologies for object-oriented software design and engineering. They are widely analysed and discussed in various software-engineering related papers and books by different authors, for e.g., Nash [169]. However, analysis of similar methodologies, previously applied for engineering of various DSS, in [57] showed that there is not much research done in this field – some of surveyed techniques are targeted at generic DSS development, although there were approaches based on MDA, Unified Process. However, for a sophisticated DSS described in this work (which might be considered both as data-driven and solver, also including expert knowledge expressed in rules), approaches for more complex systems should be considered. Domain-driven design and development (abbr. as DDD), an approach extending model driven development (abbr. as MDD), has been introduced by Evans in [79] and at the time of writing is widely adopted for enterprise software development. This approach can be considered as a possible solution for developed system as the multilayered design of proposed framework simplifies its integration by describing mappings between layers of proposed framework and corresponding layers in responsibility layer structure of DDD; other core concepts are domain, model, ubiquitous language and context. Strategic Domain Driven Design provides a set of principles to ensure model integrity, refine domain model and work with several models. The definitions of these concepts, DDD patterns and best practices are described in [57, 79].

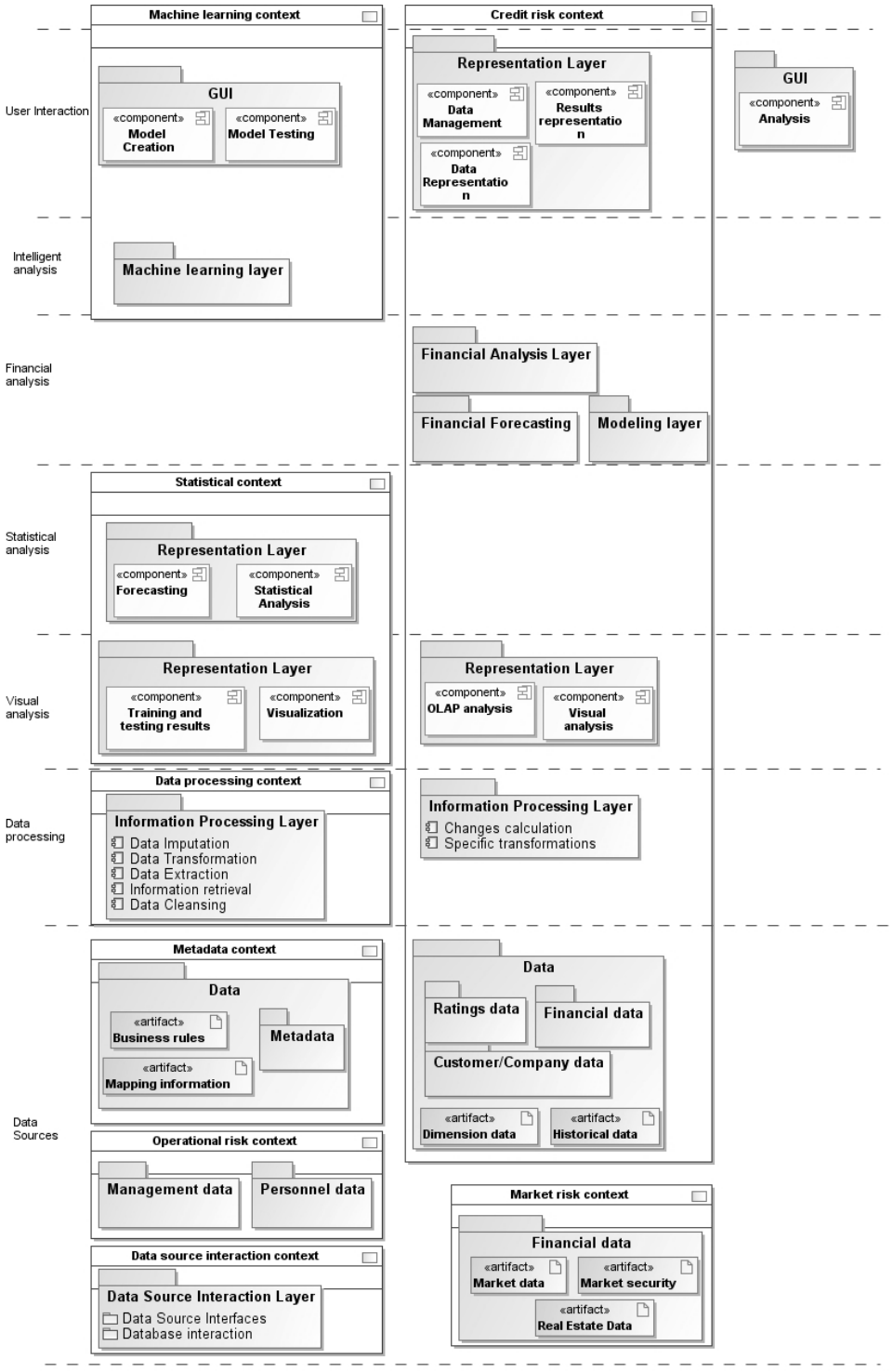
According to [57], such core DDD patterns are proposed for implementation of proposed system:

- highlighted core to simplify core domain description and highlight its main aspects;
- segregated and abstract core models to refactor core domain in order to obtain clearer models;
- responsibility layers can be used to refactor the model in a multilayered way such that the responsibilities of each domain object, aggregates and modules fall into separate layers. This is an important aspect in multidomain, multicontext and multiaspect DSS development, such as the one proposed in Section 5.1;
- knowledge level pattern can be used as extension to responsibility layers to implement such layer system in hierarchical manner, where each level directly depends on lower level;
- pluggable component framework as a pattern for component substitution. It also allows separating and encapsulating several bounded contexts by exposing their functionality as components, with shared kernel as core [57, 79].

Therefore, as Evans highlights [79; cited by 57], context, distillation, and large-scale structure design principles are complementary and interact in many ways, for e.g., a large-scale structure can exist within one bounded context, or it can cut across many of them and organize the context map. Thus, as proposed in [57], 7 contexts were refined in the proposed framework – statistical context, machine learning context, credit risk context, operational risk context, market risk context, data source interaction context, metadata context.

Each of these contexts represents different aspects of the system. This should not be confused with modules, as bounded contexts provide the logical frame inside of which the model evolves, and modules are more of a tool for organization of models' elements. To extend the framework by introducing additional layers with meaningful semantics, it was extended with 7 responsibility layers from DDD, as it is shown in Figure 27. A simplified architecture with reduced complexity for developed DSS framework is also proposed in [57], which can be chosen as more suitable option for practical development. Another development technology, which can be considered as an option for development process, is Feature Driven Development (abbr. as FDD). This is one of agile techniques compatible with an iterative and incrementing model-driven software development process, using UML diagrams in each of its steps. It is based on short iterations and five activities: overall model

development, feature list building, plan by feature, design by feature and build by feature; other three activities are iterative, where special feature is developed [172].



Source: P. Danenas, G.Garsva. Domain Driven Development and Feature Driven Development for Development of Decision Support Systems.

Figure 27. Responsibility layers for credit risk DSS

According to Palmer and Felsing, feature is described as small function valued

by client, expressed in <*action*><*result*><*object*> form (e.g., calculate rating of customer). Domain analysis for each feature is done by domain expert. Palmer and Felsing [172] also describe a generic FDD architecture consisting of 4 layers; its connection to simplified architecture of DDD is demonstrated in [57]. FDD can also be considered as as a good choice for development of complex systems, as each feature is developed iteratively and often – independently from others; it is perfectly suitable for features which comprise machine learning and statistics based techniques, whether it is a class, component or a module [57]. Danenas and Garsva also give FDD technical architecture for proposed DSS framework in their work [57]. They also discuss particular layers and their structure, as well as communication with external systems and Web Services. These authors also note that, although separate features would be developed by separate development teams, integration of such features requires higher levels experts.

5.5. Description of developed prototype

This section describes the implementation of the system developed using framework described in Section 5.1 and design methodology in Section 5.4. The implementation model described in Section 5.3 was used to guideline the technological design and development of the system. The developed system uses EDGAR database based on SEC model presented in Section 5.2.

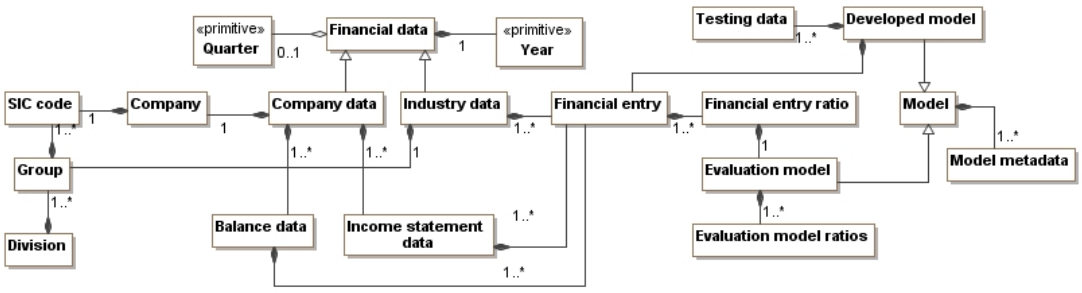


Figure 28. Prototype object model

At the time of writing it consisted of quarterly and yearly financial data of 9365 companies located in USA in period of 1999-2008; the subsets of this data were also used in the research. Appendix F gives a detailed description of currently used data while 0 shows database structure of the developed prototype. System objects are described using UML object diagram (Figure 28). PostgreSQL database management system was used as data storage facility, which offers deployment abilities in heterogeneous platorms, high-performance and extensibility using stored procedures

implemented in C, Tcl or Python. Most of data processing and model developed functionality was developed using Weka framework. The system is implemented using scenario described in Figure 25. Such features are currently implemented:

- Capabilities of dataset formation from database (based on object model defined in Figure 28), saving into and loading from ARFF, CSV and binary BSI files;

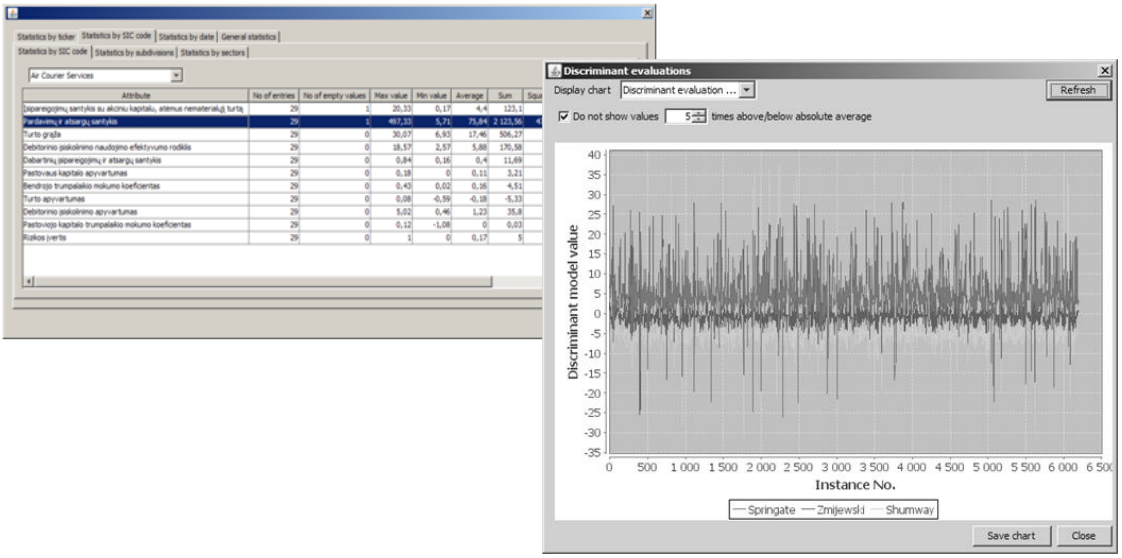


Figure 29. GUI examples of developed DSS prototype

- Enter new, edit or delete objects shown in Figure 28 such as company data, financial data, sector data, model data;
- Calculation of various statistics (number of missing values, mean, maximum, minimum values, sums, averages, standard deviations) in different dimensions (company, SIC code, division, general dataset, etc.);
- Editing datasets – removing unlabeled instances;;
- Data preprocessing features – data scaling (normalization/standardization), imputation (standard. EM, average of company data), transformation (principal components, differences, etc.);
- Statistics and visualization of evaluations by different models;
- Groups and metadata of financial ratios;
- Classification using SVM.

Figure 29 illustrates user interface for developed prototype. More examples of GUI for this DSS are given in figures in Appendix K.

5.6. Conclusions

1. Main components of proposed framework for intelligent DSS based on SVM and XBRL development are identified and described. The proposed framework is expressed as UML package diagram consists of 5 layers which represent most important aspects: data, business logic (financial analysis, modelling and forecasting) and representation, as well as data source interaction which describes mapping information for XBRL based taxonomies. Such structure is also useful for further development as it separates various aspects (components) which can be extended or adopted for other similar problems (whole components or parts of them). A possible implementation scenario as UML implementation diagram is also presented.

2. Mapping model between XBRL taxonomies and data storage facilities is developed as a tool for automated data import from XBRL instances, together with a scenario of its application. Its implementation details are also discussed.

3. DSS implementation using Java technologies with PostgreSQL data storage facility is described, together with its currently implemented features. Currently it offers abilities to load, analyse, preprocess and transform data as well as perform classification using SVM techniques. Guidelines for further development are also provided.

THESIS RESULTS AND CONCLUSIONS

1. The overview of previous research in computational intelligence in credit risk domain has shown that:

a. SVM oriented research at the moment of writing tends to be dominating among research of various statistical and computational intelligence techniques in financial and credit risk domain and usually shows comparable or better results than similar statistical or computational intelligence techniques, thus their research is promising.

b. The results presented by different authors who applied computational intelligence techniques in credit risk domain are difficult to compare, as the results highly depend on data used in the experiment, the approach used in the experiment, implementation of techniques, the resources that were available during the experiment, experiment configuration.

c. Hybrid methods based on SVM tend to show comparable or better performance than similar statistical, econometrical or standalone SVM methods.

d. Support Vector Machines based technique has several advantages over similar techniques, such as comparably simple architecture, ability to avoid overtraining and overfitting. It also has numerous algorithms and implementations. Its main disadvantages are complicated choice of optimal parameters, lack of mature framework combining all or most of these techniques which slows down its research and adoption.

e. There is a large variety of financial attributes (20 groups of such ratios are described in Appendix H of the thesis) which can be used in feature vector formation; however, all papers use different subsets of such ratios. Therefore, feature selection step can be applied to select most significant features.

2. The overview of previous research in credit risk domain and decision support systems has shown that:

a. Credit risk evaluation process is a multidimensional process, as credit risk analysis can be applied in several dimensions: financial sector (group, industrial code), financial data period (quarter, year), type of obligor (individual, company, government), type of rating, currency, globality (national, international), period (short, medium, long). Therefore, a multidimensional model for result analysis, consistent with the technique in this work, has been developed as a tool for such

evaluation.

b. Multiple discriminant analysis and logistic regression techniques are mostly known and applied in real world statistical techniques, with Altman, Springate, Zmijewski, Ohlson models referred as the most popular.

c. Several authors give their classifications of decision support systems; however, none of them describes a DSS with automated data import functionality or integration with metadata that can be provided by some modern standards. Therefore, model of combined database and solver oriented DSS architecture described by Holsapple is extended with automated data integration and model update functionality layer, forming similar to agent-based DSS architecture, and is selected as a choice for architecture of intelligent DSS proposed in this work.

d. Overview of banking regulation standards (particularly Basel standard) identified the key requirements for such system in terms of security, data stored, supervisory access and storage facility, as well as requirements for ratings themselves.

e. Financial standards such as XBRL, RIXML are becoming an important part of financial regulation as they offer clear, extensible, flexible, adaptable and structured framework for financial reporting, data transfer and representation. XBRL enables definition of validation or derived rules which can be applied in decision support to ensure data integrity or derivation of new variables. These standards can be applied to solve data interchange and data quality problems often faced in banking institutions. None of previous work related to development of similar frameworks or decision support systems for credit risk evaluation described possible interfacing with XBRL, thus a mapping model compatible with classification problem researched in this work is also proposed as part of the proposed framework.

f. The analysis of taxonomies for decision support systems, existing structures for their applications in credit risk domain helped to identify core components of these systems – model repository, inference engine (business logic), knowledge base and data storage facility.

3. A new classifier PSO-LinSVM, based on Particle Swarm Optimization and linear SVM, is proposed with following capabilities:

a. More suitable for large-scale learning than similar nonlinear SVM techniques.

b. Automatic selection of SVM classifier from a family of similar classifiers with the same parameters.

c. Less complex configuration than using other evolutionary techniques, e.g., Genetic Algorithm approach.

d. An option to optimize for either accuracy or TP ratio performance which makes it usable with both balanced and unbalanced data.

4. Experimental research of PSO-LinSVM identified that:

a. According to experiments performed on German and Australian credit datasets, PSO-LinSVM is capable to show better performance compared to similar optimized linear and nonlinear SVM classifiers.

b. It also resulted in best quality of data separation in terms of the sum of sensitivity and specificity.

c. Experimental results showed that it covered a large space of possible solutions and resulted in larger variety of obtained classifiers, compared to similarly developed GA-LinSVM technique, where single classifier dominated.

5. Two approaches for development of classifier for credit risk evaluation using external evaluations – FS-SVM^{DA} and FS-SVM^{SWTest} - are presented and researched in this work. They combine feature selection, classification; FS-SVM^{SWTest} also uses sliding window testing. They have following properties:

a. Both of these approaches were tested on datasets of various sizes which make them suitable for both small and large scale learning.

b. Feature selection step automatically identifies significant ratios.

c. In case of the second technique, the testing is done for next several periods; this helps to ensure that trained classifier is consistent not only with the following, but also but much larger period.

6. Experimental evaluation of FS-SVM^{DA} and FS-SVM^{SWTest} approach identified that:

a. The results varied on different sectors; therefore it highly depends on the dataset that is used in the research.

b. Experimental results of FS-SVM^{DA} using data from all sectors showed that average accuracy was above 80% for linear SVM based classifiers, and over 86% for C-SVC based classifiers (Altman based evaluator was used in experiments). However, it was not efficient in prediction of evaluator changes.

c. SVM-based classifiers SMO, Core Vector Machines, Ball Vector Machines, mySVM are a good alternative for larger scale learning and show performance comparative or better than standard implementations (e.g., LibSVM or SVM^{Light}). This implies that more attention should be given to these techniques in future research.

d. Results of FS-SVM^{SWTest} approach and PSO-LinSVM as base classifier showed that it is capable of performing classification with high accuracy (over 90%), although accuracy varied, depending on the sector and evaluator used.

e. Application of FS-SVM^{SWTest} approach to actual bankruptcy identification resulted in promising results as ir performed better than original evaluator. Although these results should be treated carefully at the moment, they give a premise to develop an approach for classifier selection with respect to original evaluations and identification performance based on proposed technique.

7. A framework for intelligent DSS based on SVM and XBRL development, consistent with proposed techniques, is designed and described using UML diagrams:

a. It consists of 5 layers which represent most important aspects: data, business (domain) logic, models (machine learning) and representation, as well as data source interaction together with mapping information.

b. It enables reuse for other similar problems (whole components or parts of them) as such structure clearly separates various aspects as components.

c. A possible implementation scenario as UML implementation diagram is also presented; it proposes development of cross-platform and data source independent DSS.

d. A combined methodology based on Domain Driven Design and Feature Driven Development is described and proposed for development of DSS based on suggested framework together with architectural design models for this framework.

e. A prototype using this scenario is implemented.

REFERENCES

1. Abner S. L., Fuzzy Logic in Expert Systems, http://www.cs.rockhurst.edu/seminars/CS2002/Sundance. , accessed on 2008.06.31.
2. Abramowicz W., Nowak M., Sztykiel J. Bayesian networks as a decision support tool in credit scoring domain. Managing Data Mining Technologies in Organizations: Techniques and Applications, IGI Publishing Hershey, 2003, pp. 1–20.
3. Ahn H., Lee K., Kim K.-J. Global Optimization of Support Vector Machines Using Genetic Algorithms for Bankruptcy Prediction. ICONIP'06 Proceedings of the 13th international conference on Neural information processing - Volume Part III, 2006, pp. 420-429.
4. Ahn H., Kim K. Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach," Applied Soft Computing, Vol. 9, No. 2, 2009, pp. 599–607.
5. Allen L. Credit Risk Modeling of Middle Markets, http://philadelphiafed.org/consumer-credit-and-payments/payment-cards-center/events/conferences/2002/allenpaper.pdf, 2002, accessed on 2012.08.20
6. Alter S. L. Decision Support Systems: Current Practice and Continuing Challenge. Addison-Wesley, 1980.
7. Altman E. I.. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, Vol. 23(4), 1968, pp. 589–609.
8. Altman E. I., Haldeman R. G., Narayanan P.. ZETA Analysis: A new model to identify bankruptcy risk of corporations. Journal of Banking & Finance, Vol.1(1), 1977, pp. 29-54.
9. Altman E. I. Predicting financial distress of companies: Revisiting the Z-score and Zeta models (2000), http://www.defaultrisk.com/pp_score_14.htm, accessed on 2012.08.08.S. J. Russell, P. Norvig. Artificial Intelligence: A Modern Approach. Prentice-Hall, 1995.
10. Altman E. I., Narayanan P. An International Survey of Business Failure Classification Models. Financial Markets, Institutions and Instruments, Vol. 6, No. 2, 1997, pp. 1–57.
11. American Council for Technology (ACT) , Financial Management Committee, Financial Information Sharing (FIS) Subcommittee. Transforming Financial Information – Use of XBRL in Federal Financial Management, http://xml.gov/documents/completed/iac/XBRLWhitePaper.pdf, accessed on 2012.09.15.
12. American Heritage Dictionary, 5th edition, http://www.ahdictionary.com/, accessed on 2012.09.12.
13. Anderson R. The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation, Oxford University Press Inc., New York, 2007.
14. Arelle, Open Source XBRL platform, http://www.arelle.org, accessed on 2012.09.12.
15. Arnott D., Pervan G. A critical analysis of decision support systems research. Journal of Information Technology, vol. 20, no. 2, 2005, pp. 67–87.
16. Bae J. K., Kim J. Combining models from neural networks and inductive learning algorithms. Expert Systems with Applications, Vol. 38(5), 2011, pp. 4839-4850.
17. Baesens B., Egmont-Petersen M., Castelo R., Vanthienen J. Learning Bayesian network classifiers for credit scoring using Markov Chain Monte Carlo search. ICPR "02 Proceedings of the 16 th International Conference on Pattern Recognition (ICPR"02), Vol. 16, No. 2, 2002, pp. 49–52.
18. Balthazar L. From Basel 1 to Basel 3: The Integration of State-of-the-Art Risk Modeling in Banking Regulation. Palgrave Macmillan, 2006.

19. Barthelemy S., Apoteker Th. Genetic Algorithms and Financial Crises in Emerging Markets. CEFI International Conference Proceedings, 2000. Available at SSRN: http://ssrn.com/abstract=687741

20. Beemer B. A., Gregg D. G. Advisory Systems to Support Decision Making. In: Burstein F., Holsapple C. W. (eds). Handbook on Decision Support Systems 1: Basic Themes. Springer-Verlag Berlin Heidelberg, 2008, pp. 511-527.

21. Begley J., Ming J., Watts S. G. Bankruptcy Classification Errors in the 1980s: An Empirical Analysis of Altman's and Ohlson's Models. Review of Accounting Studies, Vol. 1(4), 1997.

22. Bellovary J., Giacomino D. E., Akers M. D. A Review of Bankruptcy Prediction Studies: 1930 to Present. Journal of Financial Education, Vol. 33, 2007.

23. Bellotti T., Crook J. Support vector machines for credit scoring and discovery of significant features. Expert Systems with Applications, Vol. 36, No. 2, 2009, pp. 3302–3308.

24. Ben-Hur A., Horn D., Siegelmann H. T. Support vector clustering. The Journal of Machine Learning, Vol. 2, 2001, pp. 125–137.

25. Ben-Hur A., Ong C.S., Sonnenburg S., Schölkopf B., Rätsch G. (Support Vector Machines and Kernels for Computational Biology. PLoS Computational Biology Vol. 4(10), 2008.

26. Blum Ch. Ant colony optimization: Introduction and recent trends. Physics of Life Reviews, Vol. 2, 2005, pp. 353–373.

27. Bose I., Pal R. Using support vector machines to evaluate financial fate of dotcoms. Proceedings on Pacific Asia Conference on Information Systems (PACIS) 2005, 2005, p. 42.

28. Brabazon A., O'Neill M. Biologically inspired algorithms for financial modelling. Springer-Verlag New York Inc., 2006.

29. BSVM, http://www.csie.ntu.edu.tw/~cjlin/bsvm/index.html, accessed on 2012.07.31

30. Buzius G., Danenas P., Garsva G. Credit risk evaluation using SVM and Bayesian classifiers. Proceedings of the 15th Conference for Master and PhD students "Information Society and University studies", Kaunas, Lithuania, 2010, pp. 27-32.

31. Cambridge Advanced Learner's Dictionary, 2nd ed., access online http://dictionary.cambridge.org/dictionary/british/, accessed on 2012.09.16.

32. Cantú-paz E. A survey of parallel genetic algorithms. Calculateurs Paralleles, Reseaux et Systems Repartis, Vol. 10(2), 1998, pp.141-171.

33. Canu S., Grandvalet Y., Guigue V., Rakotomamonjy A. SVM and Kernel Methods Matlab Toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005, http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html, accessed on 2012.07.31.

34. Caouette J., Altman E., Narayanan P. Managing credit risk: the next great financial challenge. Wiley & Sons Inc., 1998.

35. CENATIC (Spanish National Reference Centre for the Application of Information and Communication Technologies) (2008), "Open Source XBRL Tools Study, Final Report", http://observatorio.cenatic.es/index.php?option=com_rubberdoc&view=doc&id=50&format=raw, accessed on 2012.09.15.

36. Chang C.-C., Lin C.-J. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, Vol. 2(3), 2011, Article 27.

37. Chang, E. Y., Zhu, K., Wang, H., Bai, H., Li, J., Qiu, Z., Cui, H. PSVM: Parallelizing Support Vector Machines on Distributed Computers. Advances in Neural Information Processing Systems, Vol. 20, 2007.

38. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. (2000). CRISP-DM 1.0. Step-by-step data mining guide, http://www.kde.cs.uni-kassel.de/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf, accessed on 2012.07.31

39. Chaudhuri A., De K. Fuzzy Support Vector Machine for bankruptcy prediction. Applied Soft Computing, Vol. 11, No. 2, 2011, pp. 2472–2486.

40. Chauhan N., Ravi V., Karthik Chandra D. Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. Expert Systems with Applications, Vol. 36(4), 2009, pp. 7659-7665.

41. Chawla N.V., Bowyer K. W., Hall L. O., Kegelmeyer W. Ph. 2002. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, Vol. 16(1), 2002, pp. 321-357.

42. Chen C., Chen M., Hsieh C.-H. A Financial Distress Prediction System Construction based on Particles Swarm Optimization and Support Vector Machines." Proceedings of 2010 International Conference on E-business, Management and Economics, Vol. 3, 2010, pp. 165–169.

43. Chen H.-J., Huang S. Y., Lin C.-S. Alternative diagnosis of corporate bankruptcy: A neuro fuzzy approach, Expert Systems with Applications Vol. 36(4), 2009, pp. 7710-7720.

44. Chen M.-Y. Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches. Computers & Mathematics with Applications, Vol. 62, No. 12, 2011, pp. 4514–4524.

45. Chen S., Härdle W., Moro R. A. Estimation of Default Probabilities with Support Vector Machines. SFB 649 Discussion Paper 2006-077.

46. Chen W.-D., Li J.-M. A model based on factor analysis and Support Vector Machine for Credit Risk Identification in small-and-medium enterprises. Procedings of 2009 International Conference on Machine Learning and Cybernetics, 2009, pp. 913–918.

47. Chen W.-H., Shih J.-Y. A study of Taiwan's issuer credit rating systems using support vector machines, Expert Systems with Applications, Vol. 30, Issue 3, 2006, pp. 427–435.

48. Cheng, H., Lu, Y.-C., Sheu, C. An ontology-based business intelligence application in a financial knowledge management system. Expert Systems with Applications, vol. 36, no. 2, 2009, pp. 3614-3622.

49. Chong W., Yingjian G., Dong W. Study on Capital Risk Assessment Model of Real Estate Enterprises Based on Support Vector Machines and Fuzzy Integral. Proceedings of Control and Decision Conference, 2008, pp. 2317-2320.

50. Cybenko G. Continuous valued neural networks with two hidden layers are sufficient (Technical Report). Department of Computer Science, Tufts University, Medford, MA, 1988.

51. Core Vector Machine, http://www.cse.ust.hk/~ivor/cvm.html, accessed on 2012.07.31

52. Cortes C., Vapnik V. Support-vector networks. Machine learning, Vol. 20, No. 3, 1995, pp. 273–297.

53. Crammer K., Singer Y. On the learnability and design of output codes for multiclass problems. Proc. of the 13th Annual Conference on Computational Learning Theory, Vol. 28, 2000, pp 35–46.

54. Cristianini N., Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, 2000.

55. D&B – Business Information – Get Credit Reports, http://www.dnb.com/, accessed on 2012.08.06.

56. Danenas P., Garsva G. Credit risk evaluation using SVM-based classifier. Lecture Notes in Business Information Processing, Berlin, Springer, Vol. 57, Part 1, 2010, pp. 7-12.

57. Danenas P., Garsva G. Domain Driven Development and Feature Driven Development for Development of Decision Support Systems. Information and Software Technologies: Proceedings of 18th International Conference (ICIST 2012), Communications in Computer and Information Science, Vol. 319, Part 4, 2012, pp. 187-198.

58. Danenas P., Garsva G., Gudas S. Credit Risk Evaluation Model Development Using Support Vector Based Classifiers. Procedia Computer Science, Vol. 4, Elsevier, 2011, pp. 1699-1707.

59. Danenas P., Garsva G. Credit Risk Modeling of USA Manufacturing Companies Using Linear SVM and Sliding Window Testing Approach. Lecture Notes in Business Information Processing, Vol. 117, Part 8, 2012, pp. 249-259.

60. Danenas P., Garsva G. Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach. Procedia Computer Science, Vol. 9, 2012, pp. 1324 – 1333.

61. Danenas P., Garsva G. Simutis R. Development of Discriminant Analysis and Majority-Voting Based Credit Risk Assessment Classifier. Proceedings of the 2011 International Conference on Artificial Intelligence (ICAI 2011), CSREA Press, Vol.1, 2011, pp. 204-209.

62. Danenas P., Garsva G. PSO-based Linear SVM Classifier Selection for Credit Risk Evaluation Modeling Process. Proceedings of 14th International Conference on Enterprise Information Systems (ICEIS 2012), Vol. 1, 2012, SciTePress.

63. Danenas P., Garsva G. Support Vector Machines and their Application In Credit Risk Evaluation Process. Transformations in Business & Economics, Vol. 8, No. 3 (18), 2009, pp. 46-58.

64. Danenas P., Garsva G. SVM and XBRL based decision support system for credit risk evaluation. Proceedings of the 17th International Conference on Information and Software Technologies (IT 2011), Technologija, Kaunas, Lithuania, 2011, pp. 190-198.

65. Danenas P., Merkevicius E., Garsva G. Sistemos modulio, skirto intelektualių modelių kredito rizikos vertinimui kūrimui, koncepcinė struktūra. Informacinės technologijos, 2008, pp.62-68.

66. De Brabanter K., Karsmakers P., Ojeda F., Alzate C., De Brabanter J., Pelckmans K., De Moor B., Vandewalle J., Suykens J.A.K. LS-SVMlab Toolbox User's Guide version 1.8, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2010, http://www.esat.kuleuven.be/sista/lssvmlab/downloads/tutorialv1_8.pdf, accessed on 2012.07.31

67. Deakin E. A discriminant analysis of predictors of business failure. Journal of Accounting Research, Vol. 10(1), 1972, pp. 167-179.

68. Debnath R., Takahide N., Takahashi H. A decision based one-against-one method for multi-class support vector machine. Pattern Analysis and Applications, Vol. 7(2), 2004, pp. 164-175.

69. Deboeck G. Financial Applications of Self-Organizing Maps. Neural Network World, Vol. 8(2), pp.213 -241.

70. Debreceny R., Felden C., Ochocki B., Piechocki M., Piechocki M.: XBRL for interactive data: Engineering the Information Value Chain. Springer, 2009.

71. Dong Y. Reasoning System for Customer Credit Scoring: Comparative Study of Similarity Measure. Proceedings of The 51st Annual Meeting of the International Society for the Systems Sciences, CD-ROM, Tokyo, Japan, 2007.

72. Dorigo M., Stützle T. An Experimental Study of the Simple Ant Colony Optimization Algorithm. Proceedings of the WSES International Conference on Evolutionary Computation, 2001, pp. 253–258.

73. Dunham M. H. Data Mining: Introductory and Advanced Topics. Prentice-Hall, 2003.

74. Eberhart R.C., Shi Y. Particle Swarm Optimization: Developments, Applications and Resources. In Proceedings of the IEEE Congress on Evolutionary Computation, Vol. 1, 2001, pp. 27–30.

75. Edelkamp S., Schroedl S. Heuristic Search Theory and Applications. Morgan Kaufmann, 2011.

76. Elizalde A. Credit risk models II: structural models. Working Papers wp2006_0606, CEMFI, 2006.

77. Elizalde A. Credit Risk Models III: Reconciliation Reduced-Structural Models. Working Papers wp2006_0607, CEMFI, 2006.

78. Engelbrecht A. Computational intelligence: an introduction, 2nd Ed., Wiley & Sons Inc., 2007.

79. Evans, E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison Wesley, 2003.

80. Fan R., Chang K., Hsieh C., Wang X., Lin C. LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research, Vol. 9. 2008, pp.1871–1874.

81. Fan A., Palaniswami M. Selecting Bankruptcy Predictors Using a Support Vector Machine Approach. IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00), Vol. 6, 2000, pp.6354.

82. Fischer, H., Mueller, D. Enabling Comparability and Data Mining with the Arelle® Open Source Unified Model. First Conference on Financial Reporting in the 21st Century: Standards, Technology, and Tools in Macerata, Italy, 2011-09-09, http://arelle.org/wordpress/wp-content/uploads/downloads/2011/09/ComparabilityAndDataMiningUnifiedModel-Paper.pdf, accessed on 2013.01.15.

83. Fitch Ratings homepage, http://www.fitchratings.com/web/en/dynamic/fitch-home.jsp, accessed on 2012.08.06.

84. FOLDOC - Free On-line Dictionary of Computing, http://foldoc.org/, accessed on 2012.08.06.

85. Fung G., Mangasarian O. L. A Feature Selection Newton Method for Support Vector Machine Classification. Computational Optimization and Applications. Vol. 28(2), 2004, pp. 185 – 202.

86. Fung G., Mangasarian O. L. Proximal Support Vector Machine Classifiers. Proceedings of KDD-2001: Knowledge Discovery and Data Mining, San Francisco, CA, 2001, pp. 77-86.

87. Galkus E., Danenas P., Garsva G. Application of ensemble classification methods in credit risk evaluation (in Lithuanian). Conference Proceedings of "Information Technology 2012", 17th Conference for Master and PhD students, Kaunas, Lithuania, 2012, pp. 70-73

88. Gao Zh., Cui M., Po L.-M. Enterprise Bankruptcy Prediction Using Noisy-Tolerant Support Vector Machine. Proceedings of 2008 International Seminar on Future Information Technology and Management Engineering, 2008, pp.153-156.

89. Garsva G., Danenas P. XBRL Integration Into Intelligent System For Credit Risk Evaluation. Transformations in Business & Economics, Vol. 10, No. 2 (23), 2011, pp. 88-103.

90. Ghodselahi A. A Hybrid Support Vector Machine Ensemble Model for Credit Scoring. International Journal of Computer Applications, Vol. 17, No. 5, 2011, pp. 1–5.

91. Gluchowski P., Pastwa A. Process and Technical Design of an Integrated Solution for (Semi-) Automated Basel II-Reporting Using XBRL and Web Services. New Dimensions of Business Reporting and XBRL, Springer, 2007, pp. 211-233.

92. Grice J. S., Dugan M. T. The Limitations of Bankruptcy Prediction Models: Some Cautions for the Researcher. Review of Quantitative Finance and Accounting Vol. 17(2), 2001, pp. 151-166.

93. Grice J. S., Ingram R. W. Tests of the generalizability of Altman's bankruptcy prediction model. Journal of Business Research Vol. 54, 2001, pp. 53-61.

94. Guermeur Y. Homepage, http://www.loria.fr/~guermeur, accessed on 2012.07.31.

95. Guo-an Y., Hong-bing X., Chao W. Design and implementation of an agent-oriented expert system of loan risk evaluation. Proc. Of International Conference on Integration of Knowledge Intensive Multi-Agent Systems, pp. 41-45 (2003).

96. Hall M.A. Correlation-based feature subset selection for machine learning. PhD thesis, Hamilton, New Zealand, 1998.

97. Hall M.A. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. Working paper, 2000, available at http://www.cs.waikato.ac.nz/pubs/wp/2000/uow-cs-wp-2000-08.pdf.

98. Hamilton A. IT62 Decision Support Systems. Lectures. University of Stirling 2004.

99. Han J., Kamber M. Data Mining: Concepts and Techniques, 1st ed. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor, Morgan Kaufmann Publishers, 2001.

100. Hao Y., Chi Z., Yan D. Fuzzy Support Vector Machine Based on Vague Sets for Credit Assessment. FSKD '07 Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 1, 2007, pp. 603–607.

101. Hao P.-Y, Lin M.-Sh., Tsai L.-B. A New Support Vector Machine with Fuzzy Hyper-Plane and Its Application to Evaluate Credit Risk. 2008 Eighth International Conference on Intelligent Systems Design and Applications, Vol. 3, 2008, pp.83-88

102. Haupt R. L., Haupt S. E. Practical Genetic Algorithms, 2nd ed. Wiley, 2004.

103. Härdle W., Moro R., Schafer D. Estimating probabilities of default with support vector machines. SFB 649 Discussion Papers SFB649DP2007-035, Sonderforschungsbereich 649, Humboldt University, Berlin, Germany, 2007.

104. Henke S. Artificial Intelligence History.: http://www.stottlerhenke.com/ai_general/history.htm, accessed on 2012.07.31.

105. HeroSvm 2.1, http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html, accessed on 2012.07.31

106. Holsapple C. W. DSS Architecture and Types. In: Burstein F., Holsapple C. W. (eds). Handbook on Decision Support Systems 1: Basic Themes. Springer-Verlag Berlin Heidelberg, 2008.

107. Holsapple C. A. Whinston. Decision Support Systems: A Knowledge-Based Approach. West Publishing Company , 1996.

108. Hsieh C.-J., Chang K.-W., Lin C.-J., Keerthi S. S., Sundararajan S.. A dual coordinate descent method for large-scale linear SVM. Proceedings of the 25th international conference on Machine learning (ICML 2008), pp. 408-415.

109. Hsu C., Lin C. A simple decomposition method for support vector machines. Machine Learning, Vol. 46, 2002, pp. 291–314.

110. Hu Y., Li Y. LS-SVM for bad debt risk assessment in enterprises. Proceedings of 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1665–1669.

111. Huai, W. The Framework Design and Research On Enterprises Group Financial Decision Support System. In: Proceedings of Management and Service Science (MASS), 2010, pp. 1-4, Wuhan.

112. Huang A. Z., Chen A. H., Hsu A. C.-J., Chen B. W-H., Wu, S. Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision Support Systems Vol. 37, 2004, pp. 543– 558.

113. Huang C.-L., Chen M., Wang C. Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, Vol. 33, No. 4, 2007, pp. 847–856.

114. Huang F. A Particle Swarm Optimized Fuzzy Neural Network for Credit Risk Evaluation, Proc. of 2008 Second International Conference on Genetic and Evolutionary Computing, 2008, pp. 153-157.

115. Huang J., Zhou H. Specification analysis of structural credit risk models. Finance and Economics Discussion Series 2008-55, Board of Governors of the Federal Reserve System (U.S.), 2008.

116. Huang T. M., Kecman V., Kopriva I. Kernel based algorithms for mining huge data sets: supervised, semi-supervised, and unsupervised learning. Studies in Computational Intelligence, Vol.17 (editor J. Kacprzyk), Springer Verlag, 2006.

117. Jayanthi J., Suresh J., Vaishnavi J. Bankruptcy Prediction using SVM and Hybrid SVM Survey. International Journal of Computer Applications, Vol. 34, No. 7, 2011, pp. 39–45.

118. Jiang M.-H., Yuan X.-C.. Construction and Application of PSO-SVM Model for Personal Credit Scoring. Proceedings of the 7th international conference on Computational Science, Part IV: ICCS 2007 (ICCS '07), Yong Shi, Geert Dick Albada, Jack Dongarra, and Peter M. Sloot (Eds.). Springer-Verlag, Berlin, Heidelberg, 2007, pp. 158-161.

119. Joachims Th. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

120. Joachims Th. SVMlight - Support Vector Machine, http://www.cs.cornell.edu/people/tj/svm_light, accessed on 2012.07.31

121. Joachims Th..SVM^{multiclass} - Multi-Class Support Vector Machine, http://svmlight.joachims.org/svm_multiclass.html, accessed on 2012.07.31

122. Joachims, T. Text categorization with support vector machines: learning with many relevant features. Proceedings of ECML-98, 10th European Conference on Machine Learning, 1998, pp. 137–142.

123. Jolliffe I.T., Principal Component Analysis, Springer, New York, 1986.

124. Kan S. H. Metrics and Models in Software Quality Engineering, 2nd ed. Addison Wesley, 2002.

125. Kancerevyčius G. Finansai ir investicijos (in Lithuanian). Kaunas, Smaltija, 2004.

126. Kaski S., Sinkkonen J., Peltonen J. Bankruptcy Analysis with Self-Organizing Maps in Learning Metrics. IEEE Transactions on Neural Networks, 2001.

127. Kennedy J., Eberhart R. C., Shi Y. Swarm intelligence. Morgan Kaufmann Publishers, 2001.

128. Kennedy J. The Particle Swarm: Social Adaptation of Knowledge. Proceedings of the IEEE International Conference on Evolutionary Computation, 1997, pp. 303–308.

129. Khan A., Baharudin B., Lee L.H., Khan Kh. A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances in Information Technology, Vol 1, No 1, pp. 4-20, 2010, doi:10.4304/jait.1.1.4-20.

130. Kim K., Ahn H. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. Computers & Operations Research, Vol. 39, No. 8, 2012, pp. 1800–1811.

131. Kin Y.G., Ding W.X. A DSS for Bank Credit Evaluation under Risk. National University at Singapore, Technical Report, 1995, access via the Internet http://dl.comp.nus.edu.sg/dspace/handle/1900.100/1393 (referred on 2012.08.19)

132. Knebel T., Hochreiter S.. Software Documentation of the Potential Support Vector Machine http://ni.cs.tu-berlin.de/software/psvm/PSVMDocumentation.pdf, accessed on 2012.07.31

133. Koster A. Expert Systems for Management Of Financial Regulations: Application To Workers' Compensation Insurance Premium Auditing And Evaluation. Managerial Finance, Vol. 15, Issue 5, 1993, pp.7 – 18

134. Kotsiantis S., Kanellopoulos D. Multi-instance learning for predicting fraudulent financial statements. Proceedings of Third International Conference on Convergence and Hybrid Information Technology (ICCIT '08), Vol. 1, 2008. pp. 448-452.

135. Kotsiantis S. B., Kanellopoulos D., Karioti V., Tampakas V. An ontology-based portal for credit risk analysis. 2nd IEEE International Conference on Computer Science and Information Technology, 2009, pp. 165-169

136. Kou G., Peng Y., Shi Y., Chen Z. Multiclass Credit Cardholders' Behaviors Classification Methods. ICCS'06 Proceedings of the 6th international conference on Computational Science - Volume Part IV, 2006, pp. 485-492.

137. Kubat M., Holte R., Matwin S. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning, Vol. 30, 1998, pp. 195–215.

138. Kuo H.-Ch. Cognitive Management System Based Support for Bank Credit Granting Decision: An Integrated and Practical Design for Taiwan. Journal of Systems Integration, 1997, Vol. 7, pp. 77–91

139. Lai K. K., Yu L., Wang S., Huang W. An Intelligent CRM System for Identifying High-Risk Customers: An Ensemble Data Mining Approach. ICCS '07 Proceedings of the 7th international conference on Computational Science, Part II, 2007, pp. 486-489.

140. Lai K. K., Yu L., Wang S., Zhou L. Credit risk analysis using a reliability-based neural network ensemble model. ICANN'06 Proceedings of the 16th international conference on Artificial Neural Networks, Part II, 2006, pp. 682–690.

141. Lai K. K., Zhou L., Yu L. A Two-Phase Model Based on SVM and Conjoint Analysis for Credit Scoring. ICCS '07 Proceedings of the 7th international conference on Computational Science, Part II, 2007, pp. 494–498.

142. Lee Y.-J., Mangasarian O. L. A Smooth Support Vector Machine. Computational Optimization and Applications, Vol. 20, 2001, pp.5-22.

143. Lessmann S. Solving imbalanced classification problems with support vector machines. Proceedings of International Conference on Artificial Intelligence, 2004, pp. 214–220

144. Lewis D. D., Gale W. Training text classifiers by uncertainty sampling. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-94), 1994, pp.3-12.

145. Li S., Tsang I. W., Chaudhari N. S. Relevance vector machine based infinite decision agent ensemble learning for credit risk analysis. Expert Systems with Applications, Vol. 39, No. 5, 2012, pp. 4947–4953.

146. Lin C.-J., Weng R. C., Keerthi S. S. Trust region Newton method for large-scale logistic regression. Journal of Machine Learning Research 9 (2008), pp. 627-650

147. Liu X., Fu H., Lin W. A Modified Support Vector Machine model for Credit Scoring. International Journal of Computational Intelligence Systems, Vol. 3(6), 2010, pp. 797–804.

148. Lo Sh.-Ch., Lin Ch.-Ch., Chuang Y.-Ch. Using Support Vector Machine and Sequential Pattern Mining to Construct Financial Prediction Model. Proceedings of 2008 IEEE Asia-Pacific Services Computing Conference, 2008, pp. 993-998.

149. Longman Dictionary of Contemporary English, 4th ed, http://www.ldoceonline.com/, accessed on at 2012.09.16.

150. Lv G.-L., Peng L. Commercial Banks' Credit Risk Assessment Based on Rough Sets and SVM. Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008, pp. 1-4.

151. Mahmoud M., Algadi N., Ali A. Expert System for Banking Credit Decision. 2008 International Conference on Computer Science and Information Technology, 2008, pp. 813–819.

152. Mahotra R., Mahotra D.K. Differentiating between good credits and bad credits using neuro-fuzzy systems. European Journal of Operational Research, Vol. 136, 2002, pp. 190-211.

153. Mangasarian O. L., Musicant D. Active Support Vector Machine Classification. Advances in Neural Information Processing Systems 13, MIT-Press, 2000, pp. 577-583.

154. Mangasarian O. L., Musicant D. Lagrangian Support Vector Machine Classification. The Journal of Machine Learning Research, Vol. 1, 2001, pp.161 – 177.

155. Martinez W. L., Martinez A. R. Computational statistics handbook with MATLAB. Chapman & Hall/CRC Press, 2002.

156. Martišius I., Damaševičius R., Jusas V., Birvinskas D. Using higher order nonlinear operators for SVM classification of EEG data. Electronics and Electrical Engineering. No. 3(119). 2012, pp. 99-102.

157. Masnadi-Shirazi H., Vasconcelos N. Risk minimization, probability elicitation, and cost-sensitive SVMs. Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 759–766.

158. Matsasinis F.N. CCAS: An Intelligent Decision Support System for Credit Card Assessment. Journal Of Multi-Criteria Decision Analysis, Vol. 11, 2002, pp. 213–235.

159. Merkevičius E. Savitvarkių neuroninių tinklų- diskriminantinio modelio tyrimai kredito rizikos vertinimo sprendimų paramos sistemoje (in Lithuanian). PhD thesis, Vilnius University, 2007.

160. Merkevičius E., Garšva G., Cepkovataja O. Intelektualios sprendimų paramos sistemos kredito rizikos vertinimui struktūra (in Lithuanian). Proc. of Informacinės technologijos'2006, 2006, pp.725-733.

161. Merkevicius E., Garsva G., Girdzijauskas S. A hybrid SOM-Altman model for bankruptcy prediction. Lecture Notes in Computer Science, Vol. 3994, 2006, pp. 364-371.

162. Merkevicius E., Garsva G., Simutis R. Neuro-discriminate Model for the Forecasting of Changes of Companies Financial Standings on the Basis of Self-organizing Maps. Lecture Notes In Computer Science, Vol. 4488, 2007, pp. 439-446.

163. Merriam-Webster's Collegiate® Dictionary, 11th ed., online access at http://www.merriam-webster.com/netdict.htm, accessed on 2012.09.12

164. Min, J. H., Lee, Y.-Ch. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Systems with Applications Vol. 28, Issue 4 , 2005, pp. 603-614.

165. Min Z. Credit Risk Assessment Based on Fuzzy SVM and Principal Component Analysis. Proceedings of 2009 International Conference on Web Information Systems and Mining, No. 1, 2009, pp. 125–127.

166. Mitchell T. Machine learning. McGraw-Hill Science/Engineering/Math, 1997.

167. Moody's - credit ratings, research, tools and analysis for the global capital markets, http://www.moodys.com/, accessed on 2012.08.06.

168. Mora J. Open Source Solutions in XBRL. XI European Banking Supervisors XBRL Workshop, Vienna, Austria, November 16-20, 2009, http://www.eurofiling.info/11th_workshop/presentations/JMora_OpenSourceSolutions.ppt, accessed on on 2012.09.15.

169. Nash M. Java Frameworks and Components: Accelerate Your Web Application Development. Cambridge Press, 2003.

170. Nedović L., Devedžić V. Expert systems in finance – a cross-section of the field. Expert Systems With Applications, Vol.23, No.1, 2002, pp. 49-66.

171. Ohlson J. Financial Ratios and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research Vol. 18(1), 1980, pp. 109-131.

172. Palmer S. R., Felsing, J. M. A Practical Guide to Feature-Driven Development. Prentice Hall, 2002.

173. Ping Y., Yongheng L. Neighborhood rough set and SVM based hybrid credit scoring classifier. Expert Systems with Applications, Vol. 38, No. 9, 2011, pp. 11300–11304.

174. Piramuthu S. Financial credit-risk evaluation with neural and neurofuzzy systems. European Journal of Operational Research, Vol. 112, 1999, pp. 310-321.

175. Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines. Advances in Kernel Methods-Support Vector Learning, 1999, pp. 185 – 208.

176. Power D. J. Decision Support Systems: A Historical Overview. In: Burstein F., Holsapple C. W. (eds). Handbook on Decision Support Systems 1: Basic Themes. Springer-Verlag Berlin Heidelberg, 2008.

177. Raynor W. J. The international dictionary of artificial intelligence. The Glenlake Publishing Company, Ltd., 1999.

178. RapidMiner, http://rapid-i.com/content/view/181/190, accessed on 2012.07.31

179. Ravikumar P., Ravi V. Bankruptcy Prediction in Banks by an Ensemble Classifier. Proc of International Conference on Industrial Technology (ICIT 2006),2006, pp. 2032-2036.

180. Rechenberg I. Evolutionsstrategie (in German). Stuttgart: Holzmann-Froboog, 1973.

181. Ribeiro B., Vieira A., Duarte J., Silva C., Neves J., Liu Q., Sung A. Bankruptcy Analysis for Credit Risk using Manifold Learning. Proceedings of the ICONIP 2008, International Conference on Neural Information Processing, Auckland, New Zealand, 2008.

182. Ribeiro B., Vieira A., Neves J.C. Sparse Bayesian Models: Bankruptcy-Predictors of Choice?. Proceedings of 2006 International Joint Conference on Neural Networks, Vancouver, Canada, 2006, pp. 3377-3381.

183. Romanelli M.. Proposal for a COREP-metamodel (paper and presentation). II European Banking Supervisors XBRL Workshop (Madrid), 18th-21st April 2005, http://www.eurofiling.info/2nd_workshop/presentations/ProposalCOREPMetamodel-MicheleRomanelli.zip, accessed on 2012.09.12.

184. Rüping S. mySVM - a support vector machine, http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html, accessed on 2012.07.31

185. Russell S. J., Norvig P. Artificial Intelligence - A Modern Approach, 2nd ed. Pearson Education, 2010.

186. S&P Ratings, http://www.standardandpoors.com/ratings/en/us/. accessed on 2012.08.06.

187. Sands E. G., Springate G. L.V., Var T. Predicting Business Failures, CGA Magazine, 1983, pp. 24-27.

188. Saunders A., Allen L. Credit risk measurement: new approaches to value at risk and other paradigms. John Wiley & Sons, Inc., 2002.

189. Schlottmann F., Seese D. A Hybrid Genetic-Quantitative Method For Risk-Return Optimisation Of Credit Portfolios (2001). Proceedings of the Conference of Quantitative Methods in Finance, University of Technology, Sydney, Australia, 2001, p. 55.

190. Schlottmann F., Seese D. Hybrid Multi-Objective Evolutionary Computation of Constrained Downside Risk-Return Efficient Sets for Credit Portfolios, No 78, 2002, Computing in Economics and Finance 2002 from Society for Computational Economics.

191. Schölkopf B. The kernel trick for distances. Advances in Neural Information Processing Systems, 2001, pp. 301–307.

192. Schölkopf B., Smola A., Williamson R., Bartlett P. New support vector algorithms. Neural Computation, Vol. 23, No. 1, 2000, pp. 60–73.

193. Schwefel H.-P. Numerische Optimierung von Computer-Modellen (PhD thesis)(in German), 1974.

194. SEC.gov. What Is Interactive Data and Who's Using It?, http://www.sec.gov/spotlight/xbrl/what-is-idata.shtml, accessed on 2012.09.15

195. Shalev-Shwartz S., Singer Y., Srebro N. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. Proceedings of the 24th international conference on Machine learning (ICML '07), 2007, pp. 807–814.

196. Shao Y. P., Wilson A., Oppenheim C. Expert systems in UK banking. Proceedings the 11th Conference on Artificial Intelligence for Applications, 1995, pp. 18–23.

197. Shumway T. Forecasting bankruptcy more accurately: A simple hazard model. Journal of Business, Vol. 74(1), 2001, pp. 101–124.

198. SimpleMKL Toolbox, http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html, accessed on 2012.07.31

199. Sindhwani V., Keerthi S. S. Newton methods for fast solution of semi-supervised linear SVMs. Large Scale Kernel Machines (eds. L. Bottou, O. Chapelle, D. DeCoste, J. Weston), MIT Press, Cambridge, MA, 2007.

200. Springate G. L. V. Predicting the Possibility of Failure in a Canadian Firm, Unpublished M.B.A. Research Project, Simon Fraser University, 1978.

201. Stasytytė V. Investicijų portfelio sudarymas naudojant sprendimų paramos sistemą. Business: Theory and Practice, Vol.13(3), 2012, pp. 253–263.

202. Steinwart I., Christmann A. Support vector machines. Springer Science+Business Media, LLC, 2008.

203. Suykens J., Vandewalle J. Least Squares Support Vector Machine Classifiers. Neural Processing Letters, Vol. 9, No. 3, 1999, pp. 293–300.

204. Sun L., Shenoy P. P. Using Bayesian networks for bankruptcy prediction: Some methodological issues. European Journal of Operational Research, Vol. 180, No. 2, 2007, pp. 738–753.

205. SVMdark, http://www.cs.ucl.ac.uk/staff/M.Sewell/svmdark, accessed on 2012.07.31

206. SVMlin - Fast Linear SVM Solvers for Supervised and Semi-supervised Learning, http://people.cs.uchicago.edu/~vikass/svmlin.html, accessed on 2012.07.31

207. Taffler R. Forecasting company failure in the UK using discriminant analysis and financial ratio data, Journal of the Royal Statistical Society. Series A (General) Vol. 145(3), 1982, pp. 342-358.

208. Tang Y., Zhang Y.-Q., Chawla N. V., Krasser S. SVMs modeling for highly imbalanced classification. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society, vol. 39, no. 1, 2009, pp. 281–288.

209. Thai-Nghe N., Gantner Z., Schmidt-Thieme L. Cost-sensitive learning methods for imbalanced data. Proceedings of The 2010 International Joint Conference on Neural Networks (IJCNN) (2010), pp. 1-8.

210. Tian Y., Shi Y., Liu X. Recent advances on support vector machines research. Technological and Economic Development of Economy, Vol. 18, No. 1, 2012, pp. 5–33.

211. TinySVM: Support Vector Machines, http://chasen.org/~taku/software/TinySVM, accessed on 2012.07.31

212. Tipping M. E. Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research, Vol. 1(3), 2001, pp. 211-244.

213. Tsai C., Wu J. Using neural network ensembles for bankruptcy prediction and credit scoring, Expert Systems with Applications Vol. 34(4), 2008, pp. 2639-2649.

214. Tsai C.-F. Feature selection in bankruptcy prediction. Knowledge-Based Systems, Vol. 22, No. 2, 2009, pp. 120–127.

215. Tsaih R., Liu Y-J., Liu W., Lien, Y.-L. Credit scoring system for small business loans. Decision Support Systems, Vol. 38(1), 2004, pp. 91-99.

216. Tsang I. W., Kocsor A., Kwok J. T. Simpler core vector machines with enclosing balls. Proceedings of the Twenty-Fourth International Conference on Machine Learning, Corvallis, USA, 2007, pp.911-918.

217. Tsang W., Kwok J. T., Cheung P. M. Core vector machines: Fast SVM training on very large data sets. Journal of Machine Learning Research, Vol. 6, 2005, pp.363-392.

218. Turban E., Watkins P.R. Integrating Expert Systems and Decision Support Systems. MIS Quarterly, Vol. 10(2), 1986, pp. 121−136.

219. UCLA-LoPucki Bankruptcy Research Database, http://lopucki.law.ucla.edu/index.htm

220. UniverSVM, http://www.kyb.mpg.de/bs/people/fabee/universvm.html, accessed on 2012.07.31

221. van den Berg J. Credit Rating Prediction with Self-Organizing Maps. Expert Systems with Applications, Volume 30, Issue 3, April 2006, Pages 479-487.

222. van Gestel T., Baesens B. Credit Risk Management: Basic Concepts. Oxford University Press, USA, 2009.

223. van Gestel T., Baesens B., Suykens J. A.K., van Den Poel D., Baestaens D.-E., Willekens M. Bayesian kernel based classification for financial distress detection. European Journal of Operational Research, Vol. 172(3), 2006, pp. 979-1003.

224. van Hulse J., Khoshgoftaar T.M., Napolitano A. Experimental perspectives on learning from imbalanced data. Proceedings of the 24th international conference on Machine learning ICML '07, 2007, pp. 935-942.

225. Vapnik V. N. Statistical learning theory. New York: Wiley, 1998.

226. Varoneckas A. Hibridinis RR intervalų sekų modelis miego stadijoms atpažinti (in Lithuanian). PhD Thesis, Vilnius University, Institute of Mathematics and Informatics, 2009.

227. Vellido A., Lisboa P. J. G., Vaughan B. Neural networks in business: a survey of applications (1992 – 1998), Expert Systems with Applications, Vol. 17, 1999, pp. 51-70.

228. Verikas A., Gelžinis A. Neuroniniai tinklai ir neuroniniai skaičiavimai (in Lithuanian). Kaunas University of Technology, 2008.

229. Verikas A., Gelzinis A., Kovalenko M., Bacauskiene M. Selecting features from multiple feature sets for SVM committee-based screening of human larynx. Expert Systems with Applications, Vol. 37(10), 2010, pp. 6957-6962.

230. Vieira A., Ribeiro B., Duarte J., Silva C., Carvalho das Neves, J., Mukkamala, S., Sung A H. Improving Personal Credit Scoring with HLVQ. ICONIP, Lecture Notes on Computer Science, Springer, Auckland, 2008.

231. Wang B., Liu Y., Hao Y., Liu Sh. Defaults Assessment of Mortgage Loan with Rough Set and SVM. International Conference on Computational Intelligence and Security (CIS 2007), 2007, pp.981-985.

232. Wang B., Wang D., Liu Sh., Hao Y. Research of Housing Loan Credit Evaluation Based SVM. 2008 Fourth International Conference on Natural Computation, Vol. 2, 2008, pp.144-147.

233. Wang G., Ma J. A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine. Expert Systems with Applications, Vol. 39, No. 5, 2012, pp. 5325–5331.

234. Wang X. Corporate Financial Warning Model Based on PSO and SVM. Proceedings of 2nd International Conference on Information Engineering and Computer Science (ICIECS), 2010, pp. 1–5.

235. Wei L., Li J., Chen Z. Credit Risk Evaluation Using Support Vector Machine with Mixture of Kernel. ICCS '07 Proceedings of the 7th international conference on Computational Science, Part II, 2007, pp. 431–438.

236. Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka, accessed on 2012.07.31.

237. Weston J., Guyon I. Support vector machine—recursive feature elimination (SVM-RFE). U.S. Patent 8095483, issued Jan 10, 2012.

238. Wong B. K., Lai V. S., Lam J. A bibliography of neural network business applications research: 1994-1998, Computers & Operations Research, Vol. 27, 2000, pp. 1045-1076.

239. Wong B., Selvi Y. Neural network applications in finance: A review and analysis of literature (1990 - 1996), Information & Management, Vol. 34, 1998, pp. 129-139.

240. Wu C.-H., Tzeng G.-H., Goo Y.-J., Fang W.-C.A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. Expert Systems with Applications, Vol. 32, No. 2, 2007, pp. 397–408.

241. XBRL.SEC.GOV, http://xbrl.sec.gov, accessed on 2012.09.15

242. XBRL International Inc. XBRL Abstract Model 1.0, Public Working Draft 19 October 2011, accessed online at http://xbrl.org/Specification/abstractmodel-primary/PWD-2011-10-19/abstractmodel-primary-PWD-2011-10-19.html, accessed on 2012.09.15

243. XBRL International Inc. XBRL Abstract Model 2.0, Public Working Draft 06 June 2012, accessed online at http://xbrl.org/Specification/abstractmodel-primary/PWD-2012-06-06/abstractmodel-primary-pwd-2012-06-06.html, accessed on 2012.09.15

244. XBRL International Inc. XBRL Dimensions 1.0, http://www.xbrl.org/Specification/XDT-REC-2006-09-18.htm, accessed on 2012.09.15

245. XBRL International Inc. Formula 1.0, http://www.xbrl.org/Specification/formula/REC-2009-06-22/formula-REC-2009-06-22.html, accessed on 2012.09.15

246. XBRL International Inc. Inline XBRL 1.0 Background information and guidance, supporting document for a Recommendation 20 April 2010, http://www.xbrl.org/Specification/inlineXBRL/REC-2010-04-20/inlineXBRL-background-REC-2010-04-20.html, accessed on 2012.09.15.

247. XBRL International Inc. XBRL Versioning Specification 1.0, Public Working Draft, http://www.xbrl.org/Specification/Versioning/XVS-PWD-2007-11-28.htm, accessed on 2012.09.15

248. Yang C.-G., Duan X.-B. Credit risk assessment in commercial banks based on SVM using PCA. Proc. Of 2008 International Conference on Machine Learning and Cybernetics, Vol. 2, 2008, pp. 1207–1211.

249. Yang Z. R. Support vector machines for company failure prediction. Proceedings of 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003, pp. 47–54.

250. Yang Z. R. Biological applications of support vector machines. Brief Bioinformatics, Vol. 5(4), 2004, pp. 328-338.

251. Yang Z., You W., Ji G. Using partial least squares and support vector machines for bankruptcy prediction. Expert Systems with Applications, Vol. 38, No. 7, 2011, pp. 8336–8342.

252. Yoon J. , Kwon Y. S. Roh T. H. Performance Improvement of Bankruptcy Prediction using Credit Card Sales Information of Small & Micro Business. Proceedings of 5th ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2007), 2007, pp.503-512.

253. Yu L., Lai K. K., Wang S., Zhou L. Bio-Inspired Credit Risk Analysis. Berlin Heidelberg, Springer, 2008.

254. Yue Y. SVM^{map}. Support Vector Machine for Optimizing Mean Average Precision, http://projects.yisongyue.com/svmmap, accessed on 2012.07.31

255. Yue Y. SVM^{div}. Support Vector Machine for Predicting Diverse Subsets, http://projects.yisongyue.com/svmdiv, accessed on 2012.07.31

256. Yun L., Cao Q., Zhang H. Application of the PSO-SVM model for Credit Scoring. Proceedings of 2011 Seventh International Conference on Computational Intelligence and Security, 2011, pp. 47–51.

257. Zahedi F., Jaeki Song J., Jarupathirun S. Web-Based Decision Support. In: Handbook on Decision Support Systems 1: Basic Themes. Springer-Verlag Berlin Heidelberg, 2008, pp. 315-338.

258. Zanni L., Serafini T., Zanghirati G. Parallel Software for Training Large Scale Support Vector Machines on Multiprocessor Systems. Journal of Machine Learning Research, Vol. 7(Jul), 2006, pp. 1467-1492.

259. Zhang D., Chen Q., Wei L.Building Behavior Scoring Model Using Genetic Algorithm and Support Vector Machines. ICCS '07 Proceedings of the 7th international conference on Computational Science, Part II, 2007, pp. 482–485.

260. Zhang, M., Gu, Y., Zhu, J.: Analysis of the Framework for Financial Decision Support System. Proceedings of 2009 International Conference on Wireless Networks and Information Systems, Shanghai, 2009, pp. 241-244.

261. Zhou J., Bai T. Credit Risk Assessment Using Rough Set Theory and GA-Based SVM. 2008 The 3rd International Conference on Grid and Pervasive Computing - Workshops, 2008, pp.320-325

262. Zhou J. , Tian J . Credit risk assessment based on rough set theory and fuzzy support vector machine. Advances in Intelligent Systems Research, ISKE-2007 Proceedings. Atlantis Press, 2007.

263. Zhou J., Zhang A., Bai T. Client Classification on Credit Risk Using Rough Set Theory and ACO-Based Support Vector Machine. Proceedings of 4th International Conference onWireless Communications, Networking and Mobile Computing (WiCOM '08), 2008, pp.1-4.

264. Zhou L., Lai K. K. Weighted LS-SVM Credit Scoring Models with AUC Maximization by Direct Search. Proceedings of 2009 International Joint Conference on Computational Sciences and Optimization, 2009, pp. 7–11.

265. Zhou L., Lai K. K. Multi-Agent Ensemble Models Based on Weighted Least Square SVM for Credit Risk Assessment. 2009 WRI Global Congress on Intelligent Systems, 2009, pp. 559–563.

266. Zhou Q., Lin Ch. Credit Risk Assessment in Commercial Banks Based on Fuzzy Support Vector Machines. 9th International Conference on Control, Automation, Robotics and Vision, 2006, pp. 1 – 4

267. Zmijewski M. Methodological Issues Related to the Estimation of Financial Distress Prediction Models. Journal of Accounting Research Vol.22, 1984, pp. 59-82.

LIST OF APPENDICES

Appendix A

DEFINITION AND TAXONOMY OF DECISION SUPPORT SYSTEMS

A.1. Taxonomy of decision support systems

Several taxonomies of DSS can be found in the literature by different authors. One of the oldest can be found in [3; cited by 176] which was published in 1980 when Alter analysed 56 DSS and classified them to seven categories:

- File drawer systems that provide access to data items.
- Data analysis systems that support the manipulation of data by computerized tools tailored to a specific task and setting or by more general tools and operators.
- Analysis information systems that provide access to a series of decision-oriented databases and small models.
- Accounting and financial models that calculate the consequences of possible actions.
- Representational models that estimate the consequences of actions on the basis of simulation models.
- Optimization models that suggest an action according to an optimal solution consistent with provided constraints.
- Suggestion models that perform the logical processing leading to a specific suggested decision for a fairly structured or well-understood task.

It can be seen that DSS defined in Alter‘s taxonomy is not only related to information systems but also comprises modeling and optimization techniques. Therefore in 1980s DSS was considered not as only as an information system but, more generally, as an ingelligent tool to guide decision support. In 1996 Holsapple and Whinton had identified five types of DSS - text-oriented DSS, database-oriented DSS, spreadsheet-oriented DSS, solver-oriented DSS and rule-oriented DSS - according to knowledge management techniques that they are based on [106]. Holsapple renewed his taxonomy in 2008, distinguishing such categories [106]:

- *Text-Oriented DSS* – these DSS use knowledge bases which consists of various documents that can be manipulated, searched or analysed;

- *Hypertext-Oriented DSS* – similarly to text-oriented DSS, these type of systems also use sets of ducuments which are connected using hyperlinks which cane be also created, deleted or traversed through. The user does not have to memorize all the documents but rather can memorize only main links to the particular concepts and/or connections;
- *Database-Oriented DSS* – these systems use retrieval and manipulation of data stored in DBMS using the query processor and custom-built processors for particular tasks. Modern systems of this type usually use data warehouses to store and retrieve data from for further analysis;
- *Spreadsheet-Oriented DSS* – DSS systems which use spreadsheet technique for knowledge management. Users can create, view, and modify procedural knowledge assimilated in the knowledge base, as well as tell the problem solving engine to process the instructions in the spreadsheet [106].
- *Solver-Oriented DSS* – DSS of such type use solvers to solve any of a particular class of problems which depend to a particular domain (finance, economics, inverstment, insolvency analysis) or problem type (optimization, forecasting, planning, statistical analysis). The DSS might be specialized to a particular problem or a set of problems, or be capable to solve several types of problems. The problems can be presented in a fixed form (e.g., dataset) or defined in a particular domain specific language (e.g., programming language for statistical analysis) which provides more flexibility to solving various problems. The solvers can be executed in a predefined sequence to solve a particular problem with outputs from one solver as inputs of another.
- *Rule-Oriented DSS* – such DSS usually process a set of user defined rules (usually in form of *if...then* to solve particular tasks and give the user an advice and its explanation, as well as explanation of its performance during reasoning process for more detail analysis. Holsapple also classifies expert systems as rule-oriented DSS [106].
- *Compound (integrated) DSS* – these DSS combine functionality and capabilities of several types of DSS;
- *Multiparticipant DSS* – these DSSs are used by several participants which do not have authorities to make the whole decision but who can influence this decision by contributing to it [106]. Such systems are often referred as group DSS.

They often concentrate on group decision making and tasks involved such as negotiation.

Arnott and Pervan (2005) distinguished such DSS groups in their research: personal DSS, group support systems, negotiation support systems, intelligent DSS, knowledge-management based DSS, executive information systems/business intelligence, and data warehousing [15]. They also enhanced Alter's taxonomy, by adapting it to personal DSS classification and specifying two subgroups: data-oriented DSS (file drawer, data analysis and analysis information systems) and model-oriented (accounting and financial models, representational models, optimization models and suggestion models). Power (2002) also gives his taxonomy which is includes five groups of DSS [176]. These groups are based on the problems that they are targeted at and dominant architectural components that are key factors in their development:

- *Communication-driven (group-driven) DSS* – these DSS are are used to statisfy goals of more than one person by establishing communication, collaboration and collaboration facilities using network and communications technologies,
- *Data-driven DSS* – these systems implement access and usage (retrieval and manipulation) of internal and external historical data, such as management data, time series, real-time data, etc. Such systems usually use data warehouse facilities to store and retrieve data, as well as advanced tools such as online analytical processing, data cubes and data mining. This is analogous to database-oriented DSS in Holsapple's taxonomy;
- *Knowledge-driven DSS* – these systems integrate knowledge and expertise of particular domain and its problems. Expert systems can be also viewed as a subset of such kind of DSS.
- *Document-driven DSS* – this kind of DSS is oriented usage of documents, such as text, images, media, in decision support. They also include document storage, search and processing components; the search component comprising advanced search abilities and text mining functionality is a key tool of these type of systems. This corresponds to text-oriented DSS in Holsapple's taxonomy;
- *Model-driven DSS* – this kind of DSS is based on financial, optimization, simulation and/or other specific models and their management. Such DSS use limited data (less than data-driven DSS) and models with parameters obtained automatically

or provided by decision makers. This corresponds to solver-oriented and rule-oriented DSS in Holsapple's taxonomy.

The field of decision support and decision support systems is still widely developed, with new suggestions and improvements, which might even lead to new types of DSS. For example, complex and/or distributed DSS which might include characteristics of several types of DSS described above might be developed using sophisticated techniques and infrastructure for computing. Technologies such as grid computing or Apache Hadoop[20] which enables large scale data processing using Map/Reduce paradigm enable more sophisticated large-scale computations, model development which would lead to improved quality of provided solutions. The rise of so-called "Big Data" and their availability over Internet and other sources using standardized formats proposes new possibilities. Thus integration with various data standards and data sources is a relevant topic. Therefore, this work tries to address this problem as well, proposing main principles and models for possible integration with financial data standards. The complexity of such integration also requires more suitable design methodologies therefore design of such system should also be discussed in more detail.

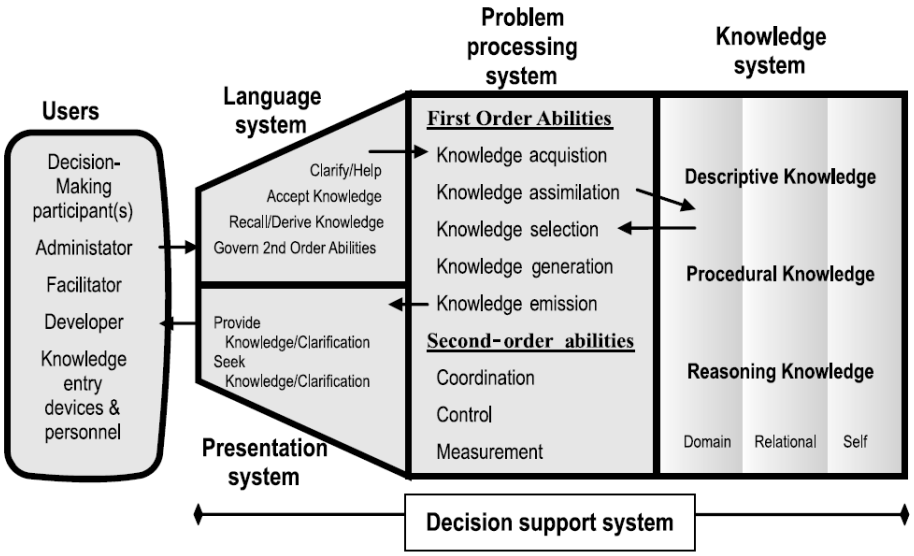
A.2. The structure of DSS and expert systems

Holsapple proposed basic message-driven DSS architecture composing of four essential components [106]:

- language system (LS), which consists of all messages the DSS can accept;
- presentation system (PS), which consists of all messages the DSS can emit;
- knowledge system (KS), that consists of all knowledge the DSS has stored and retained;
- a problem-processing system (PPS) to identify and solve problems.

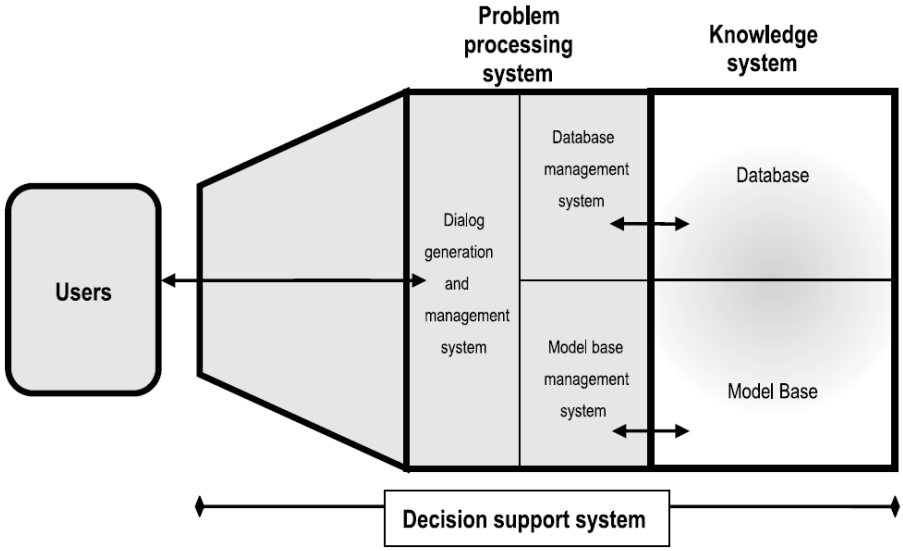
Holsapple also gives a variation of his basic architecture model for combined database and solver oriented DSS. Such framework is popular for development of DSS capable of financial forecasting, analysis and optimization tasks. DSS which are based on this framework and combine data warehouse facilities with analytical (such as OLAP) or data-mining solvers are heavily used in large organizations today. As it is also relevant in this research, this architecture is given in Figure 31.

[20] What Is Apache Hadoop?, http://hadoop.apache.org



Source: C. W.Holsapple DSS Architecture and Types.

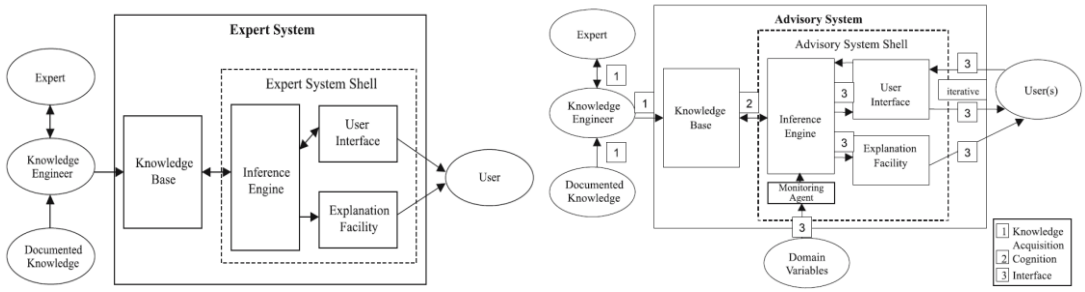
Figure 30. Holsapple's basic DSS architecture



Source: C. W. Holsapple DSS Architecture and Types.

Figure 31. Holsapple's combined database and solver oriented DSS architecture

Aronson and Turban describe basic architecture of expert system. Beemer and Gregg adapted works from Forslung (1995) and Mintzberg (1976) to give its extension to intelligent advisory decision support system. Both of these models are given in Figure 32.



a) Expert system architecture *b) Intelligent advisory system architecture*

Source: Beemer B. A., Gregg D. G. Advisory Systems to Support Decision Making.

Figure 32. Expert system and intelligent DSS architectures

The comparison between these two models shows that both of these systems have the same basic components:

- knowledge base for storing knowledge expressed in rules or other forms;
- inference engine which is the problem solving engine inferring results;
- user interface which is necessary to provide communication between user and system;
- explanation facility which is necessary to report explanations of the produced decision to the user.

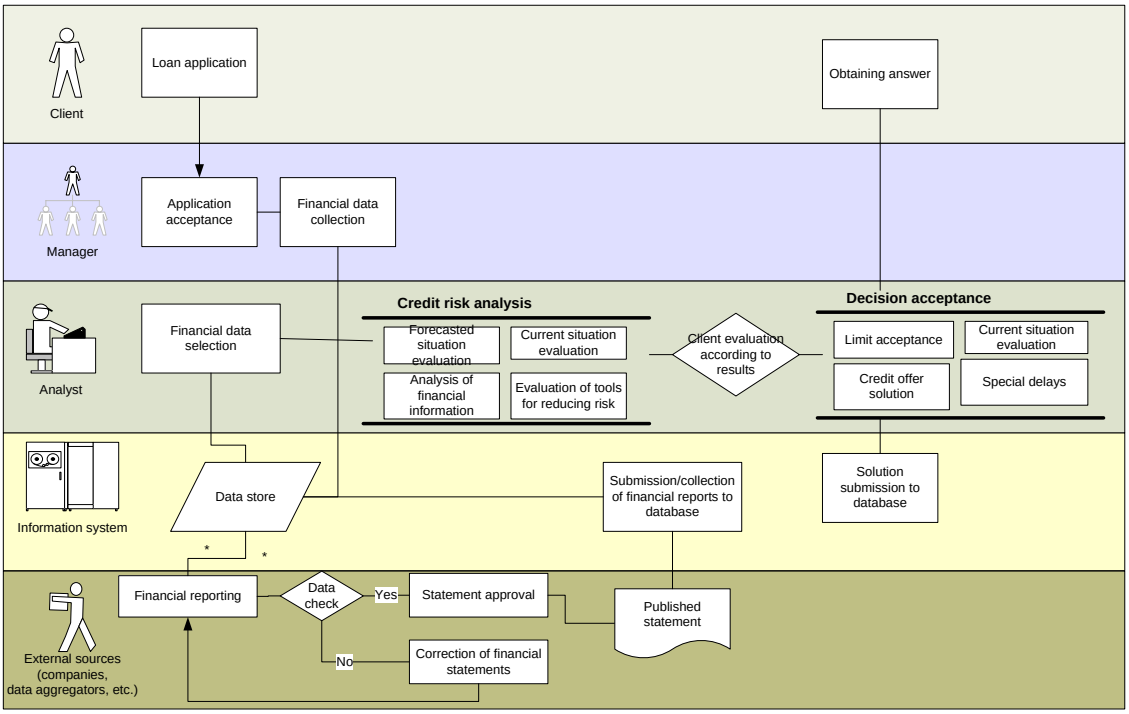
They also have the same roles (experts, knowledge engineer, user) and artifacts (documented knowledge). However, advisory systems include a monitoring agent to help identify the need for identifying unstructured decisions that need to be addressed and domain variables which are given to the inference engine [20]. According to Aronson and Turban, there are three main processes in expert systems - knowledge acquisition, inference, and interface; in advisory systems inference is replaced with cognition. Therefore, the processes which are in common may also differ, e.g., knowledge acquisition differs in the level of complexity of capturing, formalizing and organizing knowledge, interfacing is also more complex as additional user input to guide the overall decision-making process might be necessary. Cognition encapsulates the added functionality of active monitoring and problem or opportunity recognition; in addition to inference process, it also incorporates environment variables [20].

In addition these architectural differences, (Turban and Watkins, 1986; Beemer and Gregg, 2008) state that main differences between canonical expert

systems and intelligent advisory decision support systems [20, 218] lie in decision structure (in case of ES, structured vs unstructured in case of advisory DSS), AI methodology (rule-based approach vs case-based and machine learning approaches), role in decision process (decision making vs decision support). Intelligent DSS, conversely from expert systems, can identify the problem themselves, although it depends on the design of the system.

A.3. Main DSS processes in credit risk modelling

Six stages can de identified using DSS in credit issuing process [160]; a sequence diagram representing these stages is given in Figure 33. However this figure does not represent external information sources, such as lawyers, assets assessors, other information systems or facilities which act as data aggregators, etc.

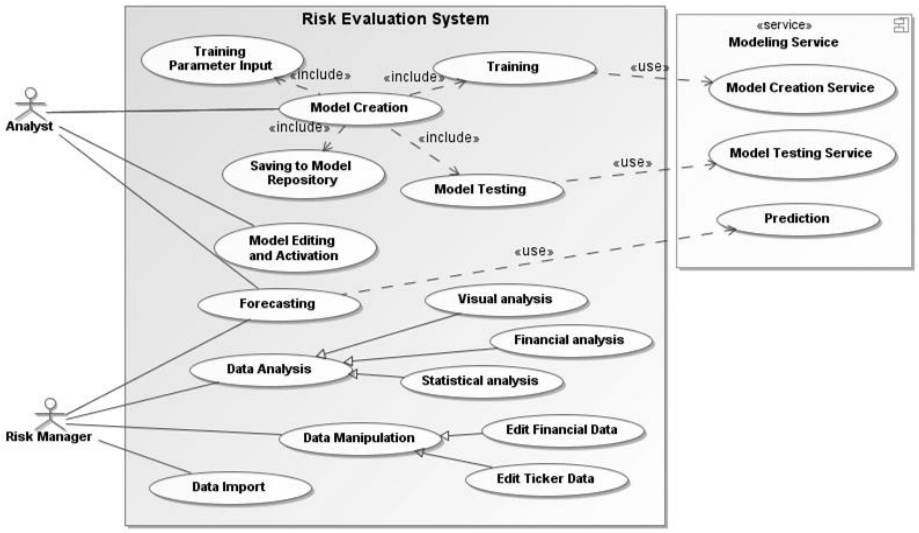


Source: created by the author according to [160].

Figure 33. Credit risk evaluation and issuing process using DSS

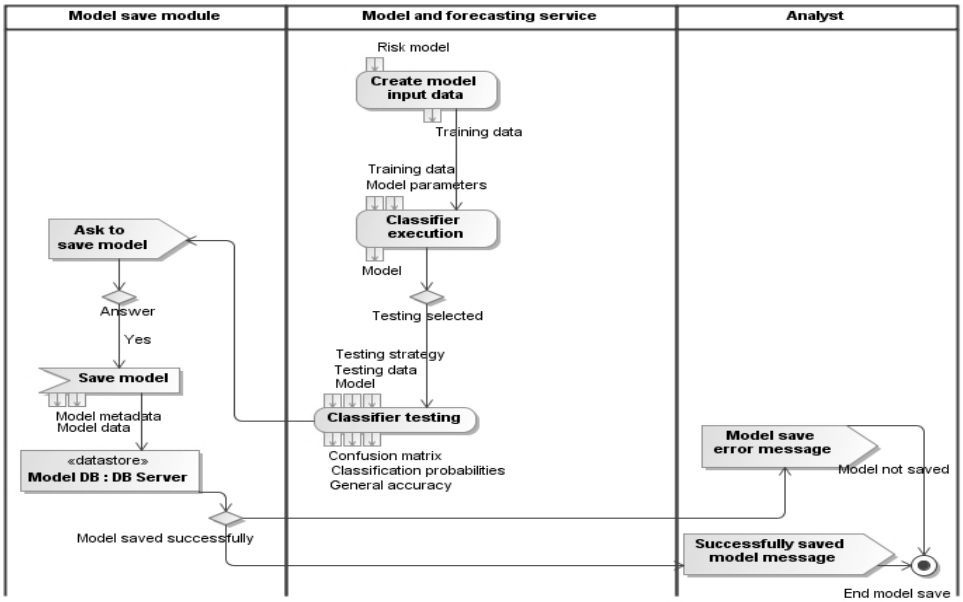
The role-system interaction can be modeled by Use-Case model (Figure 34) which defines the main roles that use the particular functions. Two main roles are defined: Analyst which perform analysis and model development tasks, and Risk Managers which use both information about debtor, risk evaluation and forecasting results to make decisions; however, it may depend on the structure or specifics of credit organization. The functions that they perform are targeted primarily at model

creation and evaluation process as well as applying the results in practical activity and data management.



Source: created by the author

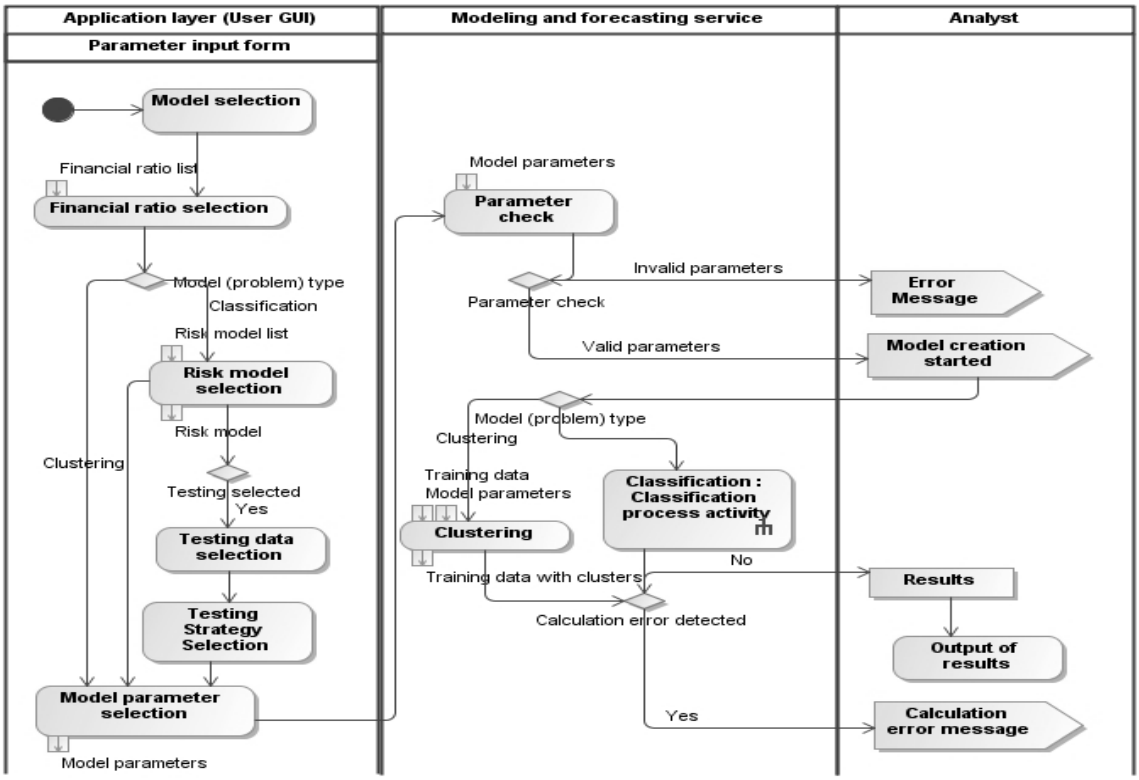
Figure 34. Standard Use-Case of credit risk DSS



Source: created by the author

Figure 35. Generalized classification activity diagram

Model creation task is defined as a complex process which includes such subtasks as model parameter selection, data preprocessing, training, testing, saving model to model repository; therefore it might be possible to automate some of these tasks using intelligent agents. Data analysis can be viewed as a generic process that can be extended to statistical, financial or visual analysis.



Source: created by the author.

Figure 36. Model development activity diagram

Yet analysis in credit risk is a complex task which needs all three ways of evaluation so this task can also be viewed as an aggregation of these three. Classification task is one of the main tasks in intelligent system development; activity digram describing this task is given in Figure 35. Activity diagram of intelligent model development process is given in Figure 36; it illustrates a case when classifier for forecasting, based on machine learning or statistical techniques, is developed using data in particular dimension (sector, period, etc.). Relevant tasks such as feature selection for relevant financial ratio selection, model training and testing are also included in this process. This model also includes classification problem formulation using output formation; this step is not needed if actual output data (bankruptcy or risk class labels) are available.

GDM-FS-CL$_{DA}$ ALGORITHM

This appendix describes a classifier based on group majority voting and discriminant analysis proposed in [61]. The main advantage of this technique is the possibility to reduce or totally eliminate dependency on single "expert" evaluation, therefore reducing subjectivity fact. Only algorithmic part is given in this appendix; for research results refer to [61]

B.1. Binary majority evaluation algorithm GDM-BME

The "expert" majority evaluation algorithm is based on majority voting technique used in ensemble classifiers, although it has some major changes. It is based on an idea in decision making that several subgroups of experts can be formed from a single set of experts which then can be propagated further, to the higher level (it can be viewed as managing level). The global decision in the higher level is made according to decisions of the subgroups. This is similar to decision making in organizations or groups of people. According to its nature (group decision making) and capabilities (binary majority), this algorithm is further referred as GDM-BME.

Algorithm GDM-BME

Input: m – number of "experts" (uncorrelated evaluators)

C – set representing possible class values ($C \in N$ and $C = N_0 \backslash N_c$, as we analyse only binary classification here, $C = \{0,1\}$)

M - predictions of experts with values from set C

M_j - prediction of j-th "expert" such that $M_j \in C$, $j = 1..m$.

if (m = 1)

$y = M_1$ *(we have single output, nothing to be done)*

else-if (m = 2)

if ($M_1 \neq M_2$)

$$y = \arg\max_{c \in M} \sum_{i:M_i=c} 1$$ *(simple majority selection)*

else

$y = rand(\sum_{i:M_i=c} 1)$ *(select value by random)*

else-if ($m = 2n-1$ and $n \geq 2$)

$$y = \arg\max_{c \in M} \sum_{i:M_i=c} 1$$ *(simple majority selection)*

else-if ($m = 2n$ and $n \geq 2$) {

k_0= size({i:M_i=c_0})

k_1= size({i:M_i=c_1})

if ($k_0 \neq k_1$)

$y = \arg\max_{c \in M} \sum_{i:M_i=c} 1$ *(simple majority selection)*

else {

Θ = {} *(init an empty set of "expert" groups)*

For k=1 to m do {

M′ = rem(M, k) *(remove k-th element from M)*

Θ = add(Θ, M′) *(add formed group to set of experts)*

}

(*remove one ensemble from set by random)*

Θ=remove(Θ, rand(1, m))

$y = \arg\max_{c \in \Theta} \sum_{i:\Theta_i=c, e \in \Theta} \arg\max_{c' \in e} \sum_{j:e_i=c'} 1$

}

}

Output: *y* - output value for instance D_i of dataset *D.*

Source: Danenas P., Garsva G. Simutis R. Development of Discriminant Analysis and Majority-Voting Based Credit Risk Assessment Classifier.

Algorithm 7. Binary majority evaluation algorithm

A more detailed explanation of algorithm for case when $m \in \{2n; \forall n \in N; \forall n \geq 2\}$ is as follows: if simple majority evaluation is not possible, we create *m* ensembles (groups of "experts") with *m-1 = 2n-1* members (such that we can apply simple majority voting principle) and randomly select *m-1 = 2n-1* evaluations from here such that expert would participate in this evaluation at least *m-1* times. Thus group majority voting evaluation is decomposed into a set of decisions by subgroups and the evaluation is obtained voting these decisions.

One of main drawbacks in this approach is decision which evaluation should be selected if *m = 2* and $M_1 = M_2$ as we have two different "equal" evaluations and no voting can be applied. Random selection was chosen to solve this problem; however, other options might be application of weights for each of "experts". If evaluators are other classifiers, it might be appropriate to select weights according to their accuracy or other evaluation metrics.

B.2. Classification technique

This section describes a method based on feature selection, machine learning technique for classification and discriminant analysis. It is very similar to technique described in Section 3.1.1; the main difference lies in evaluation step which is based on group decision making principle which is implemented in Algorithm 7. Therefore,

the modified technique is referred as GDM-FS-Cl_{DA} and described in Algorithm 8.

GDM-FS-Cl_{DA}

Input: Dataset D with a given set of companies CM, D^O – outputs of this dataset

A set of evaluators DA

C_V – set representing possible class values

A – the set of attributes

n_{att} – number of attributes (financial ratios)

l – the number of companies in the dataset

n_C – number of entries for particular company C

k – index of the company in the dataset

D_{Ck} - the subset of dataset D with size n for k-th company, $D = D_{C1} \cup D_{C2} \cup .. \cup D_{Cl,}$

1. Perform evaluation:

for $\forall D_i \in D$:

for $\forall da \in DA$

ev = evaluate(D_i, da), C_V); (Calculate evaluations and convert to bankruptcy classes)

if ev = {} (if instance cannot be evaluated)

(Exclude this "expert" evaluation from further evaluation marking it as N/A)

D = exclude(D_i, D)

else

D_i^{ev} = map(ev, C_V); (Add evaluation to the set of obtained evaluations)

D_i^O = GDM-BME(D_i^{ev}); (Deevelop a common solution and set it as the instance label)

2. Perform data imputation:

for $C \in CM$

for $i = 1,..., n_C$, $j = 1,...,n_{att}$

(i is the index of D_{Ck} instance, j is the index of financial atribute in instance i of D_{Ck})

(if the value is empty, average value for particular company is assigned)

if $D_{Ck}(x_{i,j}) = \{\}$

$$D_{Ck}(x_{i,j}) = \frac{\sum_{i=1}^{n} D_{Ck}(x_{i,j})}{n}$$

3. Divide companies to disjoint sets whose data will be used for training and testing

$C = C\_train \cup C\_test$, and $|C\_train| > |C\_test|$

4. Calculate training and testing data split percentage

5. Create disjoint sets as training and testing data by splitting data of selected companies in the sector by a percentage calculated in Step 5 ($C_D = C_D\_train \cup C_D\_test$ and $| C_D\_train| > | C_D\_test|$);

6. Apply feature selection procedure:

A' = select(A) (select attributes used in modeling)

7. Perform training, testing and evaluation procedures.

Output: a model (a list of support vectors and model parameters) that might be used to forecast

The list of selected attributes A' which forms this new model

Source: adapted from [61]

Algorithm 8. GDM-FS-Cl_{DA} algorithm

Appendix C

A FRAMEWORK FOR MULTIDIMENSIONAL ANALYSIS OF EVALUATION RESULTS[21]

As it is mentioned above, credit risk evaluation is a multidimensional task, therefore choise of business intelligence technologies for result analysis is a reasonable choice. The framework presented in this section is developed to combine various dimensions which can be useful in such research and apply business intelligence technologies such as data cube driven analysis. It includes such aspects as hierarchy of financial sectors, period od data, class formation type, various metrics for machine learning evaluation and etc.

SIC (Standard Industrial Classification) is industrial classification system used in USA. SIC can be defined as a structure of three levels consisting of four-digit code to identify particular industrial branch, codes from 01 to 99 which describe sector, and 01-09 codes for sector group identification. The main factor to select this classification system is the globality and availability of data which is also used in the system as well as its hierarchical structure which can be replaced by similar system used in other countries. The data comes from SEC (Securities Exchange Commission) EDGAR database and comprises USA companies with their financial statements.

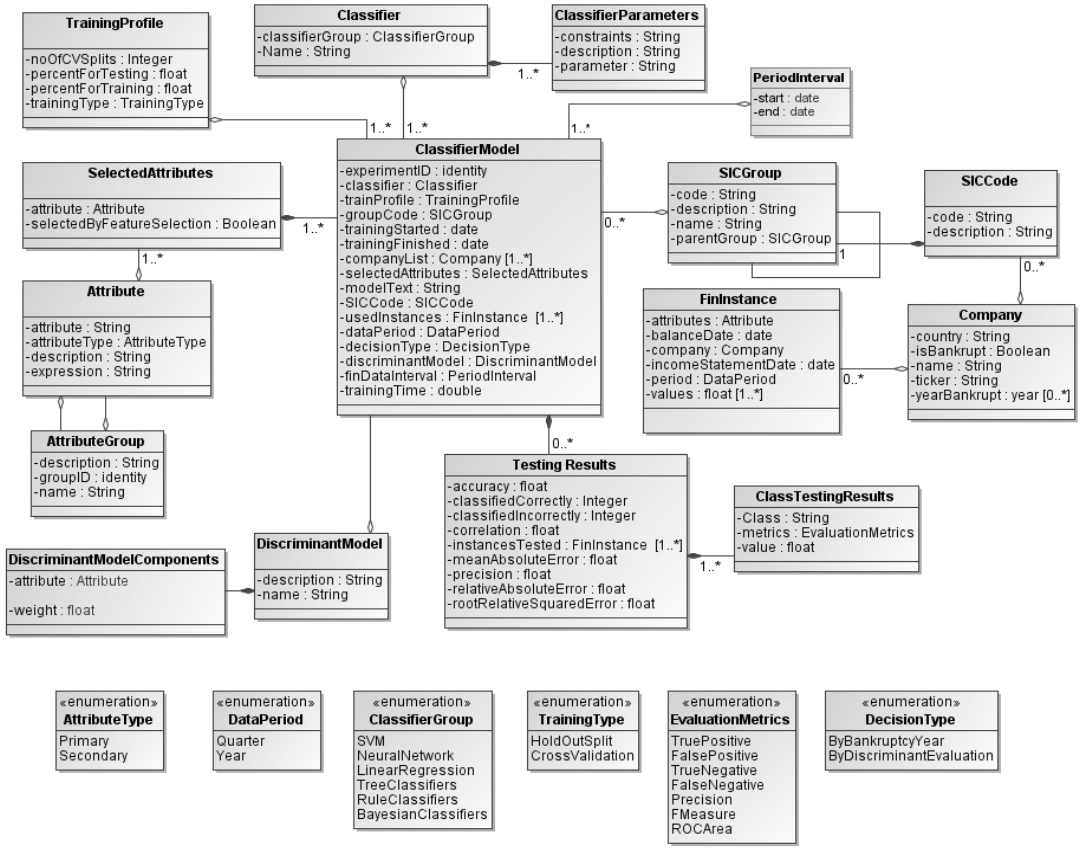
Entity-relationship model of this framework (here represented by class diagram) is given in Figure 37. It is composed of such components:

- classifier and its parameters;
- training profile (data splitting for training/testing or steps for cross validation procedure);
- periods (represented as intervals), attributes (financial ratios) and their groups (e.g., liquidity, fixed assets analysis);
- company data;
- financial data;
- SIC classification;
- components of external evaluators (based on discriminant analysis in this work);

[21] Danenas P., Garsva G. Daugiamatės analizės modelis kredito rizikos vertinimui, pagrįstam intelektiniais metodais. Informacinės technologijos (in Lithuanian),

- model and its metadata developed during model development process;
- testing results of this model (general and for classes separately). This level of detail is important as it is necessary to precisely identify bankrupt company or default possibility because loan given to debtor which is tend to default (Type I error) might result in larger loss than in the case if potential financially healthy client is identified incorrectly (Type II error).

Several components are described as enumerations such as period of financial statements (yearly/quarterly), attribute type (primary/derivative), groups of classification techniques, training type, decision type (whether classes are developed dynamically using discriminant or other techniques, or actual bankruptcy data is used) and metrics for testing results evaluation.



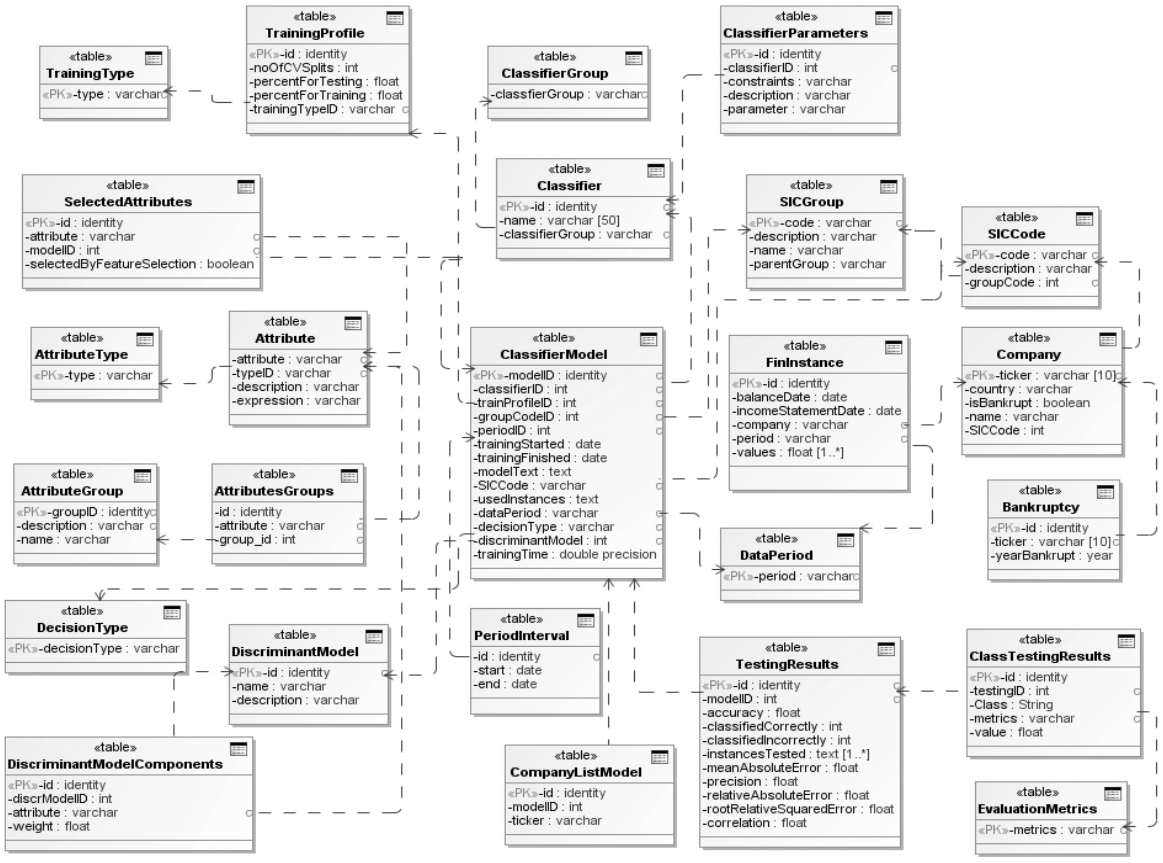
Source: Danenas P., Garsva G. Daugiamatės analizės modelis kredito rizikos vertinimui, pagrįstam intelektiniais metodais (in Lithuanian)

Figure 37. Class diagram for proposed framework

It might be necessary to transform the object model of the framework presented in Figure 37 to physical structure which can used to develop facility for storage of data described in this section and modeling results. Relational model is a widely known model used in relational DBMS which are widely used for storing

various data. Therefore, such transformation is also given in Figure 38. Such rules were observed while performing such transformation in order to avoid information and functionality loss:

- Enumerations were transformed to separate tables with single *varchar* type attribute. Corresponding attributes in structures containing fileds of this type were changed to *varchar* type attribute with external relation with table of corresponding enumeration.
- UML data types are mapped to corresponding SQL datatypes (i.e., *UML::String* to SQL *varchar* type);
- Attributes of Object type changed to attribute identifying corresponding object (usually of *identity* type, although other types might be possible in other cases, i.e., attribute identifying *Company* object is code *ticker* of *varchar* type) with external relation to corresponding table.



Source: Danenas P., Garsva G. Daugiamatės analizės modelis kredito rizikos vertinimui, pagrįstam intelektiniais metodais (in Lithuanian)

Figure 38. Transformation of the object model to relational model

The structure obtained after this transformation can be described as hybrid

(having both “start” and “snowflake” features) multidimensional scheme for cube development. This model can be useful while trying to perform comparative and generalized analysis, to present and view results in different views, in more detailed and decomposing way, and, according to results obtained, formulate conclusions. Table 33 shows more interesting views, when the number of dimensions is not higher than 5. Another important and interesting component of this framework are attributes selected by feature selection; proposed structure enables identification and analysis of the most critical and important factors in various dimensions, such as different periods or different sectors, and obtain results by performing generalization.

Table 33. Main views in developed framework

No.	View (dimensions)	Description
1	<Classifier, SIC code, Date interval, Decision type, Results>	Analysis of classification model performance by particular metrics of particular industrial branch (defined using SIC code, but also possibly higher dimension of SIC group), i.e., analysis of general or class level accuracy change in different periods or time intervals. This view enables analysis and comparison of forecast/evaluation and actual facts based model results.
2	<Classifier, SIC code Period, Date interval, Results>	Extension of view No. 1 by period dimension (with a goal to identify periods with better classification results).
3	<Classifier group, SIC code, Classifier, Date interval, Results>	Extension of view No. 1 by classifier group dimension. It enables to group classifier results by their groups (i.e., SVM, ANN or Bayesian).
4	<Classifier, SIC code, Discrimiannt technique, Date interval, Results>	This view enables analysis and comparison of results obtained by classifiers based on discriminant analysis driven evaluations thus trying to identify which is more suitable for final classifier development. It is used when actual bankruptcy data is not used.
5	<Classifier, SIC code, Training profile, Date interval, Results>	This view enables analysis and of results according to training profile. It can be useful while trying to identify the best training/testing split ratio or most effective cross validation strategy as well as classifier which can be trained using shortest time.

Source: Danenas P., Garsva G. Daugiamatės analizės modelis kredito rizikos vertinimui, pagrįstam intelektiniais metodais (in Lithuanian)

Appendix D

SVM PACKAGES AND IMPLEMENTATIONS

Name	Programming language	Implemented algorithms	Kernel functions	Authors	Flexibility, choice of parameters	Notes	Source
LibSVM	C, Python, JAVA, Ruby, MATLAB,	Classification; C-SVC, nu-SVC Regression: epsilon-SVR, nu-SVR One-class SVM for outlier detection	Linear, polynomial, sigmoid, RBF	Chang, Lin	General parameters	Many enhancements and tools for visualization	[36]
BSVM	C/C++	Classification, regression, multiclass classification using simple and Crammer-Singer formulations	Linear, polynomial, sigmoid, RBF	Chih-Wei Hsu, Chih-Jen Lin	General parameters	Based on LibSVM, includes multiclass classification	[29]
UniverSVM	C/C++	Semisupervised learning possibility, large scale transduction via CCCP optimization, sparse solutions via CCCP optimization and data-dependent regularization	Linear, polynomial, sigmoid, RBF	Fabian Sinz	General and specific parameters for tuning optimization and decision parameters	As stated by the author, it can efficiently perform with tens of thousands examples	[220]
mySVM	C/C++, JAVA	Classification, regression	Linear, polynomial, sigmoid, RBF, neural (2 layers), ANOVA, sum or product of user functions	Stefan Rüping	Parameters for performance estimation, optimization and serach of complexity parameter	SVM^{Light} optimization	[184]
SVMdark	C++	Classification, regression, ranking	Linear, polynomial, sigmoid, RBF	Martin Sewell	General parameters	Based on SVM^{Light}	[205]
TinySVM	C++, Perl, Python, JAVA	Classification, regression	Linear, polynomial, sigmoid, RBF, ANOVA		General parameters	Optimization twice faster than SVM^{Light}, fast processing of hundreds of thousands attributes	[211]
HeroSVM 2.1[22]	C++	Classification	Linear, polynomial, RBF	CENPARMI	General parameters	Faster and less memory demanding than LibSVM or SVM^{Light}	[105]
M-SVM	C	Multiclass classification	-	Yann	General parameters	Not much information	[94]

[22] The package was unavailable to download, therefore information, available on the website, is used

Name	Programming language	Implemented algorithms	Kernel functions	Authors	Flexibility, choice of parameters	Notes	Source
				Guermeur		of the package	
SimpleMKL	MATLAB	MKL binary classification SVM, MKL SVM regression, One-against-all and One-against-One MKL SVM	N/A[23]	N/A[7]	N/A[7]		[198]
SVM^{Light}	C	Classification, regression, ranking	Linear, polynomial, sigmoid, RBF, user defined	Thorsten Joachims	Very flexible – user can set many optimization and performance parameters	Fast optimization, effective work with large amounts of data	[120]
SVM^{Multiclass}	C	Classification, regression, ranking	Linear, polynomial, sigmoid, RBF, user defined	Thorsten Joachims	Very flexible – user can set many optimization and performance parameters	SVM^{Light} multiclass classification extension	[121]
SVMlin	C/C++	Linearly regularized least squares classification Semisupervised classification - Multi-switch linear transductive L2-SVMs Deterministic Annealing (DA) for Semi-supervised Linear L2-SVMs Modified finite Newton linear L2-SVM	N/A	Vikas Sindhwani	General parameters		[199]
CVM (Core Vector Machines)	C/C++	Classification; C-SVC, nu-SVC, one-class SVM, CVM, CVM-LS, BVM (Ball Vector Machine) Regression: epsilon-SVR, nu-SVR, CVR (Core Vector Regression) CVDD (Core Vector Data Description for novelty detection)	Linear, polynomial, sigmoid, RBF, Laplace, precomputed kernel, normalized polynomial, inverted distance, inverted square distance	Ivor W. Tsang, Andras Kocsor James T.Kwok	General parameters	Based on LibSVM (can be viewed as its extension)	[51] [216] [216]
PSVM (Potential - Support Vector Machine)	C/C++	Classification Regression Ranking Feature selection	Linear, polynomial, sigmoid, RBF, user defined	Knebel, Hochreiter	General parameters	Included tools for optimization, parameter selection and visual dta representation	[132]
SVM^{map}	Python	SVM algorithm, targeted at optimizing mean average precision (MAP)	N/A	Yisong Yue Thomas Finley	General parameters	Targeted at document ranking	[254]
SVM^{div}	Python	SVM algorithm for predicting diverse subsets (of documents)	N/A	Yisong Yue	General parameters	Targeted at document classification	[255]

[23] Information is unavailable

Name	Programming language	Implemented algorithms	Kernel functions	Authors	Flexibility, choice of parameters	Notes	Source
GPDT	C/C++	Binary classification using problem decomposition technique QP problem into smaller QP subproblems, each one being solved by a suitable gradient projection method (GPM). The currently implemented GPMs are the Generalized Variable Projection Method (GVPM) and the Dai-Fletcher method (DFGPM)	Linear, polynomial, RBF	T. Serafini, L. Zanni, G. Zanghirati	Flexible – algorithm and its parameters' selection		[258]
LIBLINEAR	C/C++, MATLAB, Octave, JAVA	L2-regularized logistic regression (primal) L2-regularized L2-loss support vector classification (dual) L2-regularized L2-loss support vector classification (primal) L2-regularized L1-loss support vector classification (dual) multi-class support vector classification by Crammer and Singer L1-regularized L2-loss support vector classification L1-regularized logistic regression L2-regularized logistic regression (dual)	None	R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin	General parameters	Based on LibSVM, suitable for large-scale learning	[80]
Least Squares Support Vector Machines (LS-SVM)	MATLAB	Multiclass classification, regression, clustering	Linear, polynomial, RBF, multilayer perceptron	J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle	General parameters	Tools for automatic selection of model parameters	[198] [66]

Source: created by the author, using various sources.

Appendix E

ADVANTAGES AND DISADVANTAGES OF VARIOUS COMPUTATIONAL INTELLIGENCE PARADIGMS

	Main idea	**Advantages**	**Disadvantages**
NN	Learn from examples using several constructs and algorithms just like a human being learns new things	• non-parametric method, which does not need any primary values of distribution or data mapping; • suitable for work with incomplete, missing or noisy data; • flexibility – mapping any complex nonlinearity or approximate function of any difficulty; • parallelization; • solid basis of research; • better classification results than statistical techniques; • possibility to solve problems without solutions known (Kohonen maps for clustering) • good performance at function approximation, forecasting, classification, clustering and optimization tasks	• complexity and vagueness ("black box") – there are no possibilities to evaluate the importance of variables which do not correlate and generate a set of rules to describe model operation; • problem of rule extraction; • sophisticated selection of model architecture; • overfitting and overtraining; • various aspects of model selection
EC	Mimics Darwinian principles of evolution to solve highly nonlinear, non-convex global optimization problems	• ability to store unfinished and non-optimal solutions; • ability to solve nonlinear, nonflexible problems; • conceptual simplicity; • wide application possibilities; • better performance than classical methods; • ability to use knowledge of the problem solved; • ability of parallel computing; • flexibility for dynamic changes; • self-optimization feature; • ability to solve problems without any known solutions	• Takes a long time to converge; • May not yield global optimal solution always unless it is augmented by a suitable direct search method • Rarely used separately, usually together with other techniques – additional time resources needed when creating model
FL	Models imprecision and ambiguity in the data using fuzzy sets and incorporates the human experiential knowledge into the model	• expert knowledge integration; • solid statistical and logic basis; • good at deriving human comprehensible fuzzy 'if–then' rules; • low computational requirements	• arbitrary choice of Membership function skews the results; • the problem of selection of membership function shapes, connectives for fuzzy sets and defuzzification operators;
SVM	Based on statistical learning theory to perform classification and regression tasks	• Yields global optimal solution as the problem is converted to a quadratic programming problem - no local minimums; • Optimal and wide distribution of	• Difficult to select kernel and its parameters properly; • Very slow in test phase; • High algorithmic complexity - creation of SVM-based hybrid

	Main idea	Advantages	Disadvantages
		solutions; • Control of space by optimizing bound parameter • Four main problems – training and testing efficiency, overtraining and architecture selection – are avoided • Good performance results; • Many variations and algorithms; • Solid basis of researches made; • Parallel computing ability	models is more complex than models based on other techniques; • Extensive memory usage; • No professional software or toolbox for work with various SVM methods
CBR	Learns from examples using the euclidean distance and k-nearest neighbor method	• Good for small data sets and when the data appears as cases; • similar to human like decision-making	• Not suitable to large data sets; • Poor in generalization
RS	Use lower and upper approximation to model uncertainty in the data	• Yield 'if–then' rules involving ordinal values to perform classification tasks	• Sometimes impractical to apply as it may lead to an empty set; • Sensitive to changes in data • Lack of accuracy
DT	Use of recursive partitioning technique and measures like entropy	• Some of them (e.g., CART) solve both classification and regression problems • Yield human comprehensible binary 'if–then' rules	• Overfitting problem; • Need a lot of data samples for reliable predictions; • Many of them can solve only classification problems
Bayes	Use of statistical probabilities	• Estimated probability reevaluation instead of complete elimination in case of inconsistency; • Prior knowledge can be combined with observed data to determine the final probability of a hypothesis; • Use of hypotheses that make probabilistic predictions • New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities. • Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making for measuring other methods	• Initial knowledge of many probabilities requirement. When these probabilities are not known in advance they are often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions. • Significant computational cost required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses).
AR	Discovery of rules using different measures of interestingness	• Yield human comprehensible binary 'if–then' rules • Can be used as a supervised and as an unsupervised technique • Are particularly well suited to finding local patterns in the data.	• Local patterns are not always suitable as global patterns

Appendix F

SUMMARY OF PREVIOUS SVM RESEARCH IN CREDIT RISK AND BANKRUPTCY DOMAIN

Authors	Techniques compared	Best performed	Accuracy	No of classes	No of ratios
Ahn et al. [3]	COSVM, KPSVM (SVM-GA), FSSVM, ISSVM (Instance Selection), SOSVM	SOSVM	79.68	2	39
Chaudhuri et al. [39]	PNN, Fuzzy SVM	Fuzzy SVM	94	2	14
Chen et al. [42]	PSO-SVM, Grid-SVM, SOM- SVM, SVM, SOM	PSO-SVM	92.19	2	13
Chen [44]	LDA, LR, C5.0, CART, SOM, LVQ, SVM, GA–SVM, PSO–SVM	PSO–SVM	varies	2	8
Chen et al. [45]	SVM, Logit	SVM	70.04	2	21
Chen et al. [46]	BPNN, SVM	SVM	87.5	2	6
Chen et al. [47]	SVM (LibSVM), ANN, LR	SVM	84.62	4	72
Chong et al.[49]	SVM, Neurofuzzy, fuzzy-integral SVM, majority-voting SVM	fuzzy-integral SVM	87.09	5	8
Fan et al. [81]	MDA, ANN, LVQ, SVM	SVM	70.97	2	11
Gao et al. [88]	SVM, KNN-SVM	KNN-SVM	92.5	2	5
Ghodselahi [90]	DA, LR, DT, RBFN, SVM, Bagging DT, Bagging NN, Bagging SVM, MLP, Boosting DT, Boosting NN, Boosting SVM, stacking, ensemble SVM	ensemble SVM	81.42	2	20
Hao et al. [100]	SVM, Fuzzy SVM, B-FSVM (Bilateral-weight fuzzy SVM), VS-FSVM (Vague Sets Fuzzy SVM)	VS-FSVM	86.70	2	16
Hao et al. [101]	SVM, Fuzzy SVM, Fuzzy SVM with fuzzy hyperplane (Fuzzy SVM with FH)	Fuzzy SVM with FH	86.05	2	N/A
Hu et al. [110]	3-layer LS-SVM, ANN	3-layer LS-SVM	92.5	4	N/A
Huang et al. [112]	Logit, NN, SVM	NN	79.81	5	21
Yang et al. [249]	FDA, LA, BPNN, PNN, RHPNN, SVM	SVM	96.2	2	18
Yang et al. [251]	LVQ, PLS-SVM	PLS-SVM	79	2	30
Yoon et al. [252]	SVM, ANN, C5.0, CART, MDA, LR	SVM	74.2	2	12
Yun et al. [256]	Grid Search+SVM, GA+SVM, PSO+SVM, DT+SVM, LDA+SVM, RST+SVM	PSO-SVM	87.10 78.70	2 2	6 14
Jiang et al.[118]	ANN, PSO-SVM	PSO-SVM	94.14	2	10
Kim, Ahn [130]	MDA, MLOGIT, CBR, ANN, Conventional MSVM (OAA, OAO, DAGSVM ECOC, Weston and Watkins. Crammer and Singer), OMSVM	OMSVM	67.98	5	14
Kotsiantis et al.[134]	CitationKNN, DD, MIEMDD, MISMO, MIBoost-DS, MISVM, MIRIPPER, KPSVM, FSSVM, ISSVM, SOSVM	MIBoost-DS MIRIPPER SOSVM	91.8 86.8 81.94	2	23
Kou et al. [136]	LDA, See5, SVM^{Light}, LibSVM, MCCQP	LibSVM See5 MCCQP	94.00 86.52 72.30	2 2 2	24 15 13

Authors	Techniques compared	Best performed	Accuracy	No of classes	No of ratios
Lai et al. [139]	Logit,ANN and SVM ensemble using majority voting and Bayesian rules	SVM using Bayesian	87.06	N/A	N/A
Lai et al. [141]	DNT (RBF), GA+SVM, MOE, LVQ, SVM+GS, SVM+GS+F-Score, FAR, LS-SVM+DOE	LS-SVM+DOE	77.96	2	24
Li et al. [145]	LogR, SVM, RVM, RVM+Ada, RVM_{ideal}	RVM_{ideal}	95.5 88.0	2 2	14 15
Liu et al. [147]	SVM, GA-SVM, gr-GA-SVM	gr-GA-SVM	86.84 75.50	2 2	14 24
Lo et al. [148]	SVM	SVM	71.8	2	8
Lv et al. [150]	RS-SVM, BPNN	RS-SVM	72.5	2	10
Min et al. [164]	SVM, BPNN, MDA, Logit	SVM	83.07	2	50
Min [165]	BPNN, SVM, PCA-FSVM	PCA-FSVM	97.55	N/A	N/A
Ping et al. [173]	LDA, LogR, ANN, RS-SVM	RS-SVM	87.52 76.60	2 2	14 24
Ravikumar et al. [179]	ANFIS (1), LIBSVM (2), Linear RBF (3), semi-online RBF1 (4), semi-online RBF2 (5), Orthogonal RBF (6), MLP (7), their various ensemble combinations	1357 2357	94.2 90	2 2	5 9
Ribeiro et al. [181]	S-Isomap, KNN, SVM, RVM	SVM	varies	N/A	30
Ribeiro et al. [182]	SVM, RVM, MLP, HLVQ	RVM	90.18	2	21
Van Gestel et al.[223]	LDA, Logit, LS-SVM, Bayesian LS-SVM	LS-SVM, BayLS-SVM	88.39	2	40
Vieira et al. [230]	Logistic, MLP, SVM, AdaboostM1, HLVQ-C	AdaboostM1	84.1	2	18
Wang et al. [231]	SVM, BPNN, C4.5, RS-SVM	RS-SVM	88.20	2	21
Wang et al. [232]	RS_RP_G_SVM, RS_G_SVM, RS_RP_S_SVM, RS_S_SVM, RS_C4.5, RS_RP_P_SVM, RS_P_SVM, Scoring method, RS_RP_L_SVM, RS_L_SVM	RS_G_SVM	85.5	2	23
Wang, Ma [233]	LRA. DT. ANN. Linear and poly SVM, Bagging SVM (linear & poly), RandomSubspace SVM (linear & poly), Boosting SVM (linear & poly), RSB-SVM (linear & poly)	RSB-SVM (polynomial)	78.98	2	18
Wang [234]	SVM+FS, PSO-SVM	PSO-SVM	90.30	2	52
Wei et al. [235]	MCLP, MCNP, DT, ANN, SVM-MK	SVM-MK	76.78	2	65
Wu et al. [240]	MDA, Logit, Probit, NN, SVM, GA-SVM	SVM, GA-SVM	varies	varies	varies
Zhang et. al [259]	GA+SVM, SVM, BPNN, GP, LR	GA+SVM	89.40	2	17
Zhou et al. [261]	MDA, BPNN, SVM, GA+SVM	GA+SVM	92.4	2	12
Zhou et al. [262]	RS, MDA, BPNN, SVM, Fuzzy SVM, GA-SVM	RS GA-SVM	95.51 91.63	2	12
Zhou et al. [263]	Fisher, Probit, ANN, SVM, ACO-SVM	ACO-SVM	75.57	2	12
Zhou , Lai [264]	$BPNN_{Sigmoid}$, $BPNN_{Linear}$, DTC4.5, KNN50, Adaboost, WLSSVM	WLSSVM	93.19 79.71	2 2	14 24
Zhou, Lai [265]	LDA, QDA, LR, DT, wSVM, UVe, PWBVSe, PWBTSe	PWBTSe	78.13	2	24
Zhou et al. [266]	SVM, Fuzzy SVM	Fuzzy SVM	81.43	3	12

Source: created by the author, using various sources

SVM research for credit risk evaluation by Yu et al.

Used techniques	**Best performed**	**Accuracy**	**No of classes**	**No of ratios**
LDA, QDA, LogR, DT, k-NN, DS_{lssvm} (LS-SVM with Direct Search)	DS_{lssvm}	77.10 86.96	2 2	24 14
LogR, ANN, SVM, RS, Hybrid (SVM+RS+FS)	Hybrid technique	90.15	2	12
LinR, LogR, ANN, SVM, LSSVM, FSVM, LS-FSVM	LS-FSVM	89.21	2	14
LinR, LogR, ANN, SVM (Lin, Poly, RBF), U-FSVM and B-FSVM with various kernels and membership functions	B-FSVM with Poly/RBF kernel and Logit regress. Membership	83.94 79.00 66.17	2 2 2	12 12 12
LDA, BPNN, Standard SVM, $LSSVM_{poly}$, $LSSVM_{rbf}$, $LSSVM_{sig}$, $LSSVM_{mix}$, LSSVM+GA+FS, LSSVM+GA, Evolving LSSVM	Evolving LSSVM	79.49 72.89 77.32	2 2 2	12 14 20
Ensemble techniques				
LinR, LogR, ANN, Fuzzy SVM, ensembles based on SVM and ANN: voting based (majority) and reliability-based (max, min, median, mean, product)	SVM ensemble with product rule	88.42 86.12	2 2	13 12
LinR, LogR, ANN, SVM, ensembles: majority voting, ANN , SVM with and without PCA based metamodels	SVM and PCA based metamodel	89.76	2	13
Ensembles: MDA+LogR, MDA+ANN, MDA+SVM, LogR+ANN, LogR+SVM, ANN+SVM, MDA+LogR+ANN, MDA+LogR+SVM, MDA+ANN+SVM, LogR+ANN+SVM, MDA+LogR+ANN+SVM using majority voting and evolutionary programing	LogR+ANN+SVM MDA+LogR+ANN+SVM	88.09 85.35	2	13
LinR, LogR, BPNN, RFNN, SVM, ensembles: BPNN, RFNN, SVM, Majority GDM, Fuzzy GDM	Fuzzy GDM	80.14 86.17 82.00	2 2 2	14 13 20

Source: adopted from Yu L., Lai K. K., Wang S., Zhou L. Bio-Inspired Credit Risk Analysis.

Appendix G

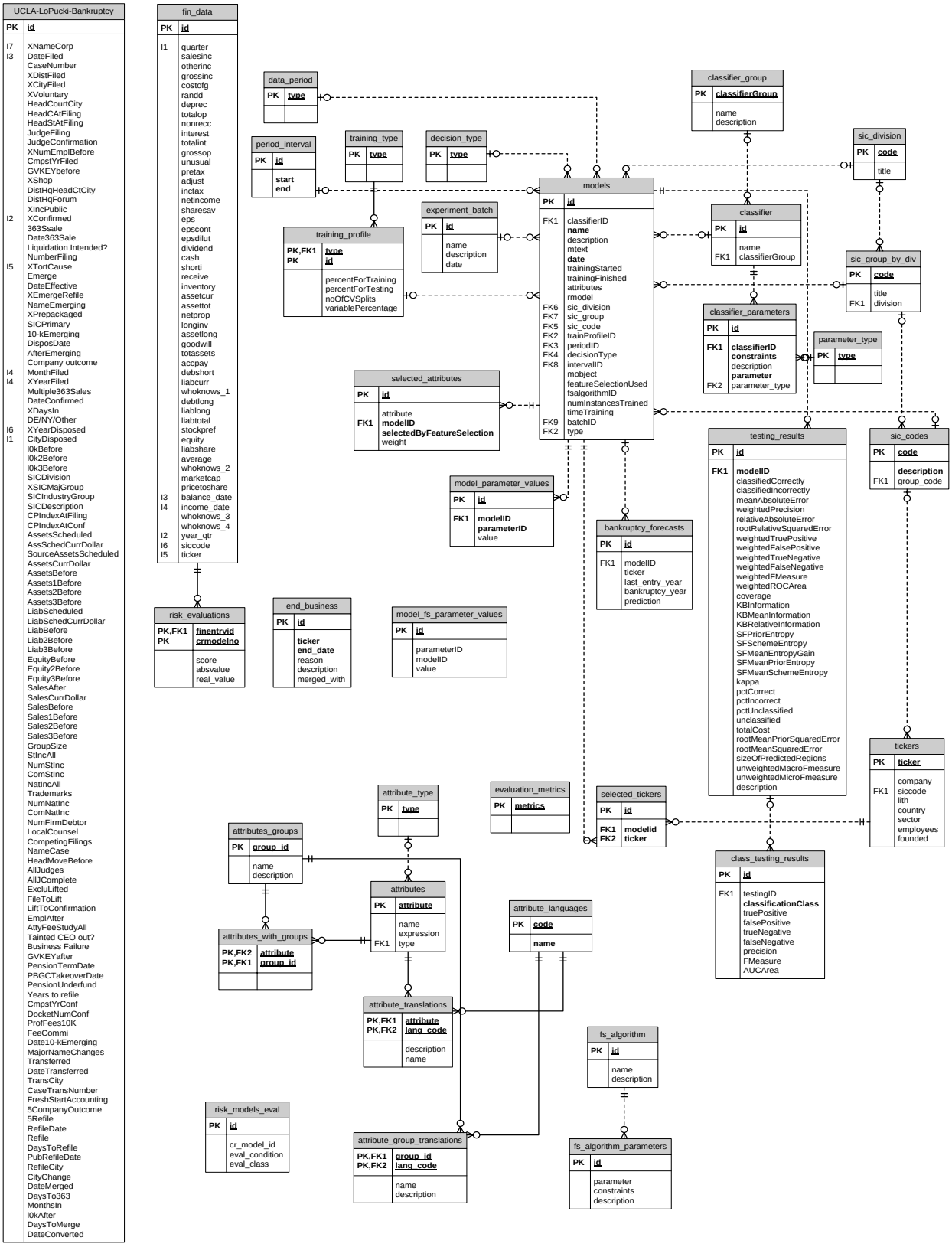
TYPES OF RISKS RELATED TO INSOLVENCY AND TECHNIQUES FOR THEIR EVALUATION

Risk	Quantitative	Analytical	Methods for risk evaluation and reduction
Market			
Currency	☑		Technical analysis, forecasting techniques value at risk (VaR) techniques (VaR, Markowitz), modern statistical, intelligent and econometric models, stress testing, Monte Carlo analysis, sensitivity analysis, dynamic strategies, portfolio analysis, Incremental Risk Charge (IRC), profit and loss attribution, variance / covariance estimation
Commodity	☑		
Interest rate	☑		
Reappraisal	☑		
Yield difference	☑		
Stock trading	☑		
Option	☑		
Capital	☑		
Operational			
Internal cheating		☑	Monitoring of company's internal operations and their analysis of legality and company policy
External cheating		☑	Monitoring of company's external operations and their analysis of legality and company policy
Safety		☑	Analysis and monitoring of safety violations and their causes
Working environment		☑	Analysis of assurance of internal work safety policy, history of incidents in the workplace
Personnel relationships		☑	Analysis of inner conflicts and working atmosphere, reviews of the company
Damage for fixed assets	☑	☑	Analysis of facts in financial reports, related to fixed assets
Bad business practices		☑	Analysis of business plan, activities, orders history
Inquality production		☑	Analysis of client reviews, production data, quality analysis
Errors in business management systems	☑	☑	Analysis of business management software errors and financial loss originating from these errors
Financial transaction execution	☑	☑	Analysis of financial transactions' historical data
Management of business processes	☑	☑	Analysis of business process monitoring and quality data
Orders' execution	☑	☑	Orders history analysis
New products	☑	☑	Market research with evaluation of possible demand, future income and expenses, profitability of products and/or services
Technology	☑	☑	Analysis of errors or problems arised in activity and their numbers, consolidated reports, effectiveness analysis
Client solvency	☑	☑	Various financial, mathematical, statistical, analytical, expert and intelligent techniques
Logistics	☑	☑	Analysis of problems of unfulfilled orders related to transportation
Business environment			
Legal environment		☑	Expert, analytical (analysis of related enactments)
Economical risk		☑	Expert, analytical (analysis of economical policy and money policy executed by central bank)
Competition	☑	☑	Market research and analysis
Reputation risk		☑	Analysis of client reviews, documents and related data
Country risk	☑		Special systems of ratios; monitoring and analysis of country's economical, political and social situation and ratings
Business strategies		☑	Analysis of business plan and projected startegies
Liquidity risk		☑	Internal institution analysis
Credit risk	☑	☑	Various financial, mathematical, statistical, analytical, expert and intelligent techniques, credit value adjustment, potential future exposure, stress testing

Source: created by the author, using various source

Appendix H

DATABASE STRUCTURE OF THE IMPLEMENTED PROTOTYPE



Appendix I

MAIN CHARACTERISTICS OF DATASETS USED IN EXPERIMENTS

This appendix describes main characteristics for each dataset used in experiments. The tables present number of instances and intances which belong to particular classes.

SIC code	Title
01-09	Agriculture, Forestry, And Fishing
10-14	Mining
15-17	Construction
20-39	Manufacturing
40-49	Transportation, Communications, Electric, Gas, And Sanitary Services
50-51	Wholesale Trade
52-59	Retail Trade
60-67	Finance, Insurance, And Real Estate
70-89	Services

Specifications of datasets based on incremental manner (used in conjunction with sliding window testing approach)

Altman evaluation based datasets

Sector	**01-09**	**10-14**	**15-17**	**20-39**	**40-49**	**50-51**	**52-59**	**60-67**	**70-89**
1999-2000	45 [6, 13, 26]	464 [113, 99, 252]	101 [7, 8, 86]	3501 [159, 204, 3138]	885 [223, 238, 424]	318 [16, 12, 290]	416 [12, 12, 392]	2084 [1638, 99, 347]	1785 [194, 168, 1423]
1999-2001	74 [11, 21, 42]	777 [199, 174, 404]	158 [13, 11, 134]	5773 [318, 377, 5078]	1426 [382, 374, 670]	515 [27, 21, 467]	699 [23, 18, 658]	3347 [2615, 161, 571]	2992 [393, 293, 2306]
1999-2002	104 [18, 30, 56]	1108 [288, 252, 568]	215 [16, 18, 181]	8091 [544, 582, 6965]	1975 [548, 526, 901]	710 [38, 29, 643]	988 [32, 26, 930]	4627 [3632, 230, 765]	4204 [607, 407, 3190]
1999-2003	136 [26, 36, 74]	1472 [360, 321, 791]	270 [20, 21, 229]	10444 [716, 722, 9006]	2547 [690, 683, 1174]	904 [52, 35, 817]	1278 [43, 30, 1205]	6012 [4669, 313, 1030]	5435 [767, 517, 4151]
1999-2004	166 [29, 44, 93]	1876 [427, 388, 1061]	335 [26, 23, 286]	12953 [882, 864, 11207]	3150 [808, 858, 1484]	1103 [64, 44, 995]	1567 [52, 39, 1476]	7507 [5747, 415, 1345]	6748 [910, 630, 5208]
1999-2005	196 [34, 47, 115]	2295 [488, 447, 1360]	398 [30, 30, 338]	15590 [1078, 1019, 13493]	3788 [937, 1035, 1816]	1312 [76, 50, 1186]	1895 [57, 50, 1788]	9101 [6896, 511, 1694]	8105 [1072, 734, 6299]
1999-2006	218 [37, 50, 131]	2646 [539, 518, 1589]	454 [35, 35, 384]	17700 [1206, 1129, 15365]	4288 [1023, 1181, 2084]	1474 [79, 55, 1340]	2192 [63, 57, 2072]	10457 [7815, 582, 2060]	9133 [1169, 815, 7149]
1999-2007	239 [40, 55, 144]	2986 [603, 583, 1800]	511 [40, 39, 432]	19675 [1333, 1242, 17100]	4778 [1111, 1321, 2346]	1624 [82, 59, 1483]	2436 [66, 62, 2308]	11804 [8732, 656, 2416]	10069 [1257, 897, 7915]

Springate evaluation based datasets

Sector	01-09	10-14	15-17	20-39	40-49	50-51	52-59	60-67	70-89
1999-2000	54 [3, 51]	666 [90, 576]	88 [2, 86]	4587 [332, 4255]	1202 [98, 1104]	408 [20, 388]	559 [8, 551]	351 [60, 291]	2507 [246, 2261]
1999-2001	85 [6, 79]	1041 [144, 897]	134 [5, 129]	7157 [527, 6630]	1830 [142, 1688]	630 [30, 600]	892 [14, 878]	541 [89, 452]	3848 [356, 3492]
1999-2002	118 [9, 109]	1416 [211, 1205]	176 [5, 171]	9779 [751, 9028]	2461 [188, 2273]	850 [41, 809]	1229 [16, 1213]	726 [119, 607]	5174 [483, 4691]
1999-2003	152 [12, 140]	1820 [278, 1542]	218 [5, 213]	12399 [1002, 11397]	3114 [234, 2880]	1060 [51, 1009]	1556 [24, 1532]	918 [161, 757]	6500 [616, 5884]
1999-2004	182 [17, 165]	2241 [348, 1893]	262 [6, 256]	15062 [1227, 13835]	3778 [275, 3503]	1270 [63, 1207]	1876 [30, 1846]	1108 [194, 914]	7836 [735, 7101]
1999-2005	212 [19, 193]	2666 [414, 2252]	304 [8, 296]	17773 [1437, 16336]	4442 [310, 4132]	1480 [75, 1405]	2223 [37, 2186]	1300 [237, 1063]	9171 [860, 8311]
1999-2006	234 [20, 214]	3014 [470, 2544]	342 [8, 334]	19888 [1564, 18324]	4943 [328, 4615]	1642 [80, 1562]	2526 [41, 2485]	1440 [264, 1176]	10174 [941, 9233]
1999-2007	255 [23, 232]	3351 [516, 2835]	381 [9, 372]	21859 [1676, 20183]	5428 [343, 5085]	1791 [82, 1709]	2770 [43, 2727]	1567 [283, 1284]	11081 [1007, 10074]

Zmijewski evaluation based datasets

Sector	01-09	10-14	15-17	20-39	40-49	50-51	52-59	60-67	70-89
1999-2000	43 [36, 7]	492 [343, 149]	111 [84, 27]	4307 [3181, 1126]	1157 [799, 358]	374 [258, 116]	529 [398, 131]	1750 [711, 1039]	2353 [1497, 856]
1999-2001	68 [57, 11]	784 [558, 226]	173 [130, 43]	6732 [4934, 1798]	1766 [1182, 584]	582 [404, 178]	847 [632, 215]	2659 [1074, 1585]	3636 [2229, 1407]
1999-2002	96 [78, 18]	1080 [770, 310]	230 [168, 62]	9218 [6643, 2575]	2381 [1558, 823]	789 [551, 238]	1172 [875, 297]	3621 [1462, 2159]	4924 [2985, 1939]
1999-2003	124 [101, 23]	1398 [1011, 387]	288 [217, 71]	11711 [8413, 3298]	3018 [1975, 1043]	984 [701, 283]	1490 [1118, 372]	4643 [1922, 2721]	6215 [3815, 2400]
1999-2004	150 [122, 28]	1731 [1271, 460]	351 [268, 83]	14268 [10333, 3935]	3672 [2435, 1237]	1181 [854, 327]	1798 [1364, 434]	5692 [2437, 3255]	7532 [4715, 2817]
1999-2005	176 [142, 34]	2077 [1550, 527]	414 [320, 94]	16892 [12297, 4595]	4323 [2913, 1410]	1383 [1010, 373]	2132 [1635, 497]	6777 [3005, 3772]	8854 [5641, 3213]
1999-2006	194 [157, 37]	2366 [1788, 578]	468 [368, 100]	18957 [13933, 5024]	4816 [3288, 1528]	1543 [1141, 402]	2427 [1883, 544]	7696 [3549, 4147]	9857 [6367, 3490]
1999-2007	213 [171, 42]	2660 [2030, 630]	523 [413, 110]	20895 [15478, 5417]	5295 [3655, 1640]	1691 [1263, 428]	2666 [2092, 574]	8599 [4073, 4526]	10774 [7030, 3744]

Specifications for datasets formed on basis for each period

The same characteristics for datasets which are generated on periodical basis, i.e., for each year.

Altman evaluation based datasets

Sector	01-09	10-14	15-17	20-39	40-49	50-51	52-59	60-67	70-89
1999	18 [3, 5, 10]	201 [52, 49, 100]	46 [3, 3, 40]	1359 [67, 81, 1211]	388 [86, 96, 206]	127 [8, 4, 115]	128 [2, 5, 121]	946 [754, 40, 152]	684 [60, 51, 573]
2000	27 [3, 8, 16]	263 [61, 50, 152]	55 [4, 5, 46]	2142 [92, 123, 1927]	497 [137, 142, 218]	191 [8, 8, 175]	288 [10, 7, 271]	1138 [884, 59, 195]	1101 [134, 117, 850]
2001	29 [5, 8, 16]	313 [86, 75, 152]	57 [6, 3, 48]	2272 [159, 173, 1940]	541 [159, 136, 246]	197 [11, 9, 177]	283 [11, 6, 266]	1263 [977, 62, 224]	1207 [199, 125, 883]
2002	30 [7, 9, 14]	331 [89, 78, 164]	57 [3, 7, 47]	2318 [226, 205, 1887]	549 [166, 152, 231]	195 [11, 8, 176]	289 [9, 8, 272]	1280 [1017, 69, 194]	1212 [214, 114, 884]
2003	32 [8, 6, 18]	364 [72, 69, 223]	55 [4, 3, 48]	2353 [172, 140, 2041]	572 [142, 157, 273]	194 [14, 6, 174]	290 [11, 4, 275]	1385 [1037, 83, 265]	1231 [160, 110, 961]
2004	30 [3, 8, 19]	404 [67, 67, 270]	65 [6, 2, 57]	2509 [166, 142, 2201]	603 [118, 175, 310]	199 [12, 9, 178]	289 [9, 9, 271]	1495 [1078, 102, 315]	1313 [143, 113, 1057]
2005	30 [5, 3, 22]	419 [61, 59, 299]	63 [4, 7, 52]	2637 [196, 155, 2286]	638 [129, 177, 332]	209 [12, 6, 191]	328 [5, 11, 312]	1594 [1149, 96, 349]	1357 [162, 104, 1091]
2006	22 [3, 3, 16]	351 [51, 71, 229]	56 [5, 5, 46]	2110 [128, 110, 1872]	500 [86, 146, 268]	162 [3, 5, 154]	297 [6, 7, 284]	1356 [919, 71, 366]	1028 [97, 81, 850]
2007	21 [3, 5, 13]	340 [64, 65, 211]	57 [5, 4, 48]	1975 [127, 113, 1735]	490 [88, 140, 262]	150 [3, 4, 143]	244 [3, 5, 236]	1347 [917, 74, 356]	936 [88, 82, 766]

Springate evaluation based datasets

Sector	01-09	10-14	15-17	20-39	40-49	50-51	52-59	60-67	70-89
1999	24 [1, 23]	305 [44, 261]	39 [2, 37]	1974 [160, 1814]	566 [45, 521]	180 [10, 170]	191 [4, 187]	156 [24, 132]	1113 [124, 989]
2000	30 [2, 28]	361 [46, 315]	49 [49]	2613 [172, 2441]	636 [53, 583]	228 [10, 218]	368 [4, 364]	195 [36, 159]	1394 [122, 1272]
2001	31 [3, 28]	375 [54, 321]	46 [3, 43]	2570 [195, 2375]	628 [44, 584]	222 [10, 212]	333 [6, 327]	190 [29, 161]	1341 [110, 1231]
2002	33 [3, 30]	375 [67, 308]	42 [42]	2622 [224, 2398]	631 [46, 585]	220 [11, 209]	337 [2, 335]	185 [30, 155]	1326 [127, 1199]
2003	34 [3, 31]	404 [67, 337]	42 [42]	2620 [251, 2369]	653 [46, 607]	210 [10, 200]	327 [8, 319]	192 [42, 150]	1326 [133, 1193]
2004	30 [5, 25]	421 [70, 351]	44 [1, 43]	2663 [225, 2438]	664 [41, 623]	210 [12, 198]	320 [6, 314]	190 [33, 157]	1336 [119, 1217]
2005	30 [2, 28]	425 [66, 359]	42 [2, 40]	2711 [210, 2501]	664 [35, 629]	210 [12, 198]	347 [7, 340]	192 [43, 149]	1335 [125, 1210]
2006	22 [1, 21]	348 [56, 292]	38 [38]	2115 [127, 1988]	501 [18, 483]	162 [5, 157]	303 [4, 299]	140 [27, 113]	1003 [81, 922]
2007	21 [3, 18]	337 [46, 291]	39 [1, 38]	1971 [112, 1859]	485 [15, 470]	149 [2, 147]	244 [2, 242]	127 [19, 108]	907 [66, 841]

Zmijewski evaluation based datasets

Sector	01-09	10-14	15-17	20-39	40-49	50-51	52-59	60-67	70-89
1999	19 [17, 2]	222 [152, 70]	49 [38, 11]	1849 [1312, 537]	542 [376, 166]	163 [112, 51]	178 [131, 47]	809 [345, 464]	1041 [655, 386]
2000	24 [19, 5]	270 [191, 79]	62 [46, 16]	2458 [1869, 589]	615 [423, 192]	211 [146, 65]	351 [267, 84]	941 [366, 575]	1312 [842, 470]
2001	25 [21, 4]	292 [215, 77]	62 [46, 16]	2425 [1753, 672]	609 [383, 226]	208 [146, 62]	318 [234, 84]	909 [363, 546]	1283 [732, 551]
2002	28 [21, 7]	296 [212, 84]	57 [38, 19]	2486 [1709, 777]	615 [376, 239]	207 [147, 60]	325 [243, 82]	962 [388, 574]	1288 [756, 532]
2003	28 [23, 5]	318 [241, 77]	58 [49, 9]	2493 [1770, 723]	637 [417, 220]	195 [150, 45]	318 [243, 75]	1022 [460, 562]	1291 [830, 461]
2004	26 [21, 5]	333 [260, 73]	63 [51, 12]	2557 [1920, 637]	654 [460, 194]	197 [153, 44]	308 [246, 62]	1049 [515, 534]	1317 [900, 417]
2005	26 [20, 6]	346 [279, 67]	63 [52, 11]	2624 [1964, 660]	651 [478, 173]	202 [156, 46]	334 [271, 63]	1085 [568, 517]	1322 [926, 396]
2006	18 [15, 3]	289 [238, 51]	54 [48, 6]	2065 [1636, 429]	493 [375, 118]	160 [131, 29]	295 [248, 47]	919 [544, 375]	1003 [726, 277]
2007	19 [14, 5]	294 [242, 52]	55 [45, 10]	1938 [1545, 393]	479 [367, 112]	148 [122, 26]	239 [209, 30]	903 [524, 379]	917 [663, 254]

Appendix J

SPECIFICATIONS OF GERMAN AND AUSTRALIAN DATASETS

J.1. German dataset

Two variations of German credit dataset are provided: the original dataset which contains categorical/symbolic attributes and one for algorithms that need numerical attributes (such as SVM), Strathclyde University produced the file "german.data-numeric". This file has been edited and several indicator variables added to make it suitable for algorithms which cannot cope with categorical (nominal) variables. Several attributes that are ordered categorical (such as attribute 17) have been coded as integer.

Number of attributes german: 20 (7 numerical, 13 categorical)
Number of attributes german.numer: 24 (24 numerical)
Number of instances: 1000

Attribute	**Type**	**Main characteristics**	**No of values**
Status of existing checking account salary assignments for at least 1 year	qualitative	A11: < 0 DM A12 : 0 <= ... < 200 DM A13 : >= 200 DM / A14 : no checking account	274 269 63 394
Duration in month	numerical	Min value: 4, max value: 72, mean: 20.903	
Credit history	qualitative	A30 : no credits taken/all credits paid back duly A31: all credits at this bank paid back duly A32 : existing credits paid back duly till now A33 : delay in paying off in the past A34 : critical account/other credits existing (not at this bank)	40 49 530 88 293
Purpose	qualitative	A40 : car (new) A41 : car (used) A42 : furniture/equipment A43 : radio/television A44 : domestic appliances A45 : repairs A46 : education A47 : (vacation - does not exist?) A48 : retraining A49 : business A410 : others	234 103 181 280 12 22 50 0 9 97 12
Credit amount	numerical	Min value: 250, max value: 18242, mean: 3271.258	
Savings account/bonds	qualitative	A61 : < 100 DM A62 : 100 <= ... < 500 DM A63 : 500 <= ... < 1000 DM A64 : >= 1000 DM A65 : unknown/ no savings account	603 103 63 48 183
Present employment since	qualitative	A71 : unemployed A72 : < 1 year A73 : 1 <= ... < 4 years A74 : 4 <= ... < 7 years A75 : >= 7 years	62 172 339 174 253
Installment rate in percentage of disposable income	numerical	Min value: 1 Max value: 4 Mean: 2.973	
Personal status and sex	qualitative	A91 : male: divorced/separated A92 : female: divorced/separated/married A93 : male: single A94 : male: married/widowed	50 310 548 92

Attribute	Type	Main characteristics	No of values
		A95 : female : single	0
Other debtors / guarantors	qualitative	A101 : none A102 : co-applicant A103 : guarantor	907 41 52
Present residence since	numerical	Min value: 1, max value: 4, mean: 2.845	
Property	qualitative	A121 : real estate A122 : if not A121 : building society savings agreement/ life insurance A123 : if not A121/A122 : car or other, not in Savings account/bonds A124 : unknown / no property	282 232 332 154
Age in years	numerical	Min value: 19, max value: 75, mean: 35.546	
Other installment plans	qualitative	A141 : bank A142 : stores A143 : none	139 47 814
Housing	qualitative	A151 : rent A152 : own A153 : for free	179 713 108
Number of existing credits at this bank	numerical	Min value: 1, max value: 4, mean: 1.407	
Job	qualitative	A171 : unemployed/ unskilled - non-resident A172 : unskilled - resident A173 : skilled employee /official A174 : management/self-employed/highly qualified employee/officer	22 200 630 148
Number of people being liable to provide maintenance for	numerical	Min value: 1 Max value: 2 Mean: 1.155	
Telephone	qualitative	A191 : none A192 : yes, registered under the customers name	596 404
Foreign worker	qualitative	A201 : yes A202 : no	963 37
Class	qualitative	1: 700 2: 300	

J.2. Australian credit approval dataset

This file concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. Numerical version of this dataset is also provided as well.

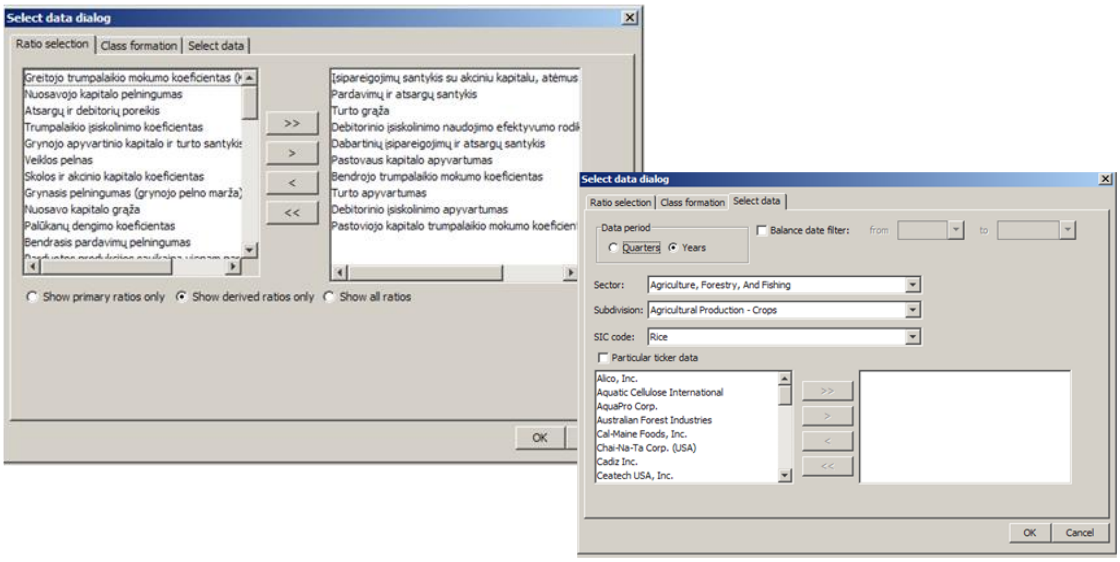
Number of attributes australian: 14 (6 numerical, 8 categorical)
Number of instances: 690

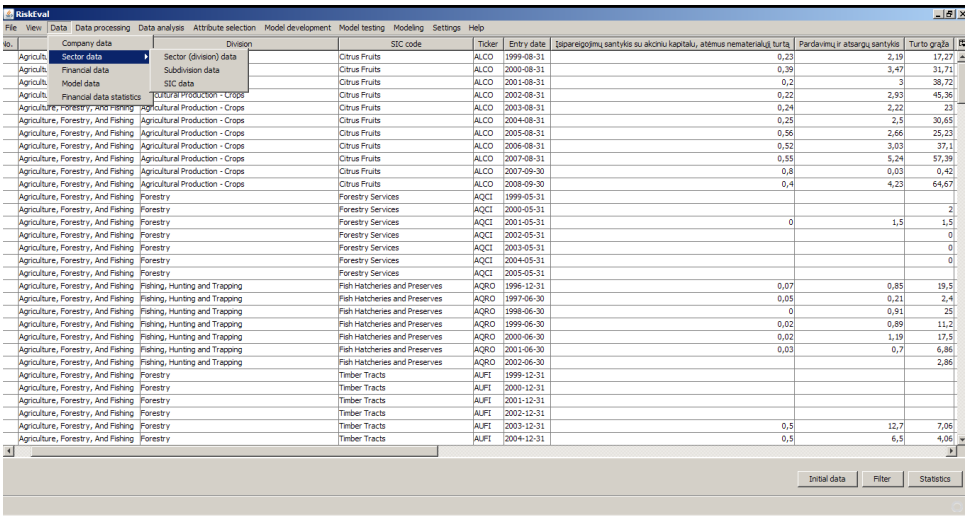
Attribute	Type	Main characteristics
A1	qualitative	a (222 instances), b (468 instances)
A2	numeric	Value range: [13.75;80.25] Mean: 31.568
A3	numeric	Value range: [0; 28] Mean: 4.759
A4	qualitative	p (163 instances), g (525 instances), gg (2 instances)
A5	qualitative	1 (53 instances), 2 (30 instances), 3 (59 instances), 4 (51 instance), 5 (10 instances), 6 (54 instances), 7 (38 instances), 8 (146 instances), 9 (64 instances), 10 (25 instances), 11 (78 instances), 12 (3 instances), 13 (41 instances), 14 (38 instances)
A6	qualitative	1 (57 instances), 2 (6 instances), 3 (8 instances), 4 (408 instances), 5 (59 instances), 6 (0 instances), 7 (6 instances), 8 (138 instances), 9 (8 instances)
A7	numeric	Value range: [0; 28.5] Mean: 2.223
A8	qualitative	0 (329 instances), 1 (361 instances)

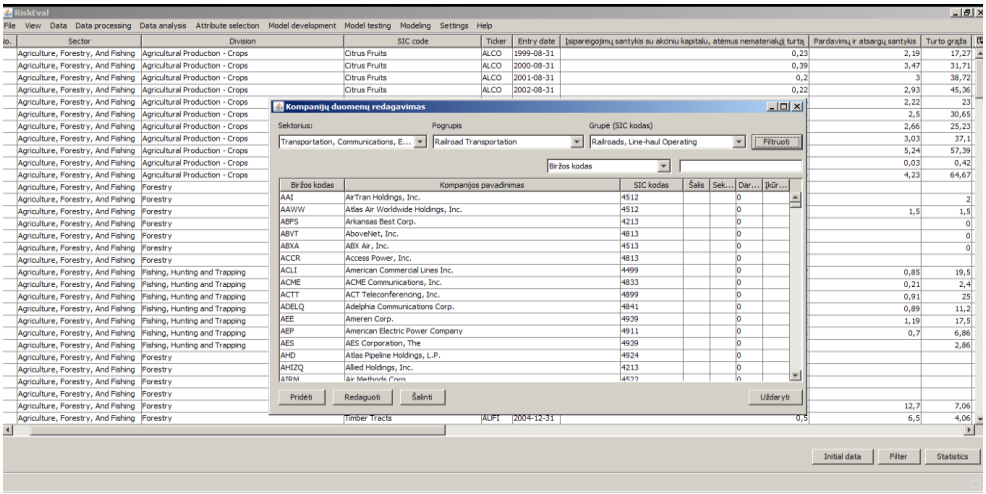
Attribute	Type	Main characteristics
A9	qualitative	0 (395 instances), 1 (295 instances)
A10	numeric	Value range: [0; 67] Mean: 2.4
A11	qualitative	0 (374 instances), 1 (316 instances)
A12	qualitative	1 (57 instances), 2 (625 instances) , 3 (8 instances)
A13	numeric	Value range: [0; 2000] Mean: 2.4
A14	numeric	Value range: [0; 100001] Mean: 1018.386
A15 (class)	qualitative	0 (383 instances) 1 (307 instances)

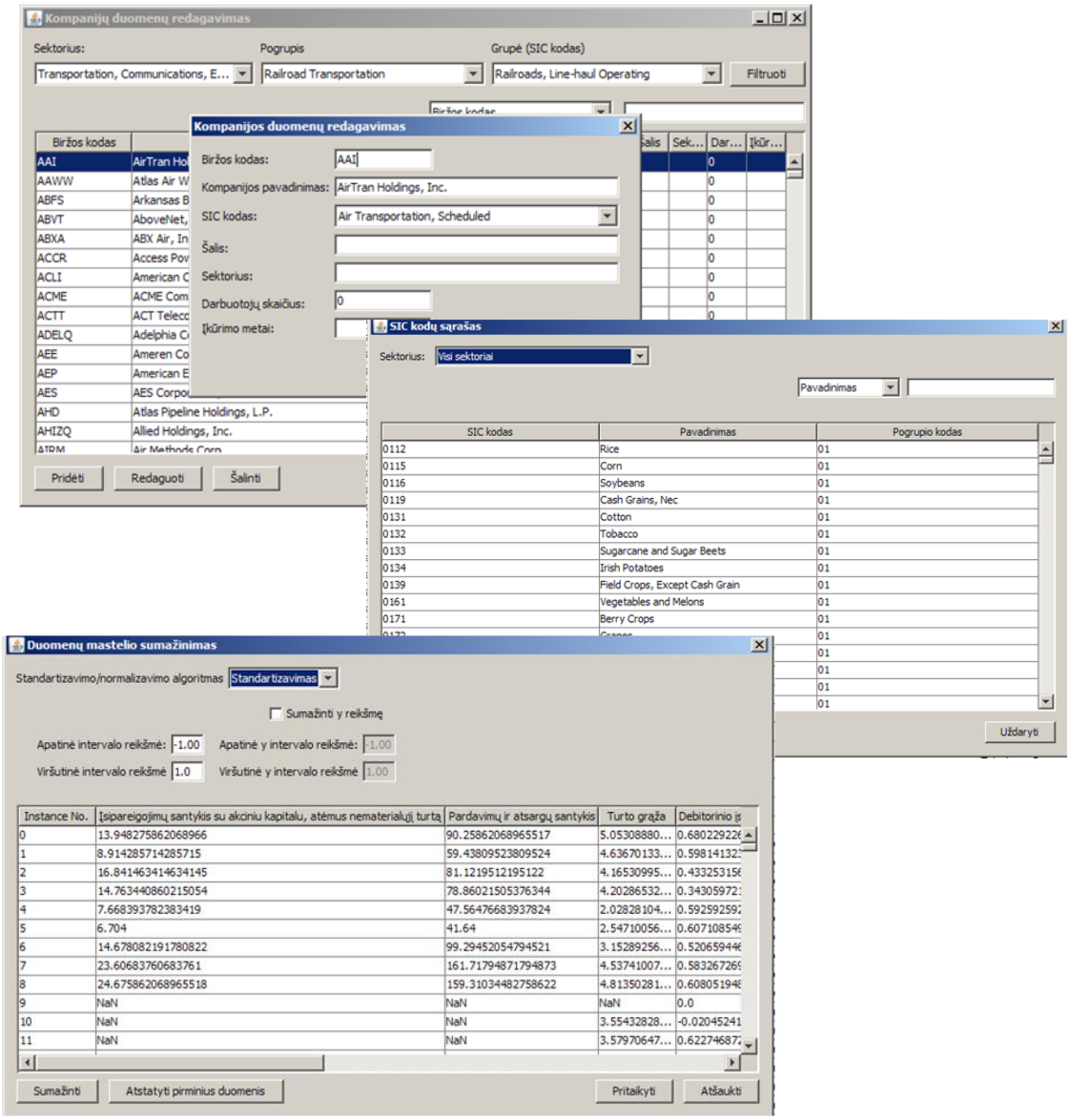
Appendix K

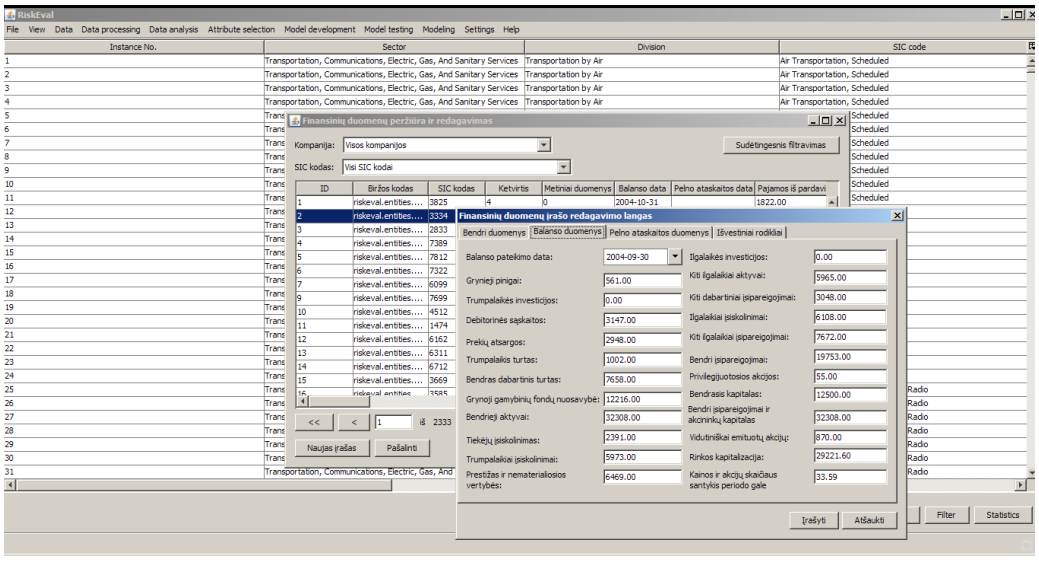
USER INTERFACE EXAMPLES OF DEVELOPED DSS



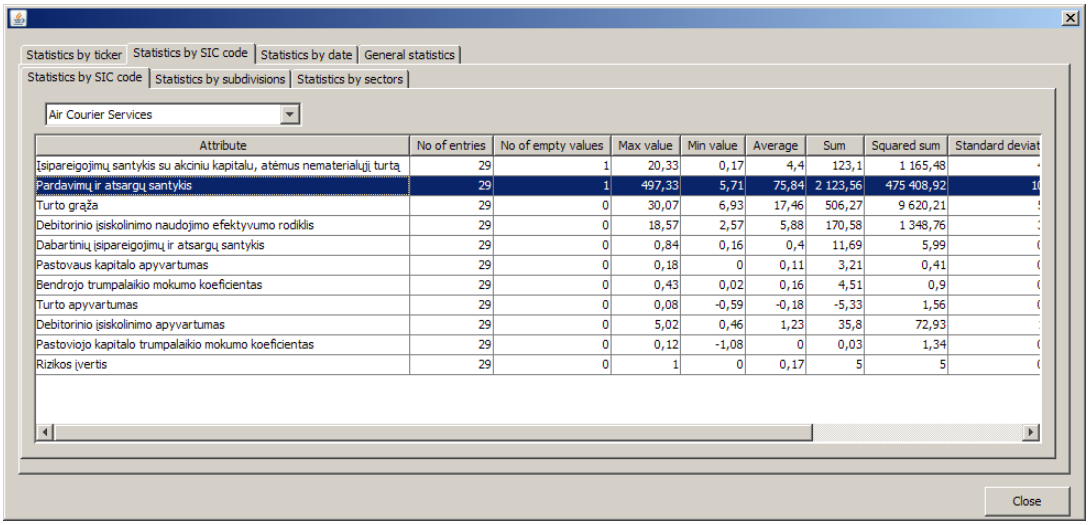


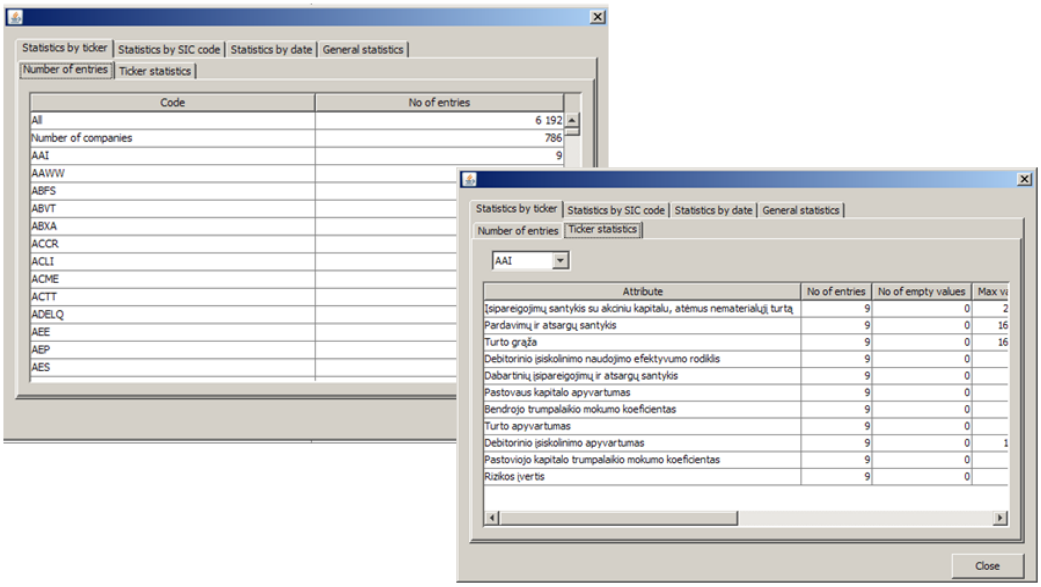


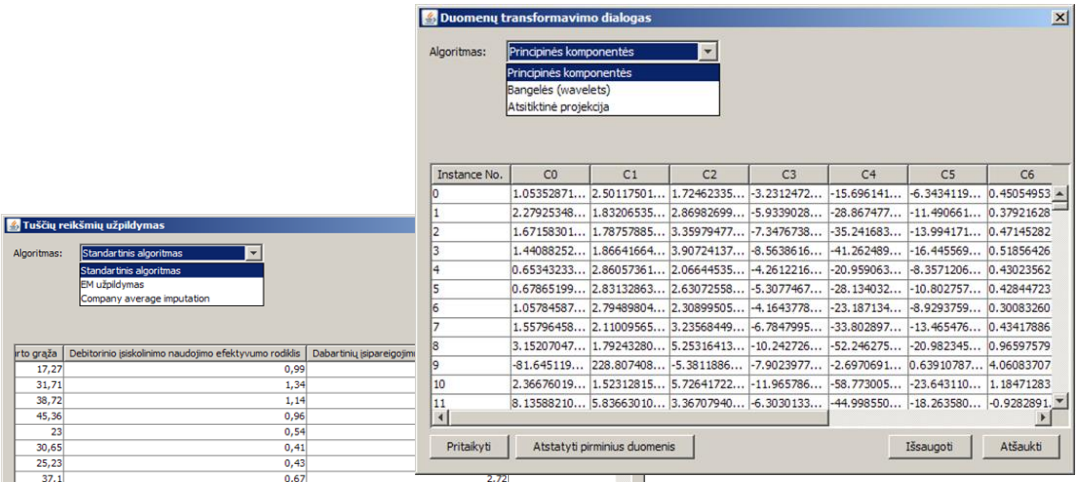


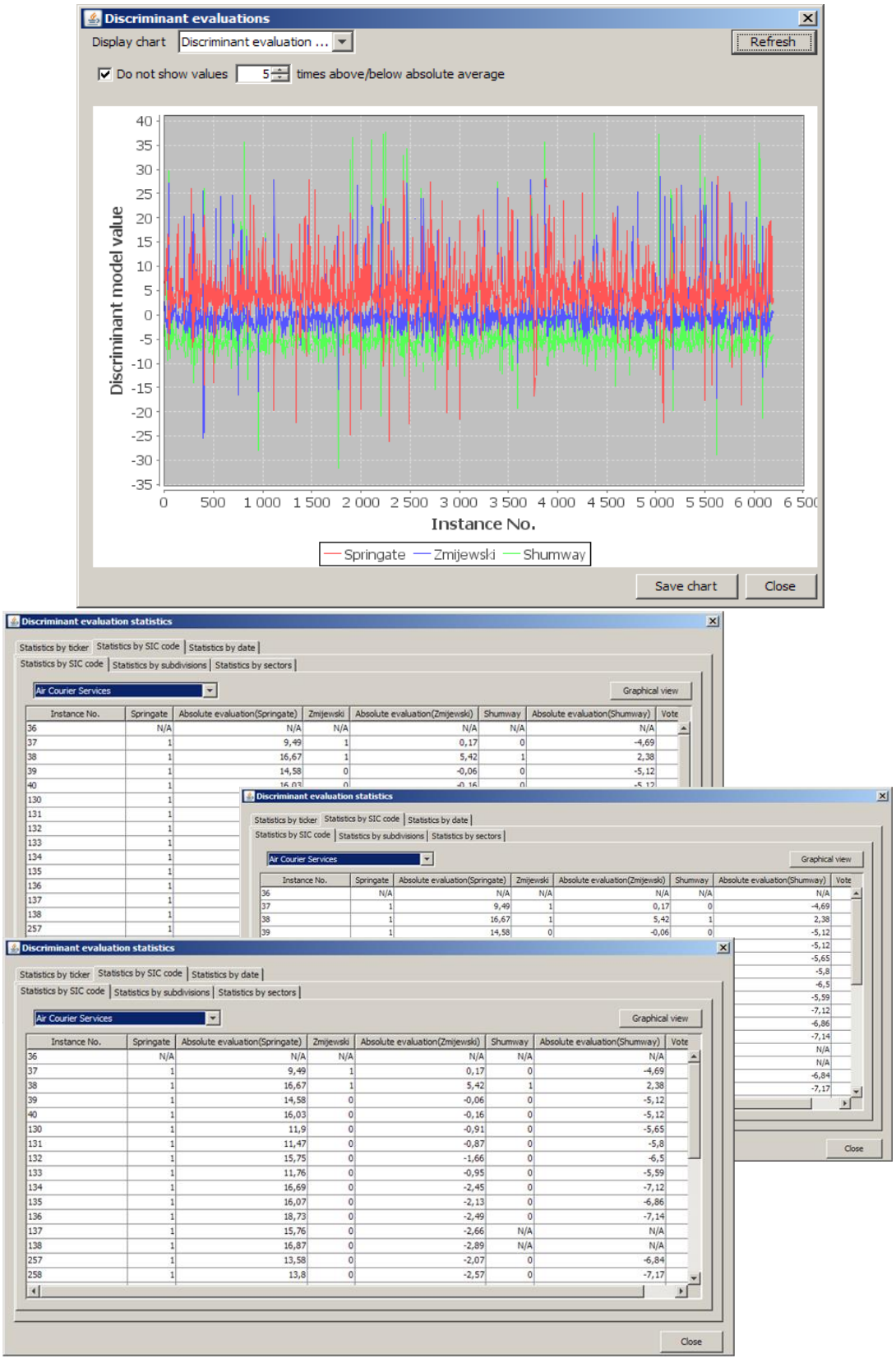


Data entry and selection dialogs

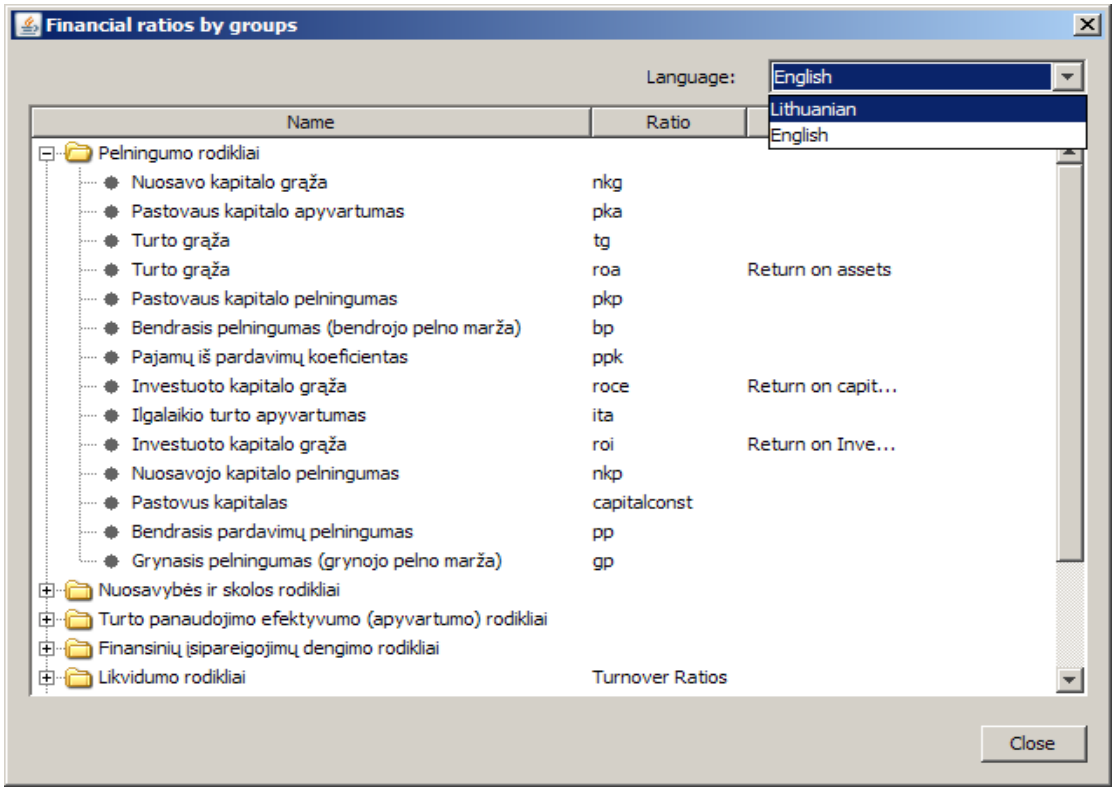

Statistics by ticker
Statistics by SIC code
Statistics by date
General statistics
Statistics by SIC code
Statistics by subdivisions
Statistics by sectors
Air Courier Services
Attribute
No of entries
No of empty values
Max value
Min value
Average
Sum
Squared sum
Standard deviat
Įsipareigojimų santykis su akciniu kapitalu, atėmus nematerialųjį turtą 29 1 20,33 0,17 4,4 123,1 1 165,48
Pardavimų ir atsargų santykis 29 1 497,33 5,71 75,84 2 123,56 475 408,92
Turto grąža 29 0 30,07 6,93 17,46 506,27 9 620,21
Debitorinio įsiskolinimo naudojimo efektyvumo rodiklis 29 0 18,57 2,57 5,88 170,58 1 348,76
Dabartinių įsipareigojimų ir atsargų santykis 29 0 0,84 0,16 0,4 11,69 5,99
Pastovaus kapitalo apyvartumas 29 0 0,18 0 0,11 3,21 0,41
Bendrojo trumpalaikio mokumo koeficientas 29 0 0,43 0,02 0,16 4,51 0,9
Turto apyvartumas 29 0 0,08 -0,59 -0,18 -5,33 1,56
Debitorinio įsiskolinimo apyvartumas 29 0 5,02 0,46 1,23 35,8 72,93
Pastoviojo kapitalo trumpalaikio mokumo koeficientas 29 0 0,12 -1,08 0 0,03 1,34
Rizikos įvertis 29 0 1 0 0,17 5 5
Close

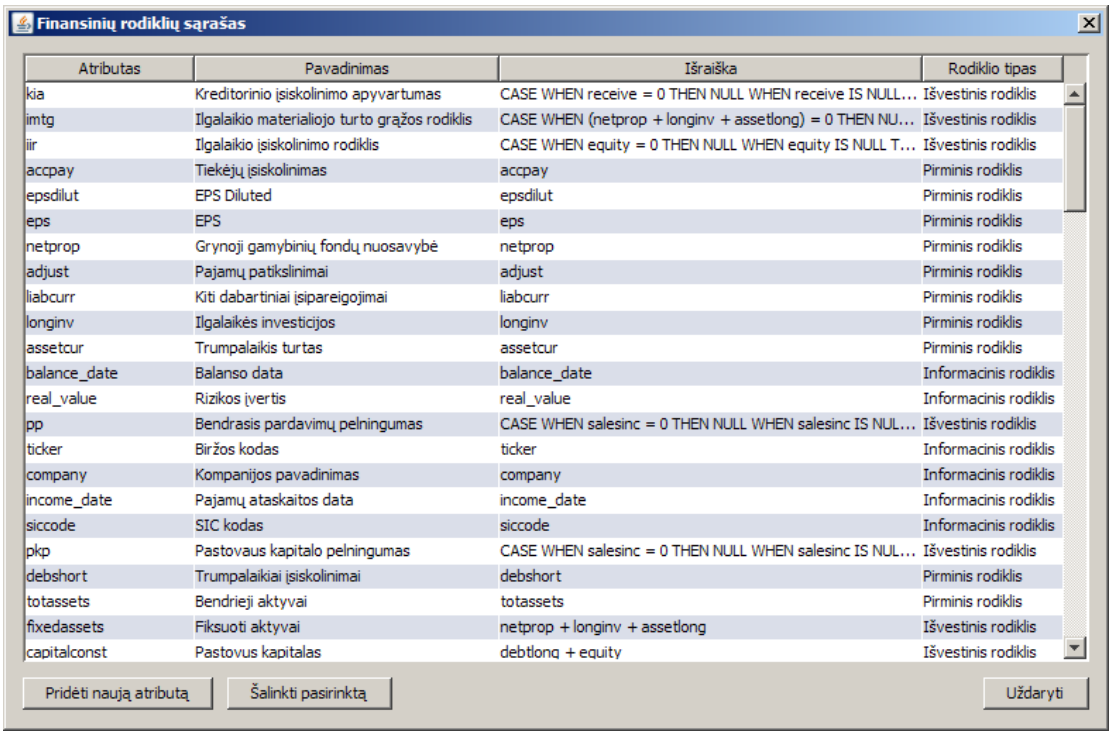

Statistics by ticker
Statistics by SIC code
Statistics by date
General statistics
Number of entries
Ticker statistics
Code
No of entries
All 6 192
Number of companies 786
AAI 9
AAWW
ABFS
ABVT
ABXA
ACCR
ACLI
ACME
ACTT
ADELQ
AEE
AEP
AES
Statistics by ticker
Statistics by SIC code
Statistics by date
General statistics
Number of entries
Ticker statistics
AAI
Attribute
No of entries
No of empty values
Įsipareigojimų santykis su akciniu kapitalu, atėmus nematerialųjį turtą 9 0
Pardavimų ir atsargų santykis 9 0
Turto grąža 9 0
Debitorinio įsiskolinimo naudojimo efektyvumo rodiklis 9 0
Dabartinių įsipareigojimų ir atsargų santykis 9 0
Pastovaus kapitalo apyvartumas 9 0
Bendrojo trumpalaikio mokumo koeficientas 9 0
Turto apyvartumas 9 0
Debitorinio įsiskolinimo apyvartumas 9 0
Pastoviojo kapitalo trumpalaikio mokumo koeficientas 9 0
Rizikos įvertis 9 0
Close


Duomenų transformavimo dialogas
Algoritmas:
Principinės komponentės
Principinės komponentės
Bangelės (wavelets)
Atsitiktinė projekcija
Instance No.
C0
C1
C2
C3
C4
C5
C6
0 1.05352871... 2.50117501... 1.72462335... -3.2312472... -15.696141... -6.3434119... 0.45054953
1 2.27925348... 1.83206535... 2.86982699... -5.9339028... -28.867477... -11.490661... 0.37921628
2 1.67158301... 1.78757885... 3.35979477... -7.3476738... -35.241683... -13.994171... 0.47145282
3 1.44088252... 1.86641664... 3.90724137... -8.5638616... -41.262489... -16.445569... 0.51856426
4 0.65343233... 2.86057361... 2.06644535... -4.2612216... -20.959063... -8.3571206... 0.43023562
5 0.67865199... 2.83132863... 2.63072558... -5.3077467... -28.134032... -10.802757... 0.42844723
6 1.05784587... 2.79489804... 2.30899505... -4.1643778... -23.187134... -8.9293759... 0.30083260
7 1.55796458... 2.11009565... 3.23568449... -6.7847995... -33.802897... -13.465476... 0.43417886
8 3.15207047... 1.79243280... 5.25316413... -10.242726... -52.246275... -20.982345... 0.96597579
9 -81.645119... 228.807408... -5.3811886... -7.9023977... -2.6970691... 0.63910787... 4.06083707
10 2.36676019... 1.52312815... 5.72641722... -11.965786... -58.773005... -23.643110... 1.18471283
11 8.13588210... 5.83663010... 3.36707940... -6.3030133... -44.998550... -18.263580... -0.9282891...
Pritaikyti
Atstatyti pirminius duomenis
Išsaugoti
Atšaukti
Tuščių reikšmių užpildymas
Algoritmas:
Standartinis algoritmas
Standartinis algoritmas
EM užpildymas
Company average imputation
rto grąža
Debitorinio įsiskolinimo naudojimo efektyvumo rodiklis
Dabartinių įsipareigojim
17,27 0,99
31,71 1,34
38,72 1,14
45,36 0,96
23 0,54
30,65 0,41
25,23 0,43
37,1 0,67 2,72
57,39 1,04 1,93
0,42 0,01 314,25
64,67 0,95 2,13
48,78 0 4,18
2 0,67 1,25
1,5 1 0,67
0 2,04 4,18
0 2,04 4,18
0 0 4,18
Pritaikyti
Atstatyti pirminius duomenis
Išsaugoti
Atšaukti



Statistics and preprocessing dialogs





Settings windows

Appendix L

FINANCIAL RATIOS USED IN RESEARCH

Abbreviation	Name of ratio in Lithuanian	Name of ratio in English	Calculation expression
Income statement			
salesinc	Pajamos iš pardavimų	Sales income	
otherinc	Kitos pajamos	Other income	
grossinc	Bendrosios įplaukos	Gross income	
costofg	Parduotų prekių ar paslaugų kaina	Cost of goods sold	
randd	Tyrimai ir plėtra	Research and development	
deprec	Nuvertėjimas	Depreciation	
totalop	Bendros veiklos išlaidos	Total operation expenses	
nonrecc	Neperiodinės lėšos balanse	Nonrecurring items	
interest	Išlaidos palūkanoms	Interest expenses	
totalint	Bendros išlaidos palūkanoms	Total interest expenses	
grossop	Bendrosios veiklos pajamos	Gross operating expenses	
unusual	Netikėtos pajamos	Unusual income	
pretax	Pajamos prieš mokesčius	Pre-tax income	
adjust	Pajamų patikslinimai	Adjustments to income	
inctax	Pajamų mokestis	Income tax	
netincome	Grynasis pelnas	Net income	
sharesav	Vidutinė akcijos kaina	Shares average	
eps	EPS	EPS (akcijos pelnas)	
epscont	Dabartinis (continued) EPS	Continued EPS	
epsdilut	EPS Diluted	Blogiausia tikėtina EPS reikšmė	
dividend	Dividendai	Dividend	
Balance			
cash	Grynieji pinigai	Cash	
shorti	Trumpalaikės investicijos	Short-term investments	
receive	Debitorinės sąskaitos	Receivables	
inventory	Prekių atsargos	Inventory	
assetcur	Trumpalaikis turtas	Other Current Assets	
assettot	Bendras dabartinis turtas	Total Current Assets	
netprop	Grynoji gamybinių fondų nuosavybė	Net property plant and equipment	
longinv	Ilgalaikės investicijos	Long-term investments	
assetlong	Kiti ilgalaikiai aktyvai	Other long-term assets	
goodwill	Prestižo vertė ir nematerialiosios vertybės	Goodwill and intangibles	
totassets	Bendrieji aktyvai	Total assets	
accpay	Tiekėjų įsiskolinimas	Accounts payable	
debshort	Trumpalaikiai įsiskolinimai	Short-term debt	
liabcurr	Kiti dabartiniai įsipareigojimai	Other current liabilities	
debtlong	Ilgalaikiai įsiskolinimai	Long-term debt	
liablong	Kiti ilgalaikiai įsipareigojimai	Other long-term liabilities	
liabtotal	Bendri įsipareigojimai	Total liabilities	
stockpref	Privilegijuotosios akcijos	Stocks preferred	
equity	Bendrasis kapitalas	Common equity	
liabshare	Bendri įsipareigojimai ir akcininkų kapitalas	Total liabilities and shareholder's equity	
average	Vidutiniškai emituotų akcijų	Average shares outstanding	
marketcap	Rinkos kapitalizacija	Market capitalization	
pricetoshare	Kainos ir akcijų skaičiaus santykis periodo gale	Price to share (end of period)	
Derived (secondary) ratios			

Abbreviation	Name of ratio in Lithuanian	Name of ratio in English	Calculation expression
liquidity	Likvidumas	Current ratio	assetcur-debshort
capitalconst	Pastovus kapitalas	Constant capital	equity+debtlong
liabinventory	Trumpalaikių įsipareigojimų ir inventoriaus santykis	Current liabilities to inventory ratio	liabcurr/inventory
totalliab	Visi įsiskolinimai grynai vertei	Total liabilities to net worth ratio	liabtotal/assetlong
salesinventory	Pardavimų ir inventoriaus koeficientas	Sales to inventory ratio	salesinc/inventory
assetssales	Turto ir pardavimų koeficientas	Assets to sales ratio	assettot/salesinc
salescapital	Pardavimų ir grynojo įstatinio kapitalo santykis	Sales to net working capital	salesinc/equity
accountssales	Mokėjimų ir pardavimų koeficientas	Accounts payable to sales ratio	accpay/salesinc
quickratio	Skubaus padengimo koeficientas	Quick ratio	(cash+salesinc)/debshort
pp	Bendrasis pardavimų pelningumas	General profitability ratio	netincome/salesinc
ta	Turto apyvartumas	Asset turnover	salesinc/assettot
pkp	Pastovaus kapitalo pelningumas	Constant capital profitability	netincome/equity /(1-debtlong /equity)
pka	Pastovaus kapitalo apyvartumas	Constant capital turover	(debtlong + equity) / salesinc
vta	Viso apyvartumo rodiklis	Total turnover ratio	salesinc/assettot
ita	Ilgalaikio turto apyvartumas	Fixed assets turnover	salesinc/assetlong
tta	Trumpalaikio turto apyvartumas	Current assets turnover	salesinc/assetcur
dia	Debitorinio įsiskolinimo apyvartumas	Receivables turnover	salesinc/accpay
kia	Kreditorinio įsiskolinimo apyvartumas	Liabilities turnover	salesinc/receive
vs	Veiklos sąnaudos vienam pardavimų vienetui	Total operational costs for each sale unit ratio	totalop/salesinc
pps	Parduotos produkcijos savikaina vienam pardavimui	Cost of goods for each sale unit ratio	costofg/salesinc
deprcoef	Ilgalaikio materialiojo turto nusidėvėjimo keoficientas	Depreciation ratio	deprec /(netprop + longinv + assetlong)
imtg	Ilgalaikio materialiojo turto grąžos rodiklis	Fixed assets return ratio	costofg /(netprop + longinv + assetlong)
imti	Ilgalaikio materialiojo turto imlumo rodiklis	Fixed assets turnover ratio	(netprop + longinv + assetlong)/costofg
die	Debitorinio įsiskolinimo naudojimo efektyvumo rodiklis	Accounts receivables effectiveness ratio	receive/salesinc
btmk	Bendrojo trumpalaikio mokumo koeficientas	Short-time payment ratio	assetcur/debshort
gtmk	Greitojo trumpalaikio mokumo koeficientas	Quick liquidity ratio	(assetcur-inventory)/debshort
ppk	Pajamų iš pardavimų koeficientas	Sales income ratio	(cash + accpay - debshort) /salesinc
pktmk	Pastoviojo kapitalo trumpalaikio mokumo koeficientas	Liquidity ratio for constant capital	(cash + accpay - debshort)/ equity
iir	Ilgalaikio įsiskolinimo rodiklis	Long-term liabilities ratio	debtlong/equity
bsr	Bendras įsiskolinimo rodiklis	Total liabilities ratio	(debshort + debtlong)/ totassets
spkk	Skolos ir akcinio kapitalo koeficientas	Liabilities and equity ratio	(debshort+debtlong)/(debtlong +equity)
isak	Įsipareigojimų santykis su akciniu kapitalu, atėmus nematerialųjį turtą	Liabilities to equity without goodwill and intangibles ratio	(debshort+debtlong)/(equity - goodwill)
sagpk	Skolos apdraustumo grynaisiais pinigais koeficientas	Debt coverage in cash ratio	cash / (debshort + debtlong)
isaitk	Ilgalaikių skolų apdraustumo ilgalaikiu turtu koeficientas	Long term liabilities coverage in fixed assets	(netprop+longinv +assetlong) /debtlong

PSO-LINSVM CLASSIFICATION PERFORMANCE RESULTS

This appendix gives main classification performance results for developed classifier, such as percentage of instances correctly predicted, no of testing instances used, TP ratio, FP ratio, TN ratio, FN ratio, precision, F-Measure and AUC area values for each classes. Average values for each developed classifier are also given, together with classifier parameters obtained by PSO optimization. Two evaluators (Springate and Zmijewski based) were used to form „expert“ evaluations in order to perform classification modeling tasks.

Results of Springate evaluation based classifiers

Results for 10-14 sector (Mining)

	Correct %	**No of testing instances**	**TP Ratio**	**FP ratio**	**F-Measure**	**Precision**	**AUC area**
2000	**87,92**	**385**	**0,659**	**0,341**	**4,204**	**0,843**	**0,659**
PSO_LinSVM for 10-14 [-S, 6, -C, 69.74845662749355, -E, 1.0E-6, -B, 5.0528210932476885]							
Year 2001	88,594	377					0,64
0			0,296	0,015	0,427	0,762	
1			0,985	0,704	0,937	0,893	
Year 2002	88,533	375					0,697
0			0,403	0,01	0,557	0,9	
1			0,99	0,597	0,934	0,884	
Year 2003	86,634	404					0,639
0			0,299	0,021	0,426	0,741	
1			0,979	0,701	0,924	0,875	
2001	**91,286**	**400**	**0,793**	**0,207**	**4,963**	**0,881**	**0,793**
PSO_LinSVM for 10-14 ([-S, 6, -C, 21.815917968204168, -E, 1.0E-6, -B, -12.639228366175635])							
Year 2002	90,4	375					0,807
0			0,657	0,042	0,71	0,772	
1			0,958	0,343	0,942	0,928	
Year 2003	90,347	404					0,763
0			0,552	0,027	0,655	0,804	
1			0,973	0,448	0,944	0,916	
Year 2004	93,112	421					0,81
0			0,629	0,009	0,752	0,936	
1			0,991	0,371	0,96	0,93	

	Correct %	**No of testing instances**	**TP Ratio**	**FP ratio**	**F-Measure**	**Precision**	**AUC area**
2002	**91,342**	**417**	**0,788**	**0,212**	**4,934**	**0,874**	**0,787**
PSO_LinSVM for 10-14 ([-S, 6, -C, 18.297821655048075, -E, 1.0E-6, -B, 1.8434745322949864])							
Year 2003	89,604	404					0,764
0			0,567	0,039	0,644	0,745	
1			0,961	0,433	0,939	0,918	
Year 2004	92,637	421					0,801
0			0,614	0,011	0,735	0,915	
1			0,989	0,386	0,957	0,928	
Year 2005	91,784	426					0,797
0			0,627	0,028	0,706	0,808	
1			0,972	0,373	0,952	0,933	
2003	**91,709**	**398**	**0,781**	**0,219**	**4,927**	**0,892**	**0,78**
PSO_LinSVM for 10-14 ([-S, 6, -C, 160.70481230453694, -E, 1.0E-6, -B, -5.702812472444457])							
Year 2004	93,112	421					0,816
0			0,643	0,011	0,756	0,918	
1			0,989	0,357	0,96	0,933	
Year 2005	91,784	426					0,79
0			0,612	0,025	0,701	0,82	
1			0,975	0,388	0,952	0,931	
Year 2006	90,23	348					0,733
0			0,482	0,017	0,614	0,844	
1			0,983	0,518	0,944	0,908	
2004	**92,055**	**371**	**0,782**	**0,218**	**4,936**	**0,89**	**0,779**
PSO_LinSVM for 10-14 ([-S, 6, -C, 87.59826706903849, -E, 1.0E-6, -B, 2.8349895472409328])							
90490	91,549	426					0,783

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
0			0,597	0,025	0,69	0,816	
1			0,975	0,403	0,951	0,928	
90491	91,379	348					0,768
0			0,554	0,017	0,674	0,861	
1			0,983	0,446	0,95	0,92	
90492	93,235	340					0,787
0			0,596	0,014	0,709	0,875	
1			0,986	0,404	0,962	0,938	
2005	**92,01**	**344**	**0,754**	**0,246**	**3,226**	**0,917**	**0,751**
PSO_LinSVM for 10-14 ([-S, 6, -C, 40.65159347170523, -E, 1.0E-6, -B, 6.514948843925778])							
Year 2006	91,667	348					0,748
0			0,5	0,003	0,659	0,966	
1			0,997	0,5	0,953	0,912	
Year 2007	92,353	340					0,754
0			0,532	0,014	0,658	0,862	
1			0,986	0,468	0,957	0,929	
2006	**93,529**	**340**	**0,784**	**0,216**	**1,674**	**0,933**	**0,779**
PSO_LinSVM for 10-14 ([-S, 6, -C, 90.03645461382085, -E, 1.0E-6, -B, 2.9178401027152487])							
90496	93,529	340					0,779
0			0,574	0,007	0,711	0,931	
1			0,993	0,426	0,964	0,936	

Results for 15-17 sector (Construction)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**50,285**	**44**	**0,729**	**0,271**	**2,805**	**0,745**	**-**
PSO_LinSVM for 15-17 ([-S, 2, -C, 35.203870991490746, -E, 1.0E-6, -B, 2.931428362332511])							
Year 2001	56,522	46					0,457
0			0,333	0,419	0,091	0,053	
1			0,581	0,667	0,714	0,926	
Year 2002	40,476	42					-
1			1	0	1	1	
Year 2003	47,619	42					-
1			1	0	1	1	
2001	**49,784**	**43**	**0,715**	**0,285**	**2,914**	**0,743**	**-**
PSO_LinSVM for 15-17 ([-S, 2, -C, 17.772417640306386, -E, 1.0E-6, -B, 10.852226103435466])							
Year 2002	14,286	42					-
1			1	0	1	1	
Year 2003	16,667	42					-
1			1	0	1	1	
Year 2004	84,091	44					0,43
0			0	0,14	0	0	
1			0,86	1	0,914	0,974	
2002	**74,978**	**43**	**0,771**	**0,229**	**3,581**	**0,695**	**-**
PSO_LinSVM for 15-17 ([-S, 5, -C, 29.362560360622687, -E, 1.0E-6, -B, 6.916731383444758])							
Year 2003	7,143	42					-
1			1	0	1	1	
Year 2004	88,636	44					0,453
0			0	0,093	0	0	
1			0,907	1	0,94	0,975	
Year 2005	95,238	42					0,975
0			1	0,05	0,667	0,5	
1			0,95	0	0,974	1	
2003	**29,629**	**42**	**0,372**	**0,628**	**1,655**	**0,559**	**-**
PSO_LinSVM for 15-17 ([-S, 2, -C, 36.6849990919046, -E, 1.0E-6, -B, 5.011201837246856])							
Year 2004	18,182	44					0,093
0			0	0,814	0	0	
1			0,186	1	0,308	0,889	
Year 2005	19,048	42					0,338
0			0,5	0,825	0,056	0,029	
1			0,175	0,5	0,292	0,875	
Year 2006	73,684	38					-
1			1	0	1	1	
2004	**67,644**	**40**	**0,619**	**0,381**	**2,949**	**0,613**	**-**
PSO_LinSVM for 15-17 ([-S, 5, -C, 37.57654196331089, -E, 1.0E-6, -B, 0.774077355014035])							
Year 2005	80,952	42					0,663
0			0,5	0,175	0,2	0,125	
1			0,825	0,5	0,892	0,971	
Year 2006	26,316	38					-
1			1	0	1	1	
Year 2007	75	40					0,382
0			0	0,231	0	0	
1			0,769	1	0,857	0,968	
2005	**64,211**	**39**	**0,658**	**0,342**	**1,974**	**0,658**	**#NUM!**
PSO_LinSVM for 15-17 ([-S, 5, -C, 51.32982613006854, -E, 1.0E-6, -B, 6.488116991701496])							
Year 2006	2,632	38					-
1			1	0	1	1	
Year 2007	95	40					0,487
0			0	0,026	0	0	
1			0,974	1	0,974	0,974	
2006	**95**	**40**	**0,487**	**0,513**	**0,974**	**0,487**	**0,487**
PSO_LinSVM for 15-17 ([-S, 5, -C, 68.19869870362788, -E, 1.0E-6, -B, -4.651719143747691])							
Year 2007	95	40					0,487
0			0	0,026	0	0	
1			0,974	1	0,974	0,974	

Results for 20-39 sector (Manufacturing)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**72,234**	**2606**	**0,838**	**0,162**	**0,598**	**0,615**	**0,838**
PSO_LinSVM for 20-39 ([-S, 1, -C, 119.58570053485359, -E, 1.0E-6, -B, 8.226551872729926])							
Year 2000	72,734	2571					
0			0,969	0,293	0,35	0,214	0,839
1			0,707	0,031	0,827	0,996	0,839
Year 2001	72,229	2625					
0			0,982	0,302	0,376	0,233	0,84
1			0,698	0,018	0,821	0,998	0,84
Year 2002	71,739	2622					
0			0,984	0,311	0,4	0,251	0,836
1			0,689	0,016	0,815	0,998	0,836
2001	**96,184**	**2637**	**0,919**	**0,081**	**0,889**	**0,865**	**0,919**
PSO_LinSVM for 20-39 ([-S, 1, -C, 102.40179271231908, -E, 1.0E-6, -B, 8.181511863191139])							
Year 2002	96,381	2625					
0			0,893	0,03	0,808	0,738	0,932
1			0,97	0,107	0,98	0,99	0,932
Year 2003	96,262	2622					
0			0,88	0,029	0,819	0,765	0,926
1			0,971	0,12	0,979	0,987	0,926
Year 2004	95,908	2664					
0			0,827	0,029	0,773	0,727	0,899
1			0,971	0,173	0,978	0,984	0,899
2002	**95,28**	**2672**	**0,778**	**0,222**	**0,823**	**0,893**	**0,778**
PSO_LinSVM for 20-39 ([-S, 1, -C, 83.20568487188868, -E, 1.0E-6, -B, -1.7564614910006871])							
Year 2003	95,118	2622					
0			0,594	0,011	0,7	0,851	0,791
1			0,989	0,406	0,973	0,958	0,791
Year 2004	94,97	2664					
0			0,529	0,011	0,64	0,81	0,759
1			0,989	0,471	0,973	0,958	0,759
Year 2005	95,752	2731					
0			0,576	0,011	0,676	0,818	0,783
1			0,989	0,424	0,977	0,966	0,783
2003	**96,85**	**2504**	**0,883**	**0,117**	**0,884**	**0,886**	**0,883**
PSO_LinSVM for 20-39 ([-S, 1, -C, 133.70941538001213, -E, 1.0E-6, -B, 3.1934465667772027])							
Year 2004	96,284	2664					
0			0,764	0,019	0,777	0,789	0,873
1			0,981	0,236	0,98	0,978	0,873
Year 2005	96,814	2731					
0			0,781	0,016	0,79	0,8	0,882
1			0,984	0,219	0,983	0,982	0,882
Year 2006	97,45	2118					
0			0,803	0,015	0,791	0,779	0,894
1			0,985	0,197	0,986	0,987	0,894
2004	**96,188**	**2273**	**0,93**	**0,07**	**0,865**	**0,819**	**0,93**
PSO_LinSVM for 20-39 ([-S, 1, -C, 117.79445617931488, -E, 1.0E-6, -B, 12.398835885956121])							
Year 2005	95,899	2731					
0			0,89	0,035	0,77	0,678	0,927
1			0,965	0,11	0,977	0,991	0,927
Year 2006	96,317	2118					
0			0,913	0,034	0,748	0,634	0,94
1			0,966	0,087	0,98	0,994	0,94
Year 2007	96,347	1971					
0			0,875	0,031	0,731	0,628	0,922
1			0,969	0,125	0,98	0,992	0,922
2005	**97,641**	**2045**	**0,917**	**0,083**	**0,898**	**0,881**	**0,917**
PSO_LinSVM for 20-39 [-S, 1, -C, 216.86555434297745, -E, 1.0E-6, -B, -14.495920851150819]							
Year 2006	97,262	2118					
0			0,843	0,019	0,787	0,738	0,912
1			0,981	0,157	0,985	0,99	0,912
Year 2007	98,021	1971					
0			0,857	0,012	0,831	0,807	0,922
1			0,988	0,143	0,989	0,991	0,922
2006	**95,535**	**1971**	**0,922**	**0,078**	**0,834**	**0,781**	**0,922**
PSO_LinSVM for 20-39 [-S, 1, -C, 79.62291043044591, -E, 1.0E-6, -B, -0.14414150801651004]							
Year 2007	95,535	1971					
0			0,884	0,04	0,692	0,569	0,922
1			0,96	0,116	0,976	0,993	0,922

Results for 40-49 sector (Transportation, Communications, Electric, Gas and Sanitary Services)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**94,673**	**639**	**0,689**	**0,311**	**4,444**	**0,848**	**0,689**
PSO_LinSVM for 40-49 ([-S, 6, -C, 67.57442193139572, -E, 1.0E-6, -B, 3.4270138908697407])							
Year 2001	94,295	631					0,685
0			0,386	0,015	0,486	0,654	
1			0,985	0,614	0,97	0,955	
Year 2002	94,778	632					0,681
0			0,37	0,007	0,507	0,81	
1			0,993	0,63	0,972	0,953	
Year 2003	94,946	653					0,702
0			0,413	0,01	0,535	0,76	
1			0,99	0,587	0,973	0,957	

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2001	**95,21**	**650**	**0,716**	**0,284**	**4,597**	**0,864**	**0,716**
PSO_LinSVM for 40-49 [-S, 5, -C, 107.11498971424494, -E, 1.0E-6, -B, 2.4767574943359296]							
Year 2002	93,829	632					0,676
0			0,37	0,017	0,466	0,63	
1			0,983	0,63	0,967	0,952	
Year 2003	95,712	653					0,766
0			0,543	0,012	0,641	0,781	
1			0,988	0,457	0,977	0,966	
Year 2004	96,09	665					0,706
0			0,415	0,003	0,567	0,895	
1			0,997	0,585	0,98	0,963	
2002	**96,324**	**662**	**0,718**	**0,282**	**4,739**	**0,958**	**0,724**
PSO_LinSVM for 40-49 [-S, 1, -C, 111.70008628506906, -E, 1.0E-6, -B, 10.400614506805564]							
Year 2003	96,018	653					0,737
0			0,478	0,003	0,629	0,917	
1			0,997	0,522	0,979	0,962	
Year 2004	96,391	665					0,707
0			0,415	0	0,586	1	
1			1	0,585	0,981	0,963	
Year 2005	96,562	669					0,728
0			0,421	0,002	0,582	0,941	
1			0,998	0,579	0,982	0,966	
2003	**97,155**	**612**	**0,77**	**0,23**	**4,927**	**0,902**	**0,768**
PSO_LinSVM for 40-49 ([-S, 6, -C, 64.30231604475975, -E, 1.0E-6, -B, 6.934921034460606])							
Year 2004	97,444	665					0,804
0			0,61	0,002	0,746	0,962	
1			0,998	0,39	0,987	0,975	0,804
Year 2005	96,413	669					0,752
0			0,526	0,01	0,625	0,769	
1			0,99	0,474	0,981	0,972	
Year 2006	97,61	502					0,747
0			0,5	0,006	0,6	0,75	
1			0,994	0,5	0,988	0,982	
2004	**97,248**	**554**	**0,726**	**0,274**	**4,702**	**0,889**	**0,727**
PSO_LinSVM for 40-49 ([-S, 6, -C, 87.79967186777459, -E, 1.0E-6, -B, 3.029815109183411])							
Year 2005	95,964	669					0,697
0			0,395	0,006	0,526	0,789	
1			0,994	0,605	0,979	0,965	
Year 2006	97,41	502					0,719
0			0,444	0,006	0,552	0,727	
1			0,994	0,556	0,987	0,98	
Year 2007	98,371	491					0,766
0			0,533	0,002	0,667	0,889	
1			0,998	0,467	0,992	0,985	
2005	**97,492**	**497**	**0,786**	**0,214**	**3,203**	**0,817**	**0,786**

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
PSO_LinSVM for 40-49 ([-S, 6, -C, 87.2661286548255, -E, 1.0E-6, -B, 1.1904686155583826])							
Year 2006	96,614	502					0,742
0			0,5	0,017	0,514	0,529	
1			0,983	0,5	0,982	0,981	
Year 2007	98,371	491					0,83
0			0,667	0,006	0,714	0,769	
1			0,994	0,333	0,992	0,99	
2006	**98,371**	**491**	**0,766**	**0,234**	**1,658**	**0,937**	**0,766**
PSO_LinSVM for 40-49 ([-S, 6, -C, 69.078967006108, -E, 1.0E-6, -B, -0.8203210708849866])							
Year 2007	98,371	491					0,766
0			0,533	0,002	0,667	0,889	
1			0,998	0,467	0,992	0,985	

Results for 50-51 sector (Wholesale Trade)

	Correct %	# of test instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**96,027**	**218**	**0,688**	**0,312**	**4,364**	**0,8**	**0,688**
PSO_LinSVM for 50-51 ([-S, 6, -C, 57.31959332230502, -E, 1.0E-6, -B, 6.876351907739073])							
Year 2001	95,964	223					0,645
0			0,3	0,009	0,4	0,6	
1			0,991	0,7	0,979	0,968	
Year 2002	95,928	221					0,677
0			0,364	0,01	0,471	0,667	
1			0,99	0,636	0,979	0,967	
Year 2003	96,19	210					0,743
0			0,5	0,015	0,556	0,625	
1			0,985	0,5	0,98	0,975	
2001	**95,309**	**214**	**0,576**	**0,424**	**3,702**	**0,867**	**0,576**
PSO_LinSVM for 50-51 ([-S, 5, -C, 33.9895414961413, -E, 1.0E-6, -B, 0.12770372944171537])							
Year 2002	95,928	221					0,591
0			0,182	0	0,308	1	
1			1	0,818	0,979	0,959	
Year 2003	96,19	210					0,6
0			0,2	0	0,333	1	
1			1	0,8	0,98	0,962	
Year 2004	93,81	210					0,537
0			0,083	0,01	0,133	0,333	
1			0,99	0,917	0,968	0,947	
2002	**96,508**	**210**	**0,683**	**0,317**	**4,481**	**0,982**	**0,683**
PSO_LinSVM for 50-51 ([-S, 6, -C, 93.81675962588795, -E, 1.0E-6, -B, 5.152855694181657])							
Year 2003	98,095	210					0,8
0			0,6	0	0,75	1	

	Correct %	# of test instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
1			1	0,4	0,99	0,98	
Year 2004	96,19	210					0,667
0			0,333	0	0,5	1	
1			1	0,667	0,98	0,961	
Year 2005	95,238	210					0,583
0			0,167	0	0,286	1	
1			1	0,833	0,975	0,952	
2003	**96,526**	**194**	**0,65**	**0,35**	**4,303**	**0,982**	**0,65**
PSO_LinSVM for 50-51 ([-S, 5, -C, 53.480694315327156, -E, 1.0E-6, -B, 5.138408942041586])							
Year 2004	96,19	210					0,667
0			0,333	0	0,5	1	
1			1	0,667	0,98	0,961	
Year 2005	95,238	210					0,583
0			0,167	0	0,286	1	
1			1	0,833	0,975	0,952	
Year 2006	98,148	162					0,7
0			0,4	0	0,571	1	
1			1	0,6	0,991	0,981	
2004	**98,571**	**174**	**0,863**	**0,137**	**5,265**	**0,938**	**0,863**
PSO_LinSVM for 50-51 ([-S, 6, -C, 66.84863219398595, -E, 1.0E-6, -B, 21.52690357466631])							
Year 2005	97,619	210					0,792
0			0,583	0	0,737	1	
1			1	0,417	0,988	0,975	
Year 2006	98,765	162					0,8
0			0,6	0	0,75	1	
1			1	0,4	0,994	0,987	
Year 2007	99,329	149					0,997
0			1	0,007	0,8	0,667	
1			0,993	0	0,997	1	
2005	**98,43**	**156**	**0,675**	**0,325**	**2,984**	**0,992**	**0,675**
PSO_LinSVM for 50-51 [-S, 6, -C, 34.197220747518536, -E, 1.0E-6, -B, -0.0887036966285153]							
Year 2006	97,531	162					0,6
0			0,2	0	0,333	1	
1			1	0,8	0,987	0,975	
Year 2007	99,329	149					0,75
0			0,5	0	0,667	1	
1			1	0,5	0,997	0,993	
2006	**98,658**	**149**	**0,5**	**0,5**	**0,993**	**0,493**	**0,5**
PSO_LinSVM for 50-51 ([-S, 6, -C, 48.491821034963415, -E, 1.0E-6, -B, 2.1370868357751602])							
Year 2007	98,658	149					0,5
0			0	0	0	0	
1			1	1	0,993	0,987	

Results for 52-59 sector (Transportation, Communications, Electric, Gas, And Sanitary Services)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**96,195**	**332**	**0,79**	**0,21**	**3,882**	**0,624**	**0,79**
PSO_LinSVM for 52-59 ([-S, 6, -C, 6.524008986482985, -E, 1.0E-6, -B, -4.312664839749965])							
Year 2001	94,294	333					0,644
0			0,333	0,046	0,174	0,118	
1			0,954	0,667	0,97	0,987	
Year 2002	96,736	337					0,984
0			1	0,033	0,267	0,154	
1			0,967	0	0,983	1	
Year 2003	97,554	327					0,744
0			0,5	0,013	0,5	0,5	
1			0,987	0,5	0,987	0,987	
2001	**92,779**	**328**	**0,643**	**0,357**	**3,244**	**0,532**	**0,643**
PSO_LinSVM for 52-59 ([-S, 6, -C, 123.62502129661794, -E, 1.0E-6, -B, -4.500800136035719])							
Year 2002	93,175	337					0,717
0			0,5	0,066	0,08	0,043	
1			0,934	0,5	0,965	0,997	
Year 2003	92,661	327					0,658
0			0,375	0,06	0,2	0,136	
1			0,94	0,625	0,962	0,984	
Year 2004	92,5	320					0,553
0			0,167	0,061	0,077	0,05	
1			0,939	0,833	0,961	0,983	
2002	**95,373**	**332**	**0,787**	**0,213**	**4,006**	**0,624**	**0,787**
PSO_LinSVM for 52-59 ([-S, 6, -C, 44.334633168131866, -E, 1.0E-6, -B, 0.8819975993611597])							
Year 2003	95,719	327					0,795
0			0,625	0,034	0,417	0,313	
1			0,966	0,375	0,978	0,99	
Year 2004	94,688	320					0,728
0			0,5	0,045	0,261	0,176	
1			0,955	0,5	0,972	0,99	
Year 2005	95,714	350					0,838
0			0,714	0,038	0,4	0,278	
1			0,962	0,286	0,978	0,994	
2003	**79,537**	**325**	**0,8**	**0,2**	**3,011**	**0,53**	**0,8**
PSO_LinSVM for 52-59 ([-S, 6, -C, 62.31148464014272, -E, 1.0E-6, -B, -2.945341009932439])							
Year 2004	79,688	320					0,733
0			0,667	0,201	0,11	0,06	
1			0,799	0,333	0,885	0,992	
Year 2005	78,857	350					0,893
0			1	0,216	0,159	0,086	
1			0,784	0	0,879	1	
Year 2006	80,065	306					0,775
0			0,75	0,199	0,09	0,048	
1			0,801	0,25	0,888	0,996	

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2004	**59,241**	**300**	**0,711**	**0,289**	**2,393**	**0,514**	**0,71**
PSO_LinSVM for 52-59 ([-S, 6, -C, 43.72360630927101, -E, 1.0E-6, -B, 6.388913243347346])							
Year 2005	60,857	350					0,8
0			1	0,399	0,093	0,049	
1			0,601	0	0,75	1	
Year 2006	60,131	306					0,796
0			1	0,404	0,062	0,032	
1			0,596	0	0,747	1	
Year 2007	56,735	245					0,535
0			0,5	0,432	0,019	0,009	
1			0,568	0,5	0,723	0,993	
2005	**81,992**	**276**	**0,785**	**0,215**	**1,966**	**0,521**	**0,785**
PSO_LinSVM for 52-59 ([-S, 1, -C, 46.059802511860454, -E, 1.0E-6, -B, 1.3689685777033826])							
Year 2006	80,719	306					0,901
0			1	0,195	0,119	0,063	
1			0,805	0	0,892	1	
Year 2007	83,265	245					0,669
0			0,5	0,165	0,047	0,024	
1			0,835	0,5	0,908	0,995	
2006	**41,633**	**245**	**0,458**	**0,542**	**0,599**	**0,499**	**0,459**
PSO_LinSVM for 52-59 ([-S, 2, -C, 45.92568307543694, -E, 1.0E-6, -B, 5.579280251852021])							
Year 2007	41,633	245					0,459
0			0,5	0,584	0,014	0,007	
1			0,416	0,5	0,586	0,99	

Results for sector 60-67 (Finance, Insurance, And Real Estate)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**83,618**	**189**	**0,804**	**0,196**	**0,757**	**0,733**	**0,804**
PSO_LinSVM for 60-67 ([-S, 5, -C, 58.491598245432534, -E, 1.0E-6, -B, 4.416023370819847])							
Year 2001	83,158	190					0,788
0			0,724	0,149	0,568	0,467	
1			0,851	0,276	0,895	0,945	
Year 2002	85,405	185					0,832
0			0,8	0,135	0,64	0,533	
1			0,865	0,2	0,908	0,957	
Year 2003	82,292	192					0,792
0			0,738	0,153	0,646	0,574	
1			0,847	0,262	0,882	0,92	
2001	**90,665**	**189**	**0,862**	**0,138**	**0,849**	**0,839**	**0,862**
PSO_LinSVM for 60-67 ([-S, 1, -C, 108.71603847553826, -E, 1.0E-6, -B, 6.349758672505693])							

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
Year 2002	91,892	185					0,871
0			0,8	0,058	0,762	0,727	
1			0,942	0,2	0,951	0,961	
Year 2003	90,104	192					0,86
0			0,786	0,067	0,776	0,767	
1			0,933	0,214	0,936	0,94	
Year 2004	90	190					0,856
0			0,788	0,076	0,732	0,684	
1			0,924	0,212	0,939	0,954	
2002	**84,875**	**192**	**0,768**	**0,232**	**0,768**	**0,767**	**0,768**
PSO_LinSVM for 60-67 ([-S, 2, -C, 54.89913409380594, -E, 1.0E-6, -B, 4.518637483348714])							
Year 2003	82,292	192					0,741
0			0,595	0,113	0,595	0,595	
1			0,887	0,405	0,887	0,887	
Year 2004	86,842	190					0,765
0			0,606	0,076	0,615	0,625	
1			0,924	0,394	0,921	0,918	
Year 2005	85,492	193					0,799
0			0,698	0,1	0,682	0,667	
1			0,9	0,302	0,906	0,912	
2003	**93,917**	**174**	**0,922**	**0,078**	**0,907**	**0,893**	**0,922**
PSO_LinSVM for 60-67 ([-S, 1, -C, 152.2761380303566, -E, 1.0E-6, -B, -5.187671423594309])							
Year 2004	93,684	190					0,926
0			0,909	0,057	0,833	0,769	
1			0,943	0,091	0,961	0,98	
Year 2005	93,782	193					0,918
0			0,884	0,047	0,864	0,844	
1			0,953	0,116	0,96	0,966	
Year 2006	94,286	140					0,922
0			0,889	0,044	0,857	0,828	
1			0,956	0,111	0,964	0,973	
2004	**91,461**	**153**	**0,853**	**0,147**	**0,851**	**0,851**	**0,853**
PSO_LinSVM for 60-67 ([-S, 1, -C, 47.12889689259531, -E, 1.0E-6, -B, 2.8865775973828462])							
Year 2005	93,264	193					0,915
0			0,884	0,053	0,854	0,826	
1			0,947	0,116	0,956	0,966	
Year 2006	92,143	140					0,881
0			0,815	0,053	0,8	0,786	
1			0,947	0,185	0,951	0,955	
Year 2007	88,976	127					0,762
0			0,579	0,056	0,611	0,647	
1			0,944	0,421	0,936	0,927	
2005	**88,774**	**134**	**0,85**	**0,15**	**0,817**	**0,795**	**0,85**
PSO_LinSVM for 60-67 ([-S, 2, -C, 63.89591766722812, -E, 1.0E-6, -B, 1.8072241283206456])							
Year 2006	88,571	140					

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
0			0,852	0,106	0,742	0,657	0,873
1	88,571	140	0,894	0,148	0,927	0,962	0,873
Year 2007	88,976	127					
0			0,737	0,083	0,667	0,609	0,827
1			0,917	0,263	0,934	0,952	0,827
2006	**92,913**	**127**	**0,828**	**0,172**	**0,851**	**0,879**	**0,828**
PSO_LinSVM for 60-67 ([-S, 1, -C, 76.44594783430352, -E, 1.0E-6, -B, 9.805089935896442])							
Year 2007	92,913	127					0,828
0			0,684	0,028	0,743	0,813	
1			0,972	0,316	0,959	0,946	

Results for sector 70-89 (Services)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**94,493**	**1331**	**0,846**	**0,154**	**0,839**	**0,834**	**0,846**
PSO_LinSVM for 70-89 ([-S, 5, -C, 112.29932288598613, -E, 1.0E-6, -B, 7.465686905009287])							
Year 2001	94,034	1341					
0			0,736	0,041	0,669	0,614	0,847
1			0,959	0,264	0,967	0,976	0,847
Year 2002	94,872	1326					
0			0,748	0,03	0,736	0,725	0,859
1			0,97	0,252	0,972	0,973	0,859
Year 2003	94,574	1327					
0			0,692	0,026	0,719	0,748	0,833
1			0,974	0,308	0,97	0,966	0,833
2001	**94,084**	**1330**	**0,841**	**0,159**	**0,832**	**0,825**	**0,841**
PSO_LinSVM for 70-89 ([-S, 1, -C, 150.00620212975147, -E, 1.0E-6, -B, 2.997513694361287])							
Year 2002	93,741	1326					
0			0,748	0,043	0,696	0,651	0,853
1			0,957	0,252	0,965	0,973	0,853
Year 2003	93,821	1327					
0			0,692	0,034	0,692	0,692	0,829
1			0,966	0,308	0,966	0,966	0,829
Year 2004	94,69	1337					
0			0,714	0,03	0,705	0,697	0,842
1			0,97	0,286	0,971	0,972	0,842
2002	**95,024**	**1333**	**0,843**	**0,157**	**0,851**	**0,859**	**0,843**
PSO_LinSVM for 70-89 ([-S, 1, -C, 70.84096408223614, -E, 1.0E-6, -B, 8.826602320723444])							
Year 2003	94,65	1327					
0			0,699	0,026	0,724	0,75	0,837
1			0,974	0,301	0,97	0,967	0,837
Year 2004	95,363	1337					
0			0,723	0,024	0,735	0,748	0,849
1			0,976	0,277	0,975	0,973	0,849
Year 2005	95,06	1336					
0			0,712	0,025	0,73	0,748	0,844
1			0,975	0,288	0,973	0,97	0,844
2003	**92,528**	**1225**	**0,82**	**0,18**	**0,789**	**0,765**	**0,82**
PSO_LinSVM for 70-89 ([-S, 1, -C, 152.90450584200707, -E, 1.0E-6, -B, -12.257790203415217])							
Year 2004	92,969	1337					
0			0,731	0,051	0,649	0,584	0,84
1			0,949	0,269	0,961	0,973	0,84
Year 2005	92,59	1336					
0			0,68	0,049	0,632	0,59	0,816
1			0,951	0,32	0,959	0,966	0,816
Year 2006	92,024	1003					
0			0,667	0,057	0,574	0,505	0,805
1			0,943	0,333	0,956	0,97	0,805
2004	**94,541**	**1082**	**0,863**	**0,137**	**0,833**	**0,809**	**0,863**
PSO_LinSVM for 70-89 ([-S, 1, -C, 145.11060652610146, -E, 1.0E-6, -B, 7.875979030877845])							
Year 2005	94,536	1336					
0			0,792	0,039	0,731	0,678	0,877
1			0,961	0,208	0,97	0,978	0,877
Year 2006	93,719	1003					
0			0,716	0,043	0,648	0,592	0,836
1			0,957	0,284	0,966	0,975	0,836
Year 2007	95,369	907					
0			0,788	0,033	0,712	0,65	0,877
1			0,967	0,212	0,975	0,983	0,877
2005	**94,793**	**955**	**0,792**	**0,208**	**0,807**	**0,825**	**0,792**
PSO_LinSVM for 70-89 ([-S, 1, -C, 132.36306797190827, -E, 1.0E-6, -B, -12.620695708123165])							
Year 2006	94,217	1003					
0			0,593	0,027	0,623	0,658	0,783
1			0,973	0,407	0,969	0,965	0,783
Year 2007	95,369	907					
0			0,621	0,02	0,661	0,707	0,8
1			0,98	0,379	0,975	0,971	0,8
2006	**93,164**	**907**	**0,865**	**0,135**	**0,794**	**0,751**	**0,865**
PSO_LinSVM for 70-89 ([-S, 1, -C, 164.6990894199772, -E, 1.0E-6, -B, 3.67143165559044])							
Year 2007	93,164	907					
0			0,788	0,057	0,627	0,52	0,865
1			0,943	0,212	0,962	0,983	0,865

Results of Zmijewski evaluation based classifiers

Results for 01-09 sector

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**71,905**	**27**	**0,748**	**0,252**	**0,662**	**0,667**	**0,748**
PSO_LinSVM for 01-09 [-S, 1, -C, 54.114151641647375, -E, 1.0E-6, -B, -0.08486930728542141]							
Year 2001	80	25					
0			0,81	0,25	0,872	0,944	0,78
1			0,75	0,19	0,545	0,429	0,78
Year 2002	67,857	28					
0			0,619	0,143	0,743	0,929	0,738
1			0,857	0,381	0,571	0,429	0,738
Year 2003	67,857	28					
0			0,652	0,2	0,769	0,938	0,726
1			0,8	0,348	0,471	0,333	0,726
2001	**86,538**	**27**	**0,738**	**0,262**	**0,769**	**0,831**	**0,738**
PSO_LinSVM for 01-09 ([-S, 0, -C, 1.0, -E, 1.0E-6, -B, -6.044516332807723])							
Year 2002	82,143	28					
0			0,905	0,429	0,884	0,864	0,738
1			0,571	0,095	0,615	0,667	0,738
Year 2003	92,857	28					
0			1	0,4	0,958	0,92	0,8
1			0,6	0	0,75	1	0,8
Year 2004	84,615	26					
0			0,952	0,6	0,909	0,87	0,676
1			0,4	0,048	0,5	0,667	0,676
2002	**91,304**	**27**	**0,837**	**0,163**	**0,857**	**0,887**	**0,857**
PSO_LinSVM for 01-09 ([-S, 3, -C, 47.94290595713359, -E, 1.0E-6, -B, 5.89925330044301])							
Year 2003	92,857	28					
0			0,957	0,2	0,957	0,957	0,878
1			0,8	0,043	0,8	0,8	0,878
Year 2004	88,462	26					
0			0,952	0,4	0,93	0,909	0,776
1			0,6	0,048	0,667	0,75	0,776
Year 2005	92,593	27					
0			1	0,286	0,952	0,909	0,917
1			0,714	0	0,833	1	0,917
2003	**93,115**	**24**	**0,856**	**0,144**	**0,884**	**0,933**	**0,848**
PSO_LinSVM for 01-09 ([-S, 2, -C, 92.70087756788104, -E, 1.0E-6, -B, -2.8104360508983715])							
Year 2004	92,308	26					
0			0,952	0,2	0,952	0,952	0,876
1			0,8	0,048	0,8	0,8	0,876
Year 2005	92,593	27					
0			1	0,286	0,952	0,909	0,833
1			0,714	0	0,833	1	0,833
Year 2006	94,444	18					
0			1	0,333	0,968	0,938	0,833
1			0,667	0	0,8	1	0,833
2004	**91,748**	**21**	**0,864**	**0,136**	**0,878**	**0,905**	**0,86**
PSO_LinSVM for 01-09 ([-S, 6, -C, 69.73395962409731, -E, 1.0E-6, -B, 1.7200777443187505])							
Year 2005	96,296	27					
0			1	0,143	0,976	0,952	0,917
1			0,857	0	0,923	1	0,917
Year 2006	100	18					
0			1	0	1	1	1
1			1	0	1	1	1
Year 2007	78,947	19					
0			0,929	0,6	0,867	0,813	0,664
1			0,4	0,071	0,5	0,667	0,664
2005	**92,105**	**19**	**0,882**	**0,118**	**0,891**	**0,904**	**0,882**
PSO_LinSVM for 01-09 ([-S, 5, -C, 46.68656802539996, -E, 1.0E-6, -B, 2.098234455864053])							
Year 2006	100	18					
0			1	0	1	1	1
1			1	0	1	1	1
Year 2007	84,211	19					
0			0,929	0,4	0,897	0,867	0,764
1			0,6	0,071	0,667	0,75	0,764
2006	**84,211**	**19**	**0,929**	**0,4**	**0,897**	**0,867**	**0,764**
PSO_LinSVM for 01-09 ([-S, 6, -C, 110.39051672856617, -E, 1.0E-6, -B, 4.557221950391694])							
Year 2007	84,211	19					
0			0,929	0,4	0,897	0,867	0,764
1			0,6	0,071	0,667	0,75	0,764

Results for 10-14 sector (Mining)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**94,473**	**303**	**0,931**	**0,069**	**0,929**	**0,928**	**0,933**
PSO_LinSVM for 10-14 ([-S, 3, -C, 41.30964847253258, -E, 1.0E-6, -B, 2.4829071014334034])							
Year 2001	94,218	294					
0			0,963	0,115	0,961	0,959	0,929
1			0,885	0,037	0,89	0,896	0,929
Year 2002	93,919	296					
0			0,943	0,071	0,957	0,971	0,936

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
1			0,929	0,057	0,897	0,867	0,936
Year 2003	95,283	318					
0			0,971	0,104	0,969	0,967	0,934
1			0,896	0,029	0,902	0,908	0,934
2001	**56,179**	**316**	**0,704**	**0,296**	**0,558**	**0,675**	**0,704**
PSO_LinSVM for 10-14 ([-S, 3, -C, 58.72323899002096, -E, 1.0E-6, -B, 10.119131678092756])							
Year 2002	56,419	296					
0			0,392	0	0,563	1	0,696
1			1	0,608	0,566	0,394	0,696
Year 2003	55,66	318					
0			0,423	0,026	0,591	0,981	0,699
1			0,974	0,577	0,515	0,35	0,699
Year 2004	56,456	333					
0			0,446	0,014	0,615	0,991	0,716
1			0,986	0,554	0,498	0,333	0,716
2002	**96,9**	**333**	**0,943**	**0,057**	**0,953**	**0,965**	**0,943**
PSO_LinSVM for 10-14 ([-S, 0, -C, 41.680235838612475, -E, 1.0E-6, -B, 5.679919311212791])							
Year 2003	97,484	318					
0			0,988	0,065	0,983	0,979	0,961
1			0,935	0,012	0,947	0,96	0,961
Year 2004	96,096	333					
0			0,981	0,11	0,975	0,97	0,936
1			0,89	0,019	0,909	0,929	0,936
Year 2005	97,118	347					
0			0,996	0,132	0,982	0,969	0,931
1			0,868	0,004	0,922	0,983	0,931
2003	**97,423**	**323**	**0,947**	**0,053**	**0,958**	**0,971**	**0,946**
PSO_LinSVM for 10-14 ([-S, 6, -C, 13.478202678199947, -E, 1.0E-6, -B, 3.5657078976159218])							
Year 2004	96,997	333					
0			0,988	0,096	0,981	0,973	0,946
1			0,904	0,012	0,93	0,957	0,946
Year 2005	97,695	347					
0			0,996	0,103	0,986	0,975	0,946
1			0,897	0,004	0,938	0,984	0,946
Year 2006	97,578	289					
0			0,992	0,098	0,985	0,979	0,947
1			0,902	0,008	0,929	0,958	0,947
2004	**93,873**	**311**	**0,879**	**0,121**	**0,894**	**0,911**	**0,877**
PSO_LinSVM for 10-14 ([-S, 0, -C, 6.447196428235575, -E, 1.0E-6, -B, -0.9851238129381283])							
Year 2005	94,236	347					
0			0,978	0,206	0,965	0,951	0,885
1			0,794	0,022	0,844	0,9	0,885
Year 2006	94,118	289					
0			0,975	0,216	0,965	0,955	0,88
1			0,784	0,025	0,825	0,87	0,88

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
Year 2007	93,266	297					
			0,967	0,222	0,959	0,951	0,868
1			0,778	0,033	0,808	0,84	0,868
2005	**94,894**	**293**	**0,869**	**0,131**	**0,904**	**0,955**	**0,871**
PSO_LinSVM for 10-14 ([-S, 3, -C, 30.373714456938412, -E, 1.0E-6, -B, 8.640523349910104])							
Year 2006	95,848	289					
0			0,992	0,196	0,975	0,959	0,898
1			0,804	0,008	0,872	0,953	0,898
Year 2007	93,939	297					
0			0,996	0,315	0,964	0,934	0,844
1			0,685	0,004	0,804	0,974	0,844
2006	**96,296**	**297**	**0,92**	**0,08**	**0,935**	**0,953**	**0,917**
PSO_LinSVM for 10-14 ([-S, 6, -C, 5.591998192712291, -E, 1.0E-6, -B, 1.179875991887521])							
Year 2007	96,296	297					
0			0,988	0,148	0,978	0,968	0,917
1			0,852	0,012	0,893	0,939	0,917

Results for 15-17 sector (Construction)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**92,485**	**59**	**0,921**	**0,079**	**0,9**	**0,893**	**0,921**
PSO_LinSVM for 15-17 ([-S, 6, -C, 90.90197422492342, -E, 1.0E-6, -B, 1.8618602743841768])							
Year 2001	98,387	62					
0			0,978	0	0,989	1	0,989
1			1	0,022	0,97	0,941	0,989
Year 2002	85,965	57					
0			0,947	0,316	0,9	0,857	0,816
1			0,684	0,053	0,765	0,867	0,816
Year 2003	93,103	58					
0			0,918	0	0,957	1	0,959
1			1	0,082	0,818	0,692	0,959
2001	**80,876**	**59**	**0,735**	**0,265**	**0,723**	**0,723**	**0,735**
PSO_LinSVM for 15-17 ([-S, 6, -C, 44.6347157481821, -E, 1.0E-6, -B, -0.8981359336172923])							
Year 2002	77,193	57					
0			0,895	0,474	0,84	0,791	0,711
1			0,526	0,105	0,606	0,714	0,711
Year 2003	84,483	58					
0			0,878	0,333	0,905	0,935	0,772
1			0,667	0,122	0,571	0,5	0,772
Year 2004	80,952	63					
0			0,863	0,417	0,88	0,898	0,723

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
1			0,583	0,137	0,538	0,5	0,723
2002	**89,194**	**62**	**0,786**	**0,214**	**0,797**	**0,822**	**0,786**
PSO_LinSVM for 15-17 ([-S, 3, -C, 54.18855495615824, -E, 1.0E-6, -B, -1.218098356747974])							
Year 2003	89,655	58					
0			0,959	0,444	0,94	0,922	0,757
1			0,556	0,041	0,625	0,714	0,757
Year 2004	87,302	63					
0			0,961	0,5	0,925	0,891	0,73
1			0,5	0,039	0,6	0,75	0,73
Year 2005	90,625	64					
0			0,925	0,182	0,942	0,961	0,871
1			0,818	0,075	0,75	0,692	0,871
2003	**86,045**	**60**	**0,67**	**0,33**	**0,672**	**0,675**	**0,67**
PSO_LinSVM for 15-17 ([-S, 1, -C, 101.09608689806993, -E, 1.0E-6, -B, 4.747759635988293])							
Year 2004	87,302	63					
0			0,941	0,417	0,923	0,906	0,762
1			0,583	0,059	0,636	0,7	0,762
Year 2005	87,5	64					
0			0,925	0,364	0,925	0,925	0,78
1			0,636	0,075	0,636	0,636	0,78
Year 2006	83,333	54					
0			0,938	1	0,909	0,882	0,469
1			0	0,063	0	0	0,469
2004	**91,854**	**58**	**0,717**	**0,283**	**0,765**	**0,956**	**0,717**
PSO_LinSVM for 15-17 ([-S, 3, -C, 49.93036764718021, -E, 1.0E-6, -B, 0.28419031815306606])							
Year 2005	93,75	64					
0			1	0,364	0,964	0,93	0,818
1			0,636	0	0,778	1	0,818
Year 2006	90,741	54					
0			1	0,833	0,95	0,906	0,583
1			0,167	0	0,286	1	0,583
Year 2007	91,071	56					
0			1	0,5	0,948	0,902	0,75
1			0,5	0	0,667	1	0,75
2005	**90,013**	**55**	**0,678**	**0,322**	**0,726**	**0,868**	**0,678**
PSO_LinSVM for 15-17 ([-S, 6, -C, 22.16415203333628, -E, 1.0E-6, -B, -7.381175476378298])							
Year 2006	90,741	54					
0			0,979	0,667	0,949	0,922	0,656
1			0,333	0,021	0,444	0,667	0,656
Year 2007	89,286	56					
0			1	0,6	0,939	0,885	0,7
1			0,4	0	0,571	1	0,7
2006	**91,071**	**56**	**0,75**	**0,25**	**0,808**	**0,951**	**0,75**
PSO_LinSVM for 15-17 ([-S, 1, -C, 82.973056521839, -E, 1.0E-6, -B, 7.6676061418679])							
Year 2007	91,071	56					
0			1	0,5	0,948	0,902	0,75
1			0,5	0	0,667	1	0,75

Results for 20-39 sector (Manufacturing)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**94,326**	**2470**	**0,92**	**0,08**	**0,93**	**0,942**	**0,92**
PSO_LinSVM for 20-39 ([-S, 0, -C, 46.5067568219524, -E, 1.0E-6, -B, -3.551933423504959])							
Year 2001	95,218	2426					
0			0,984	0,131	0,967	0,951	0,927
1			0,869	0,016	0,91	0,954	0,927
Year 2002	93,853	2489					
0			0,98	0,153	0,956	0,934	0,914
1			0,847	0,02	0,896	0,951	0,914
Year 2003	93,908	2495					
0			0,967	0,13	0,957	0,948	0,919
1			0,87	0,033	0,893	0,916	0,919
2001	**95,386**	**2514**	**0,937**	**0,063**	**0,942**	**0,947**	**0,937**
PSO_LinSVM for 20-39 ([-S, 3, -C, 9.453191443581012, -E, 1.0E-6, -B, 9.533661333885911])							
Year 2002	95,46	2489					
0			0,977	0,095	0,967	0,958	0,941
1			0,905	0,023	0,926	0,948	0,941
Year 2003	95,311	2495					
0			0,972	0,094	0,967	0,962	0,939
1			0,906	0,028	0,918	0,931	0,939
Year 2004	95,387	2558					
0			0,976	0,111	0,969	0,963	0,932
1			0,889	0,024	0,906	0,923	0,932
2002	**89,025**	**2566**	**0,805**	**0,195**	**0,84**	**0,91**	**0,806**
PSO_LinSVM for 20-39 ([-S, 3, -C, 20.04522651410609, -E, 1.0E-6, -B, 3.5256686013453944])							
Year 2003	87,655	2495					
0			0,979	0,374	0,918	0,865	0,803
1			0,626	0,021	0,747	0,925	0,803
Year 2004	89,367	2558					
0			0,984	0,38	0,933	0,887	0,802
1			0,62	0,016	0,744	0,929	0,802
Year 2005	90,053	2644					
0			0,993	0,371	0,937	0,887	0,812
1			0,629	0,007	0,762	0,968	0,812
2003	**95,123**	**2423**	**0,915**	**0,085**	**0,93**	**0,948**	**0,915**
PSO_LinSVM for 20-39 ([-S, 0, -C, 76.07408799853452, -E, 1.0E-6, -B, -0.6640793163458989])							

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
Year 2004	94,253	2558					
0			0,976	0,159	0,962	0,949	0,909
1			0,841	0,024	0,879	0,921	0,909
Year 2005	94,743	2644					
0			0,985	0,164	0,965	0,946	0,91
1			0,836	0,015	0,89	0,951	0,91
Year 2006	96,373	2068					
0			0,99	0,137	0,977	0,965	0,926
1			0,863	0,01	0,908	0,959	0,926
2004	**91,898**	**2217**	**0,863**	**0,137**	**0,877**	**0,894**	**0,863**
PSO_LinSVM for 20-39 ([-S, 1, -C, 1.0, -E, 1.0E-6, -B, 5.206777482567917])							
Year 2005	91,679	2644					
0			0,967	0,231	0,945	0,925	0,868
1			0,769	0,033	0,824	0,888	0,868
Year 2006	92,94	2068					
0			0,969	0,223	0,956	0,943	0,873
1			0,777	0,031	0,821	0,87	0,873
Year 2007	91,073	1938					
0			0,954	0,26	0,945	0,935	0,847
1			0,74	0,046	0,771	0,804	0,847
2005	**92,632**	**2003**	**0,876**	**0,124**	**0,885**	**0,894**	**0,876**
PSO_LinSVM for 20-39 ([-S, 3, -C, 32.11519999132183, -E, 1.0E-6, -B, 6.75474146108926])							
Year 2006	93,52	2068					
0			0,968	0,188	0,959	0,951	0,889
1			0,812	0,032	0,839	0,868	0,889
Year 2007	91,744	1938					
0			0,955	0,229	0,949	0,942	0,863
1			0,771	0,045	0,791	0,812	0,863
2006	**86,12**	**1938**	**0,867**	**0,133**	**0,814**	**0,787**	**0,867**
PSO_LinSVM for 20-39 ([-S, 3, -C, 40.2580740280937, -E, 1.0E-6, -B, 2.2369171952739 29])							
Year 2007	86,12	1938					
0			0,857	0,122	0,908	0,965	0,867
1			0,878	0,143	0,719	0,61	0,867

Results for 40-49 sector (Transportation, Communications, Electric, Gas, And Sanitary Services)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**36,648**	**622**	**0,494**	**0,506**	**0,273**	**0,4**	**0,494**
PSO_LinSVM for 40-49 ([-S, 1, -C, 59.04995564190367, -E, 1.0E-6, -B, 6.693359195524913])							
Year 2000	36,275	612					
0			0,003	0,031	0,005	0,125	0,486
1			0,969	0,997	0,531	0,366	0,486
Year 2001	39,448	616					
0			0,011	0,004	0,021	0,8	0,503
1			0,996	0,989	0,562	0,391	0,503
Year 2002	34,223	637					
0			0,007	0,023	0,014	0,375	0,492
1			0,977	0,993	0,506	0,342	0,492
2001	**78,394**	**636**	**0,702**	**0,298**	**0,717**	**0,812**	**0,702**
PSO_LinSVM for 40-49 ([-S, 7, -C, 36.831428442167905, -E, 1.0E-6, -B, 1.9118777277972465])							
Year 2002	78,084	616					
0			0,963	0,504	0,843	0,749	0,728
1			0,496	0,037	0,638	0,895	0,728
Year 2003	76,609	637					
0			0,962	0,605	0,843	0,751	0,679
1			0,395	0,038	0,539	0,845	0,679
Year 2004	80,488	656					
0			0,963	0,569	0,874	0,8	0,698
1			0,431	0,037	0,568	0,832	0,698
2002	**82,899**	**650**	**0,743**	**0,257**	**0,768**	**0,838**	**0,741**
PSO_LinSVM for 40-49 ([-S, 6, -C, 27.38272120267969, -E, 1.0E-6, -B, -2.073903583485262])							
Year 2003	81,319	637					
0			0,957	0,459	0,87	0,798	0,749
1			0,541	0,043	0,667	0,869	0,749
Year 2004	84,299	656					
0			0,961	0,436	0,896	0,839	0,761
1			0,564	0,039	0,681	0,859	0,761
Year 2005	83,079	656					
0			0,965	0,531	0,893	0,831	0,713
1			0,469	0,035	0,599	0,83	0,713
2003	**84,402**	**602**	**0,742**	**0,258**	**0,771**	**0,842**	**0,741**
PSO_LinSVM for 40-49 ([-S, 0, -C, 59.83175257746429, -E, 1.0E-6, -B, -2.388975088327405])							
Year 2004	83,689	656					
0			0,948	0,426	0,891	0,84	0,76
1			0,574	0,052	0,677	0,824	0,76
Year 2005	83,079	656					
0			0,956	0,508	0,892	0,836	0,721
1			0,492	0,044	0,611	0,806	0,721
Year 2006	86,437	494					
0			0,979	0,496	0,916	0,862	0,744
1			0,504	0,021	0,642	0,882	0,744
2004	**81,586**	**545**	**0,645**	**0,355**	**0,67**	**0,855**	**0,64**
PSO_LinSVM for 40-49 ([-S, 1, -C, 16.477889051577552, -E, 1.0E-6, -B, 4.20763608945838])							
Year 2005	78,659	656					
0			0,975	0,723	0,87	0,785	0,62

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
1			0,277	0,025	0,412	0,803	0,62
Year 2006	82,389	494					
0			0,989	0,697	0,895	0,817	0,647
1			0,303	0,011	0,453	0,9	0,647
Year 2007	83,711	485					
0			1	0,675	0,903	0,823	0,652
1			0,325	0	0,49	1	0,652
2005	**83,76**	**490**	**0,714**	**0,286**	**0,742**	**0,806**	**0,711**
PSO_LinSVM for 40-49 ([-S, 6, -C, 15.432895256502297, -E, 1.0E-6, -B, 5.921152939907926])							
Year 2006	83,603	494					
0			0,952	0,529	0,898	0,85	0,713
1			0,471	0,048	0,58	0,757	0,713
Year 2007	83,918	485					
0			0,954	0,521	0,9	0,852	0,709
1			0,479	0,046	0,589	0,767	0,709
2006	**84,124**	**485**	**0,7**	**0,3**	**0,734**	**0,838**	**0,692**
PSO_LinSVM for 40-49 ([-S, 1, -C, 11.013830390386175, -E, 1.0E-6, -B, 0.46912277470891756])							
Year 2007	84,124	485					
0			0,973	0,573	0,903	0,842	0,692
1			0,427	0,027	0,565	0,833	0,692

Results for 50-51 sector (Wholesale Trade)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**90,551**	**204**	**0,853**	**0,147**	**0,873**	**0,905**	**0,855**
PSO_LinSVM for 50-51 ([-S, 5, -C, 37.48920017378519, -E, 1.0E-6, -B, 3.246705486379551])							
Year 2001	89,474	209					
0			0,966	0,27	0,928	0,892	0,854
1			0,73	0,034	0,807	0,902	0,854
Year 2002	90,385	208					
0			0,986	0,3	0,936	0,89	0,843
1			0,7	0,014	0,808	0,955	0,843
Year 2003	91,795	195					
0			0,96	0,222	0,947	0,935	0,869
1			0,778	0,04	0,814	0,854	0,869
2001	**82,513**	**200**	**0,758**	**0,242**	**0,76**	**0,767**	**0,757**
PSO_LinSVM for 50-51 ([-S, 0, -C, 41.94001248198117, -E, 1.0E-6, -B, -3.351580741095351])							
Year 2002	82,212	208					
0			0,919	0,417	0,88	0,845	0,751
1			0,583	0,081	0,654	0,745	0,751
Year 2003	84,615	195					
0			0,9	0,333	0,9	0,9	0,783
1			0,667	0,1	0,667	0,667	0,783
Year 2004	80,711	197					
0			0,863	0,386	0,874	0,886	0,738
1			0,614	0,137	0,587	0,563	0,738
2002	**88,387**	**198**	**0,807**	**0,193**	**0,825**	**0,849**	**0,807**
PSO_LinSVM for 50-51 ([-S, 6, -C, 87.52030870146993, -E, 1.0E-6, -B, 6.257694803645651])							
Year 2003	88,718	195					
0			0,96	0,356	0,929	0,9	0,802
1			0,644	0,04	0,725	0,829	0,802
Year 2004	88,325	197					
0			0,935	0,295	0,926	0,917	0,82
1			0,705	0,065	0,729	0,756	0,82
Year 2005	88,119	202					
0			0,949	0,348	0,925	0,902	0,8
1			0,652	0,051	0,714	0,789	0,8
2003	**89,491**	**186**	**0,813**	**0,187**	**0,832**	**0,859**	**0,813**
PSO_LinSVM for 50-51 ([-S, 6, -C, 28.282573619420702, -E, 1.0E-6, -B, 9.054033400260241])							
Year 2004	89,34	197					
0			0,961	0,341	0,933	0,907	0,81
1			0,659	0,039	0,734	0,829	0,81
Year 2005	86,634	202					
0			0,949	0,413	0,916	0,886	0,768
1			0,587	0,051	0,667	0,771	0,768
Year 2006	92,5	160					
0			0,962	0,241	0,955	0,947	0,86
1			0,759	0,038	0,786	0,815	0,86
2004	**88,782**	**170**	**0,795**	**0,205**	**0,811**	**0,832**	**0,795**
PSO_LinSVM for 50-51 ([-S, 5, -C, 27.217773020510684, -E, 1.0E-6, -B, 1.0235617740783258])							
Year 2005	86,634	202					
0			0,949	0,413	0,916	0,886	0,768
1			0,587	0,051	0,667	0,771	0,768
Year 2006	91,875	160					
0			0,962	0,276	0,951	0,94	0,843
1			0,724	0,038	0,764	0,808	0,843
Year 2007	87,838	148					
0			0,934	0,385	0,927	0,919	0,775
1			0,615	0,066	0,64	0,667	0,775
2005	**90,169**	**154**	**0,825**	**0,175**	**0,83**	**0,836**	**0,825**
PSO_LinSVM for 50-51 ([-S, 6, -C, 59.32962810425914, -E, 1.0E-6, -B, 3.023038391613204])							
Year 2006	92,5	160					
0			0,962	0,241	0,955	0,947	0,86
1			0,759	0,038	0,786	0,815	0,86
Year 2007	87,838	148					
0			0,926	0,346	0,926	0,926	0,79

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
1			0,654	0,074	0,654	0,654	0,79
2006	**87,838**	**148**	**0,79**	**0,21**	**0,79**	**0,79**	**0,79**
PSO_LinSVM for 50-51 ([-S, 6, -C, 60.81437706269477, -E, 1.0E-6, -B, -7.052362999126573])							
Year 2007	87,838	148					
0			0,926	0,346	0,926	0,926	0,79
1			0,654	0,074	0,654	0,654	0,79

Results for sector 60-67 (Finance, Insurance and Real Estate)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**87,354**	**966**	**0,869**	**0,131**	**0,869**	**0,87**	**0,869**
PSO_LinSVM for 60-67 ([-S, 7, -C, 75.82402252461432, -E, 1.0E-6, -B, 4.862656716534167])							
Year 2001	86,484	910					
0			0,835	0,115	0,831	0,828	0,86
1			0,885	0,165	0,887	0,89	0,86
Year 2002	86,722	964					
0			0,832	0,109	0,835	0,837	0,861
1			0,891	0,168	0,889	0,888	0,861
Year 2003	88,856	1023					
0			0,865	0,092	0,875	0,884	0,886
1			0,908	0,135	0,9	0,892	0,886
2001	**82,834**	**1013**	**0,817**	**0,183**	**0,821**	**0,844**	**0,817**
PSO_LinSVM for 60-67 ([-S, 3, -C, 39.7582056086078, -E, 1.0E-6, -B, 2.073310539294263])							
Year 2002	84,025	964					
0			0,724	0,082	0,785	0,857	0,821
1			0,918	0,276	0,873	0,832	0,821
Year 2003	84,457	1023					
0			0,728	0,06	0,808	0,908	0,834
1			0,94	0,272	0,869	0,809	0,834
Year 2004	80,019	1051					
0			0,645	0,05	0,76	0,925	0,797
1			0,95	0,355	0,829	0,736	0,797
2002	**86,929**	**1057**	**0,872**	**0,128**	**0,869**	**0,877**	**0,871**
PSO_LinSVM for 60-67 ([-S, 3, -C, 30.075205463276347, -E, 1.0E-6, -B, 4.408205293789532])							
Year 2003	85,435	1023					
0			0,943	0,218	0,853	0,779	0,862
1			0,782	0,057	0,855	0,944	0,862
Year 2004	86,108	1051					
0			0,936	0,211	0,868	0,81	0,862
1			0,789	0,064	0,853	0,928	0,862
Year 2005	89,243	1097					
0			0,963	0,184	0,903	0,85	0,889
1			0,816	0,037	0,879	0,953	0,889
2003	**93,908**	**1023**	**0,942**	**0,058**	**0,939**	**0,939**	**0,941**
PSO_LinSVM for 60-67 ([-S, 0, -C, 63.33894966371913, -E, 1.0E-6, -B, 8.710011892568655])							
Year 2004	93,245	1051					
0			0,891	0,028	0,928	0,968	0,932
1			0,972	0,109	0,936	0,903	0,932
Year 2005	94,348	1097					
0			0,905	0,015	0,943	0,985	0,945
1			0,985	0,095	0,944	0,906	0,945
Year 2006	94,13	920					
0			0,912	0,016	0,948	0,988	0,948
1			0,984	0,088	0,932	0,885	0,948
2004	**94,354**	**979**	**0,947**	**0,053**	**0,943**	**0,942**	**0,947**
PSO_LinSVM for 60-67 ([-S, 7, -C, 1.7298204765583056, -E, 1.0E-6, -B, 2.72357491587243])							
Year 2005	93,71	1097					
0			0,9	0,023	0,937	0,977	0,938
1			0,977	0,1	0,937	0,9	0,938
Year 2006	94,348	920					
0			0,915	0,016	0,95	0,988	0,95
1			0,984	0,085	0,934	0,889	0,95
Year 2007	95,005	921					
0			0,935	0,03	0,955	0,976	0,953
1			0,97	0,065	0,943	0,919	0,953
2005	**88,865**	**921**	**0,888**	**0,112**	**0,886**	**0,885**	**0,889**
PSO_LinSVM for 60-67 ([-S, 7, -C, 11.108748253784086, -E, 1.0E-6, -B, -2.713087283875636])							
Year 2006	90,109	920					
0			0,903	0,101	0,915	0,928	0,901
1			0,899	0,097	0,881	0,864	0,901
Year 2007	87,622	921					
0			0,878	0,126	0,89	0,902	0,878
1			0,874	0,122	0,859	0,844	0,878
2006	**93,268**	**921**	**0,936**	**0,064**	**0,932**	**0,93**	**0,937**
PSO_LinSVM for 60-67 ([-S, 6, -C, 39.07380626389353, -E, 1.0E-6, -B, 1.415377950912424])							
Year 2007	93,268	921					
0			0,91	0,038	0,939	0,97	0,937
1			0,962	0,09	0,925	0,89	0,937

Results for sector 70-89 (Services)

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
2000	**71,391**	**1288**	**0,645**	**0,355**	**0,631**	**0,811**	**0,645**

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
PSO_LinSVM for 70-89 ([-S, 1, -C, 39.25977999958768, -E, 1.0E-6, -B, 0.2004286212677331])							
Year 2001	69,758	1283					
0			0,986	0,686	0,788	0,656	0,65
1			0,314	0,014	0,471	0,945	0,65
Year 2002	71,04	1288					
0			0,992	0,69	0,801	0,671	0,651
1			0,31	0,008	0,469	0,965	0,651
Year 2003	73,375	1292					
0			0,986	0,72	0,826	0,712	0,633
1			0,28	0,014	0,429	0,915	0,633
2001	**82,103**	**1299**	**0,784**	**0,216**	**0,795**	**0,821**	**0,785**
PSO_LinSVM for 70-89 ([-S, 5, -C, 49.5779349868302, -E, 1.0E-6, -B, 3.88104622262101 94])							
Year 2002	79,736	1288					
0			0,925	0,383	0,843	0,774	0,771
1			0,617	0,075	0,715	0,852	0,771
Year 2003	82,43	1292					
0			0,925	0,358	0,871	0,823	0,784
1			0,642	0,075	0,723	0,827	0,784
Year 2004	84,143	1318					
0			0,916	0,318	0,887	0,861	0,8
1			0,682	0,084	0,732	0,789	0,8
2002	**86,458**	**1311**	**0,833**	**0,167**	**0,841**	**0,852**	**0,834**
PSO_LinSVM for 70-89 ([-S, 5, -C, 1.0, -E, 1.0E-6, -B, 7.530257820763359])							
Year 2003	84,443	1292					
0			0,917	0,286	0,883	0,852	0,815
1			0,714	0,083	0,766	0,827	0,815
Year 2004	86,646	1318					
0			0,922	0,254	0,904	0,887	0,835
1			0,746	0,078	0,78	0,817	0,835
Year 2005	88,284	1323					
0			0,931	0,23	0,918	0,905	0,851
1			0,77	0,069	0,797	0,827	0,851
2003	**89,927**	**1215**	**0,875**	**0,125**	**0,878**	**0,882**	**0,875**
PSO_LinSVM for 70-89 ([-S, 5, -C, 1.0, -E, 1.0E-6, -B, 24.94990333797131])							
Year 2004	89,681	1318					
0			0,93	0,175	0,925	0,92	0,877
1			0,825	0,07	0,835	0,846	0,877

	Correct %	No of testing instances	TP Ratio	FP ratio	F-Measure	Precision	AUC area
Year 2005	89,872	1323					
0			0,936	0,189	0,928	0,92	0,873
1			0,811	0,064	0,827	0,845	0,873
Year 2006	90,229	1003					
0			0,938	0,191	0,933	0,928	0,873
1			0,809	0,062	0,821	0,833	0,873
2004	**84,499**	**1081**	**0,795**	**0,205**	**0,804**	**0,813**	**0,795**
PSO_LinSVM for 70-89 ([-S, 0, -C, 30.058695118124945, -E, 1.0E-6, -B, 1.996461487952415])							
Year 2005	86,319	1323					
0			0,924	0,28	0,904	0,885	0,822
1			0,72	0,076	0,759	0,803	0,822
Year 2006	84,845	1003					
0			0,91	0,314	0,897	0,884	0,798
1			0,686	0,09	0,714	0,745	0,798
Year 2007	82,334	917					
0			0,894	0,362	0,88	0,866	0,766
1			0,638	0,106	0,667	0,698	0,766
2005	**83,625**	**960**	**0,754**	**0,246**	**0,775**	**0,815**	**0,754**
PSO_LinSVM for 70-89 ([-S, 5, -C, 29.158267357123187, -E, 1.0E-6, -B, -3.9831728119709457])							
Year 2006	85,244	1003					
0			0,949	0,401	0,903	0,861	0,774
1			0,599	0,051	0,692	0,818	0,774
Year 2007	82,007	917					
0			0,928	0,461	0,882	0,84	0,733
1			0,539	0,072	0,624	0,741	0,733
2006	**81,788**	**917**	**0,726**	**0,274**	**0,747**	**0,79**	**0,726**
PSO_LinSVM for 70-89 ([-S, 5, -C, 19.936703011486063, -E, 1.0E-6, -B, -0.210483837986424])							
Year 2007	81,788	917					
0			0,932	0,48	0,881	0,835	0,726
1			0,52	0,068	0,613	0,746	0,726