VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Sprendimų medžių ir dirbtinių neuroninių tinklų pritaikymai finansų rinkose

Application of decision trees and artificial neural networks in financial markets

Magistro baigiamasis darbas

Atliko:	Saulius Blažiūnas	(parašas)
Darbo vadovas:	Doc. dr. Aistis Raudys	(parašas)
Recenzentas:	Partn. doc. Arūnas Janeliūnas	(parašas)

Vilnius – 2017

Santrauka

Šiame tyrime lyginama dirbtinio neuroninio tinklo ir C5.0 sprendimų medžio prognozės. Darbo tikslas yra nustatyti, kuris iš jų geriau tinka finansinių duomenų prognozavimui ir automatizuotų prekybos strategijų kūrimui. Siekiama gauti kuo geresnius testinės imties rezultatus. Tyrimams naudojama 45 likvidžiausius ateities sandorius pasitelkiant 30 populiariausių techninių indikatorių, kurie apskaičiuojami iš kainos ir prekybos apimčių duomenų. Išvados daromos iš 16 895 eksperimentų. Tyrimo rezultatai parodė, jog dirbtinio neuroninio tinklo ir C5.0 sprendimo medžio prognozavimo modeliai duoda gana panašius rezultatus, tiek pagal prognozės tikslumą, tiek pagal modelių pelningumą. Yra pasiūlomas jungtinis prognozės modelis, kuris naudoja tiek dirbtinio neuroninio tinklo, tiek sprendimų medžio modelių prognozes priimti galutiniam prognozės sprendimui. Testai rodo, jog jungtinis metodas yra geriausias.

Raktiniai žodžiai: dirbtinis neuroninis tinklas, sprendimų medis, C5.0, ateities sandoriai, finansinių duomenų prognozė

Abstract

This research compares the performance of artificial neural network against C5.0 decision tree performance. The aim is to see which one is more suitable for financial data prediction and automated trading strategy development. The evaluation is performed on out of sample/testing data. 45 most liquid futures of various financial sectors are used in simulations with 30 most popular technical indicators derived from price and volume data. Conclusions are made from 16,895 experiments. It has been shown that artificial neural network and C5.0 decision tree models have quite similar prediction accuracy and their profitability is similar. A combination of both artificial neural network and C5.0 decision tree prediction models has been proposed. Simulations shows that the combined method is the superior one.

Key words: artificial neural network, decision tree, C5.0, futures, financial data prediction

Contents

Introduction

Profitable, accurate financial time series forecasting methods are essential for portfolio management by financial advisors, investment funds and banks, commercial banks. It is common for financial markets to be uncertain and disrupted by the major events in the world. Lately, there were Brexit, ECB's QE program, OPEC deals to boost the oil price. Because of constant uncertainty, practitioners in finance are always in need of successful forecasting methods which would help to predict uncertain future both in long and short periods ahead. In recent years, more and more attention is given to systematic trading, when predictions are based on some kind of algorithms. In figure 1 a rapid growth of assets under management of systematic trading is shown.

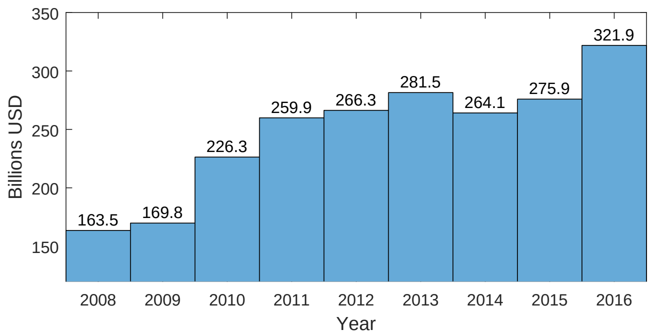

Figure 1. Rapid growth of systematic traders asset under management. Source [Bar].

A practitioner in finance faces a major problem, when he/she wants to exploit some of the forecasting methods covered in scientific literature - to determine what kind of prediction model is the best to use. It has been widely accepted by many studies that nonlinearity exists in financial markets and that artificial intelligence methods like decision tree (DT) and artificial neural network (ANN) are known to be effectively used to uncover non-linear relationships. This means that they might be successfully exploited to predict in financial markets domain. Hence, such methods are in spotlight of systematic traders lately. A reliable DT or ANN forecasting model can be of great assistance to practitioners.

The main purpose of this paper is to determine whether decision tree or artificial neural network is better to predict financial data, hence better to be used by a practitioner in the market. Currently there are numerous researches in a scientific literature available about ANN or DT improvements and various applications. However, to this day no detailed and extensive comparison of ANN and DT for financial data has been made. This paper tries to fill this gap by applying 45 future financial instruments which cover various types of asset classes. This research should be of help for practitioners to determine whether to choose ANN or DT.

A simple ANN (based on feed-forward and backpropagation with gradient descent) is used as there are no clear evidence that other, more advance ANNs are clearly better. From decision tree class, C5.0 algorithm is picked based on its popularity and researches that shows that C5.0 has one of the best classification accuracies in DT family.

Financial market is known to have a lot of noise and randomness. Therefore, this research uses large data samples to draw conclusions, as only results of tests with large data samples should be trusted. Daily data is chosen to be used with 11 years of historical data for every of 45 futures financial instruments (126,361 days in total).

One day ahead is forecasted using 30 common technical indicators. To make one day ahead forecast, a classification problem is constructed. The goal is to predict one of 3 outcomes for the next day: price will rise, price will fall, or price will not change significantly.

Studies focusing on forecasting the financial markets have been mostly preoccupied with reducing prediction errors based of MSE, RMSE, MAPE, MAD and similar measures. This research makes an effort to create profitable predictions models and portfolios, hence along with MSE statistics, prediction accuracy in percent, Sharpe ratio, and profit is used to determine whether ANN or DT is a superior prediction method. Model predictions are used to create a trading strategy so that the profitability of prediction models may be assessed.

More emphasis will be made on empirical research rather than theoretical.

The organization of this paper is as follows. In section 1 an overview of related researches about neural networks and decision trees is made. In section 2 the data which is used in this research is described. Section 3 explains what kind of trading strategy is being used and how ANN and DT realize that trading strategy. Tests results for data samples selection for training, validation, and testing data are given in section 4. In section 5 ANN properties are optimized (epoch number, learning rate, momentum rate, hidden neurons count, outputs count, inputs subset selection method). DT parameter optimization is described in section 6 (severity of pruning, minimum number of cases to follow at least two branches, errors cost). In section 7 results of ANN and DT are compared. Section 8 presents the experiments of ANN and DT combinations to create better prediction models. In section 9 an example is given about successful application of filter to select better prediction models from all prediction models set. Finally, in section 10 conclusions are made. In the end of the paper are explanations of abbreviations and appendices.

1. Literature overview

1.1. Artificial neural network

An idea of artificial neural networks is not new and it has been already widely exploited for various kind of problems. The main concept of a neural network was developed by neurophysiologist Warren McCulloch and mathematician Walter Pitts who modelled a simple neural network using electrical circuits [MP43]. In the 1950s and 1960s scientist Frank Rosenblatt who was inspired by the latter two, developed an idea of perceptron on which the idea of artificial neural network is based on [Ros58]. This, at a time completely new method did not get a lot of attention because of it lacking possibility to learn. In 1974 Werbos discovered backpropagation algorithm (algorithm which allows ANN to learn) and only after it was rediscovered in 1986 [RHW85] it gained worldwide attention. Since then over the last three decades artificial neural networks have been successfully used for data analysis, decision making and forecasting in various types of fields.

1.1.1. Neural network properties

The most commonly used type of neural network is a multilayer perceptron network which uses backpropagation algorithm to optimize the network. In a survey of Velido et al. [VLV99] they have found that backpropagation model was used in 74 out of 92 papers that they analysed (these papers covered various topics to which ANN was applied). A structure of such network is shown in a figure 2.

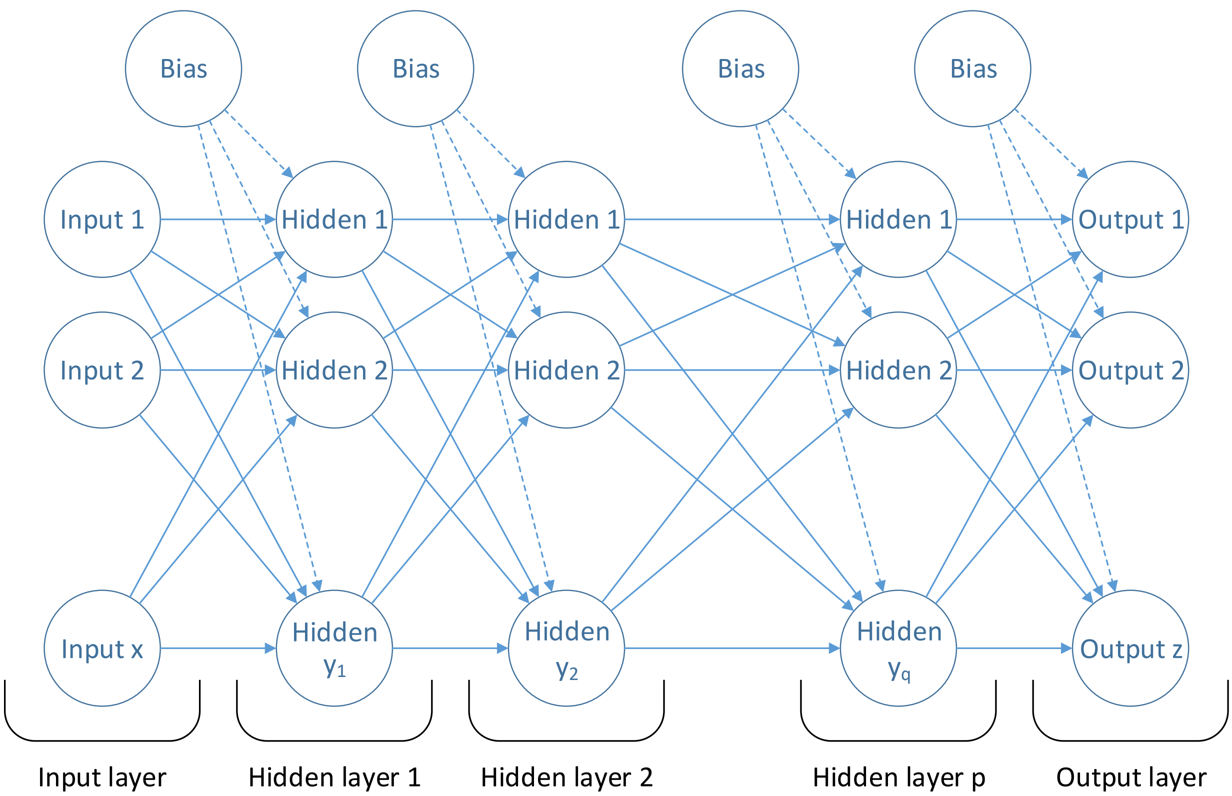

Figure 2. Abstract structure of a multilayer perceptron.

This network consists of three layers:

- Input layer - here input data are fed to the network. Data selection is an intricate and complex task. A neural network is only as good as the data which is input to the network. If some important data inputs are missing or even if the data is not carefully and appropriately prepared it can have significant impact on the neural network performance (capability to classify, predict, forecast). A study made by Cheng-Wen Ko and Hsiao-Wen Chung [KC00] analysed an effect of data preparation. They analysed EEG (Electroencephalography) data to detect epileptic spikes (epilepsy is a group of neurological diseases). Based on their study it can be believed that falsely high correct classification rate (good results by applying ANN) in tests can arise from erroneous data preparation.
- Hidden layer - in a general form hidden layer can contain multiple layers as shown in figure 2. The number of hidden layers which should be used depends on a specific task. There can be even empty hidden layer (for example Self Organizing Map networks [KSP01]).
- Output layer - outputs results of the network. Numerical values which are used to classify data or to get needed transformation of the input data (for example - a forecast).

Table 1. Parameters which are usually determined in designing a neural network with backpropagation algorithm. Categories of the table are taken from [KB96].

Data preprocessing	
frequency of data - daily, weekly, monthly, quarterly	daily
type of data - technical, fundamental	technical
method of data scaling	scale to interval $[-1,1]$
Training	
learning rate	0.2
momentum term	0.7
epochs size	pass all data once
maximum numbers of runs/epochs	750
size of training set	70%
size of validation set	15%
size of testing set	15%
Topology	
number of input neurons	30
number of hidden layers	1
number of hidden neurons in each hidden layer	10
number of output neurons	3
transfer function for each neuron	sigmoid
error function	MSE

A network can learn by changing weights of the connections between neurons. Backpropagation algorithm is one of the most widely used tools to learn neural networks by changing weights between neurons. During this research backpropagation algorithm is going to be used.

In a table 1 parameters which are usually determined in designing a neural networks which uses backpropagation algorithm is presented. Parameters listed in this table are analysed in sections 1.1.2 and 1.2, optimized in section 5.

Another important property of a neural network which must be chosen is a learning type. Machine learning algorithms including neural network algorithm can be grouped to two groups: algorithms which use supervised or unsupervised learning. For this papers problem supervised method is chosen. That is given a set of n data in a form $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ where x_i is a feature vector, $i \in \{1,2,\ldots,n\}$, y_i is its goal that neural network wants to achieve. A neural network seeks a function:

$$g : X \to Y,$$

where X is an input vector ($X = (x_1,x_2,\ldots,x_n)$) and Y is an output vector ($Y = (y_1,y_2,\ldots,y_n)$).

If it is assumed that neural network outputs value $\hat{y}$ then the task is to minimise difference between $\hat{y}$ and y_i - it is tried to adapt a neural network so that actual network outputs comes close to targeted outputs. A supervised learning method fits better to this paper problem because it has a clear goal that it wants to achieve (y_i is going to be determined as the idea of trading strategy optimization requires it).

1.1.2. Feed forward and backpropagation algorithms

A thorough explanation about a whole backpropagation algorithm used in this research can be found in a paper of Paul J. Werbos [Wer90].

In this paper sigmoidal transfer function is used as it is one of the most popular activation functions for neural networks:

$$s(r) = \frac{1}{1 + e^{-r}}$$

It is desirable to limit the connections between different neurons, because it reduces overfitting. Hence a layer based neural network is chosen to be used in this research where every layer is fully connected to a later layer. An example of connections which are used in this paper can be seen in figure 2. So, applied neural network will contain input layer, output layer and the hidden layer where every neuron in a layer is connected to every neuron from a preceding layer and to bias neuron - the one which serves in a similar purpose as the intercept in a regression model.

In order to start learning algorithm by feeding forward neural network starting values of connections weights must be chosen. That this is a difficult task can be seen from Kolen and Pollack paper [Pol90]. In their research they showed that initial weights set (starting weights) are very significant parameters in convergence time variability. They also showed that it is not possible to implement a global search in order to obtain the optimal (best) set of neural network weights meaning that an optimization technique such as backpropagation algorithm should be used to find a locally optimal neural network weights. The local optimum which is found after execution of neural network with backpropagation algorithm depends on the starting parameters - starting weights. This means that both convergence rate and the quality of the solution which is obtained are influenced by starting weights.

A research of de Castro and Von Zuben [CV01] have addressed this problem. They present an algorithm called SAND (Simulated Annealing for Diversity) which finds starting weights of ANN. They have shown that their suggested algorithm is superior to BOER [BK92], WIDROW [Ngu90] and KIM [KR91] algorithms and is similar (in a sense of having good convergence and finding good local optimum) to INIT [CV98] and OLS [LSK+95] algorithms. The authors point out that the advantage of SAND algorithm is that it does not use trading data to estimate starting weights - contrary to INIT and OLS algorithms. SAND algorithm is chosen to be used to find starting weights because of good SAND algorithms performance and its simplicity.

The idea is that weights of neurons in a layer should be as diversified as possible. This is applied to every layer separately. Suppose that in n'th layer there are 5 neurons and all of them have 10 nodes coming to them. It means that weights of connections to these neurons is 10 dimensions vectors $w_i = (w_{i1}, w_{i2}, \ldots, w_{i10})$. The goal is to distribute these 5 weight vectors in 10-dimension space so that these 5 points in hypersphere would distribute in a way that difference between these points would be as big as possible. As suggested in Castro's and Zuben's paper Euclidean difference will be used to estimate how diversified these points are. So it is needed to maximize sum of Euclidean differences as shown in equation 1.

$$ED = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sqrt{\sum_{k=1}^{d} (w_{ik} - w_{jk})^2}, \tag{1}$$

where d is dimensionality of weight vector w_i and N is a number of neurons in n'th layer.

ED can be maximized using various algorithms. Authors uses simulated annealing algorithm, which should be used in general, however as neural networks which will be used in this research do not have very high dimensionality of weight vectors it is chosen to use a brute force to generate good enough initial vectors because it is easier to implement and should quickly give decent values.

In this research used neural network backpropagation algorithm chooses ANN weights w_{ij} so as to minimize difference between ANN output $\hat{Y}$ using feed forward algorithm and desired target Y. Least square method (popular, widely used in ANN) is chosen to be used to measure difference between them. Difference (error) is calculated as shown in equation 2. T defines training set length.

$$E = \sum_{t=1}^{T} E(t) = \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{2} (\hat{Y}_i(t) - Y_i(t))^2 \tag{2}$$

After feed forward and error calculations all ANN is updated as shown in equation 3. Here LR defines a learning rate which is a small constant and MC defines momentum constant.

$$New \quad w_{ij} = MC * w_{ij} - LR * \frac{\partial E}{\partial w_{ij}} \tag{3}$$

Detailed equations of derivatives calculations can be found in a paper of Paul J. Werbos [Wer90].

One more extremely important property of ANN is a number of hidden layers and neurons in every layer (obviously input layer depends on the data to be used and output layer depends on the question which is to be answered using ANN). It is very difficult to determine the best ANN structure - many tests must be made. That is why it is extremely useful to analyse already made researches. An overview of other researches is presented in section 1.2. Based on already made applications of ANN using financial data a most common and promising layout of neural network will be chosen to be exploited in this research.

1.2. Discussion about artificial neural networks in financial markets

ANN is a universal function approximation which can map any non-linear function [Whi89]. That means that ANN is a powerful method for pattern recognition, classification problems and forecasts. Financial markets usually have characteristics that it contains a lot of noise, contains chaotic components and data distribution is heavy-tailed. ANN can tolerate these problems better than most other methods for data forecast [Mas93].

Analysis of already made researches about ANN application in financial markets gives a few contributions to this research. First of all, it gives an idea about which neural network structure (layer and neurons number) looks the most promising and should be used. Also it gives an idea about what type of data (how formatted, transformed) and what financial data in general should be used for ANN to create profitable trading strategies. Most researches uses different approaches to create a trading strategy using ANN so an attention will also be given to the ways researchers exploit ANN.

A neural network outputs some numbers. One of the suggestions is to use intervals to make buy or sell orders out of them [KB96]. For example, all outputs (forecasts) greater than 0.8 can be converted to buy orders and all outputs less than 0.2 to sell orders. A trader must keep in mind that in this scenario ANN would reduce forecast error (buy or sell signal error) but it would not depend on trading profit. A single large trade which is forecasted incorrectly (small error from ANN stand of view) could be in fault of losing all traders money (huge error from traders stand of view).

ANN application in finances as well as any other field deals with a problem to select appropriate parameters of ANN. Researchers often overlook the effect of ANN parameters on the performance of ANN forecasting. A paper of Zang and Hu [ZH98] examined the effect of the number of inputs and hidden neurons as well as the size of testing and training data samples to be used. For their experiments they used weekly British pound to US dollar exchange rates. They found that ANN outperformed linear models in forecasting.

According to Zang and Hu, the number of input and hidden neurons are two critical parameters in the design of a ANN (of ANN itself leaving out a trading strategy). For output they used only one neuron as their forecast is only one step ahead. As inputs they used level (lags) of exchange rates up to level 10. Based on out of sample prediction errors measured in RMSE, MAE, MAPE they stated that 6 inputs are an optimal option. For 6 input neurons ANN a number of hidden neurons to be used did not show any clear indication of an optimal number of hidden neurons in a sense of out of sample prediction error. This means that the number of input neurons might have more significance than the number of hidden neurons.

When deciding how many hidden layers should be used it is common to use a rule of thumb to obtain a good generalization from a ANN - it is to use the system which is the smallest and fits the data [Ree93]. Kaastra and Boyd states that „all networks should start with preferably one or at most two layers“ [KB96]. When a number of hidden layers is increased then the danger of over fitting increases as well. They suggest that if ANN which have two hidden layers is unsatisfactory, then a researcher should modify the input variables multiple times before adding a third hidden layer to ANN. According to them most empirical work suggests that networks with more than four layers (one input, two hidden and one output) will not improve the results. A reader must take into account that this discussion covers a topic of trading where only several technical or fundamental indicators are used, therefore ANN solves different problems than deep neural networks which uses many hidden layers for analysis of very complex data (for example see [HDY+12] where it writes about success of using deep neural networks in speech recognition).

Kaastra and Boyd also points out that the number of neurons in the output layer (or the number of output neurons) is a straightforward decision. A neural network which will be used in this paper, learns by choosing weights so that the difference of output results and desired results (error) would be minimized. Suppose that ANN has two output neurons where one of them is for 1-day price forecast and the other for one-month price forecast. Authors stress out that in such ANN the algorithm will concentrate on reducing difference (error) of the output neuron which has a larger error (obviously one moth forecast will have much higher error that one-day forecast). As a result, a large improvement of one-day forecast will not be made if it will increase an error of one moth forecast by a greater amount than it reduces error for one-day forecast. Therefore, such ANN should not be used. A better way would be to make two ANN for different forecasts with both of them having just one output neuron.

Shanker et al. [SHH96] made a research to see whether linear or statistical normalization methods to prepare input data to ANN increases performance. They found that in general it does, however significance diminishes as network and sample size becomes large. On Zang and Hu [ZH98] experiment in financial data they find no significant difference between normalized and raw data because weights on ANN does a scaling automatically. However, they did use natural logarithmic transformation to stabilize the time series. In this research SAND algorithm is used to select initial weights for a ANN in order to increase performance, therefore it should be better to normalize data as so that it would be more suitable for weights given from SAND algorithm.

A study of Olson and Mossman [OM03] shows that ANN can forecast better (better in a case of more profitable trading strategy) than OLS and logistic regression techniques. They forecasted

Canadian stock returns to one year ahead. They used data set of 61 annual accounting ratios and financial variables for every firm traded on the Toronto Stock Exchange (TSE) over an eighteen-year period - current ratio, annual % change in current ratio, annual % in net income to opening equity ratio and so on. Though they only forecasted data of 2352 out of 4750 companies as they used few restrictions to select only usable and representative data. They used walk-forward method using six-year window to forecast next 12 years by moving this window. Such method is preferable if enough data is available. Financial markets tend to change over time so the data which is 18 years old might not be useful for forecast. By using walk forward method a researcher can look at how his trading strategy would have performed over the years. ANN which they applied contained only one hidden layer for every walk-forward calculation. To select neurons count they used as they say „a rule commonly used by other researchers“ where the number of hidden neurons is the sum of inputs plus outputs and all divided by two. Though this is a problem of discussion as for example Salchenberger et al. [SHH93] states that a single hidden layer should contain 75% of input layer neurons number. Out of 61 input variables they randomly selected 30 of them as 61 might be too much. As a measure to compare different forecasting models results they used directional measure - what percentage of stocks did they forecasted to correct direction (direction can be that price will fall or rise) and trading strategy profit in percentage of investment. In their study ANN outperforms both OLS and logistic regression based on these measures.

Guresen et al. [GKD11] forecasted NASDAQ stock exchange index using ANN, dynamic artificial neural network and a hybrid method combining ANN with GARCH model. They used daily data and made one-day forecast. They compared results based on forecast error - MSE, RMSE, MAPE and MAD statistics. The overall results showed that classical ANN gives the best results - hybrid methods which where compared did not improve the forecast. This result should not be taken for granted, however it shows that it is best to start from a simple ANN when doing a research. For ANN learning algorithm they use cross validation method to control algorithm's running time (iterations count). In researches three stopping criterions are the most popular: fixed iteration count, threshold of ANN output's error, and cross validation. Cross validation is the best because it stops at the point when the best performance in the test set is obtained [PEL99].

Wang [Wan09] forecasted Taiwan stock index option price. The research focuses more to what inputs should be given to ANN that to ANN itself (focuses to price volatility as input). Volatility is one of the main characteristics on which many trading strategies are based. Forecast of the volatility is not a trading strategy by itself but a trader who could evaluate future volatility correctly could profit from it easily. Few variations of GARCH models were used to forecast volatility. Then the forecasted value was given to the ANN along with a strike price, current underlying asset price, short term risk free interest rate, time to maturity. So both economic data and technical indicators were used. This research used ANN with one hidden layer.

Trippi and Desieno [TD92] created ANN based trading system for Standard and Poor 500 index futures which outperformed a passive investment in the index. Their trading model gave recommendation (to go long or sell short) for the current day and it also included a trailing stop to limit losses. For ANN input they used index data and technical indicators for the two-week period prior to the trading day (open, high, low, close prices, volatility and similar). In total they trained 6 neural networks and made a composite trading rule from them. Difference between neural networks where slightly different inputs to some of ANN and different random initial weights vectors. Sixth ANN was purposely over trained to get much smaller error than the other 5 ANN as they assumed it could help to increase performance of the composite rule. Then a composite rule was created based on outputs of all 6 trained networks. Example: if all networks agrees, then make the indicated trade. If network 1 and network 2 agrees on order direction (go long or sell short) and one of networks 3 to 5 disagree, then follow network 1 decision. An explanation of how they created composite rules is given in their paper. They showed that their model performed better (in testing data sample) than any of those six neural networks separately and also outperformed buy and hold strategy of the index. It is worth mentioning that they used only 106 days (data was daily) for trading data and 39 days for testing therefore given results should not be trusted as it lacks thorough testing. However, an idea of applying neural network and then using decision rules to make final decisions looks promising.

O'Connor and Madden [OM06] applied ANN to forecast Dow Jones Industrial Average index daily prices using external factors rather than technical indicators of this index. Daily data was used and 9-day period prior to trading day to make a trading decision. Their research concluded that ANN trading model was superior trading strategy to buy and hold strategy for a bear and stable market. It must be mentioned that a benchmark trading strategy they used is not appropriate on a bear market so this trading strategy lacks suitable benchmark for comparison. For external factors they used oil prices and currency rates which are important to some of the companies of which stocks index is made of. For internal data they used spot index values. In order to select only important inputs, they grew their ANN iteratively. At first they implemented ANN with a few variables and then added another ones and used them for further iterations if they improved performance. This approach to select important inputs looks promising, however it should not be forgotten that more inputs results in higher over fitting (over-optimization of ANN) in training sample, therefore worse results in testing sample.

They stressed out that external factors such as Federal Reserve rates is announced quarterly so it is difficult to apply this data to ANN which uses daily data. They point out that interpolating data is not suitable as a future announcements are not known and also step-wise representation where value is a constant at all times until a new announcement should not be used because announcements tend to cause short-term changes in the index. They also added that rate change itself does not hugely effect the earnings of the companies, but rather effect how traders make their expectations based on rate change. Therefore, they exclude this rather important external factor from ANN even though it sometimes has a huge impact on daily performance of the index. O'Connor's and Madden's reasoning seems legit however one could notice here a common thing in most of researches concerning trading strategies that researchers sometimes does not have enough insights on the actual trading world and are missing some important information. In this particular example one could argue with their reasoning that the rate data is not as important as it effects more traders' expectations that the companies that this index is based on. On short term trading (daily trading - in their paper they apply daily trading strategy) traders' expectations have a significant impact on the price. Furthermore, they do not even consider such thing as CME FedWatch tool [CME]. Based on CME Group 30-Day Fed Fund futures prices, which have long been used to express the market's views on the likelihood of changes in U.S. monetary policy, the CME Group FedWatch tool allows market participants to view the probability of an upcoming Fed Rate hike. Mostly only real practitioners of trading know about such indicators, therefore they are usually missed out from research papers which are done by scientists. Such indicators should be at least considered if not tested when external, fundamental data is used.

Their research also implies that machine learning algorithms would be more successful in trading if they sought to optimize profitability of trading strategy rather than statistical error like RMSE, MSE and similar.

Zhang [Zha03] forecasted British pound to US dollar exchange rate using weekly data. In the paper it is shown that better results could be achieved combining ARIMA and ANN to forecast prices rather than using one of them. Reasoning to combine these methods are that ARIMA can capture linear component and ANN - nonlinear. Suppose that the price y_t contains linear component L_t and nonlinear component NL_t ($y_t = L_t + NL_t$). At first ARIMA can be used to capture $\hat{L_t}$ and then the remaining part (residuals - assumed as nonlinear component) $y_t - \hat{L_t}$ can be captured by ANN. An ANN with 7 inputs neurons, 6 or 5 hidden neurons in a single hidden layer and 1 output neuron networks were applied. ANN and ARIMA combination gave a better forecast in a sense of statistical measures MSE and MAD.

Hamm and Brorsen [HW00] applied ANN to create trading strategy for hard red winter wheat and Deutsche Mark (German mark) futures. For inputs lagged logarithmic values of prices change $r_t = \ln(\frac{P_t}{P_t - 1})$ were used. Data was of weekly frequency. Last 8-week data were used as inputs to ANN (8 input neurons) to Deutsche Mark futures and additional 3 for wheat futures which were seasonal dummy variables because wheat is a seasonal commodity. Tested numbers of hidden neurons in a single hidden layer were 4, 6, 8, and 10. Trading signals were generated as ”long” if forecast $r_{t+1} \geqslant 0$ and ”short sell” otherwise. Hamm and Brorsen highlights an importance of using testing data sample results to compare different models. Therefore, training data is used to train ANN and testing data is split to two data sets - one is used to compare different tested models to select the best model (by doing so this data set becomes validation data) and the other as the real testing sample data set. In their research no significantly profitable trading models were found. However, it does not mean that ANN sometimes does not work in forecasting financial data. Bad performance can be a result of bad ANN parameters or structure, wrong or incomplete input set, wrong interpretation of outputs or data which is unpredictable.

Leig et al. [LPR02] forecasted the NYSE composite index (stock index). To get inputs for ANN they used pattern recognition technique of template matching (bull flag - more information in [LPR02]). 10 values of both price and volume from template matching algorithm were used as inputs. In addition, 2 window height values (difference between lowest and highest point in n days window) were used as inputs. So in total 22 input neurons were used, single hidden layer contained 6 neurons and output contained 1 neuron. All inputs were made from 60-day window and the forecast was for the following 20 days. Data was normalized using z-score algorithm (to get unit variance and zero mean). It is shown that such model can successfully predict prices (meaning prices does not follow a random walk). Even better results were achieved by using genetic algorithm to select only some inputs out of 22. Coefficient of determination R^2 was used to compare accuracy between different trading models.

Quah and Srinivasan [QS99] applied ANN to select stocks which outperforms market in the following quarter. For every stock 7 ANN quarterly data inputs were used: historical and perspective P/E ratios, cash flow yield, market capitalization, earnings per share, return on equity, average of the price appreciation. 4 to 14 hidden neurons were used. Top 25 stocks which predicted to outperform market the most were added to the portfolio. This portfolio outperformed the market in 6 out of 8 quarters. Quah and Srinivasan stresses out that different timelines in market might have a significant impact to machine learning algorithm performance.

Qiu and Akagi [QSA16] used ANN to forecast return of the Nikkei 225 index (one of the main Japan stock indexes). 71 monthly frequency variables that include financial indicators and macroeconomic data were used. In order to select the important ones, fuzzy algorithm [LCC96] was used - it selected 18 inputs. All inputs were normalized to range from 0 to 1. Desired output was the monthly return of index. As done in numerous researches the parameters of ANN are determined by empirical testing. Tests were done 900 times to find best parameters in a sense of predictability accuracy measured with MSE. 10 hidden neurons, 3000 iterations, 0.4 as momentum constant and 0.1 as learning rate was found as the best parameters set. To select initial weights genetic algorithm and simulated annealing algorithm were applied. Qiu and Akagi research indicates that it is best to use genetic algorithm to get initial weights and then run ANN with backpropagation algorithm to get the best weights set.

A summary of analysed papers which did some empirical researches and did some comparison of results is presented in table 2.

When analysing researches that are already made it can be clearly seen that these researches are made by scientists rather than traders who actually uses trading strategies created by ANN to trade in actual market. All these researches gives only simulated results. Some of them do not even calculate simulated trading performance and are concentrated only in reducing forecasting error. I have not seen any real trading results given in any of the researches about ANN in finance.

Most of researches are very conservative on data that they are using - most uses price, volatility,

Table 2. Summary of ANN studies where empirical research and some comparison is done.

Study	Data used	Data frequency	Inputs used	Prediction period	Data transformation	Number of hidden neurons	How many neurons in a hidden layer	Comparison measures	Additional contribution to this research
Zang and Hu [ZH98]	British pound to US dollar exchange rates	Weekly	Levels	1 day ahead	Natural logarithmic transformation	1	Tested very wide range. No optimal selected.	RMSE, MAE, MAPE	Shows that input size is more important than hidden neurons count. More anticipation will be made for inputs selection.
Olson and Mossman [OM03]	2352 companies' stocks listed in Toronto Stock Exchange	Annual	61 annual accounting ratios and financial variables	1 year ahead	Not mentioned	1	50% of inputs + outputs	Prediction accuracy in percent and ROI	Gives a good example of portfolio construction from 2352 predition models.
Guresen et al. [GKD11]	NASDAQ stock exchange index	Daily	Levels	1 day ahead	Not mentioned	Not mentioned	Not mentioned	MSE, RMSE, MAPE, MAD	Shows that simple ANN should be started with then doing a research before trying more complicated ANNs.
Wang [Wan09]	Taiwan stock index option	Mixed	Volatility, price, and financial variables	Mixed	Not mentioned	1	Not mentioned	RMSE, MAE, MAPE	Shows the importance of using volatility as an input.
Trippi and Desieno [TD92]	S&P 500 stock index	Daily	Simplest technical indicators	1 day ahead	Not mentioned	Not mentioned	Not mentioned	Prediction accuracy in percent and ROI	Shows that combination of ANN and decision rules can increase performance. However, very small training and sample sizes were tested, hence results might be misleading.
O'Connor and Madden [OM06]	Dow Jones Industrial Average index	Daily	Levels and external prices (oil prices, currency rates)	1 day ahead	Not mentioned	2	60% to 120% of inputs	RMSE, ROI	Their research implies that prediction models creation would be more successful if profitability would be considered as performance statistic rather than RMSE, MSE, and similar.
Zhang [Zha03]	British pound to US dollar exchange rates	Weekly	Levels	One step ahead (35 and 67 weeks - two different ANN)	Natural logarithmic transformation	1	70% of inputs	MSE, MAD	Shows that combination of ANN and ARIMA gives better forecast. It implies that various combinations of ANN and other methods might give better results.
Hamm and Brorsen [HW00]	Hard red winter Wheat and Deutsche Mark futures	Weekly	Levels	1 week ahead	Natural logarithmic transformation. Scaled to interval $[-1,1]$	1	35% to 100% of inputs	ROI	Gives an example about importance of using training, validation, and testing data samples.
Leig et al. [LPR02]	NYSE composite stock index	Daily	Inputs obtained from pattern recognition technique of template matching	20 days ahead	z-score normalization	1	30% of inputs	Coefficient of determination R^2, ROI	Shows that genetic algorithm implementation to select only some of the inputs to ANN can increase performance.
Quah and Srinivasan [QS99]	Various stocks	Quarterly	Economic data about stock	1 quarter ahead	Not mentioned	1	60% to 200% of inputs	Portfolio ROI	Stresses out the importance of specific market conditions (bullish, bearish market, crisis period) to training, testing data results.
Qiu and Akagi [QSA16]	Nikkei 225 stock index	Monthly	Financial indicators and macroeconomic data	1 month ahead	Scaled to interval $[0,1]$	1	55% to 555% of inputs	MSE	Gives a good example of what parameters and in what range should be optimized.

profit ratios and similar if prediction is for stocks/stock indexes, high-low window, lags (levels) of all these values. Almost none of more advanced technical indicators are applied which are widely used in real trading (moving average convergence divergence, relative strength index, on balance volume, average directional index, Williams %R). Majority of researches takes only one financial instrument for analysis (exception is for stocks - sometimes even a few thousands of stocks are used). None of the analysed researches tries to apply their models to variety of different financial instruments. Even though these researches are considered empirical, they actually do not test their ANN trading strategies as real traders would.

In this paper a broader approach is done where many commodities, interest rates, currency exchange rates, bonds, various indexes are used as data. Various popular trading indicators are applied.

Other researches does help a lot to make an educated guess of what ANN set of parameters should be empirically tested to get the optimal ones. It can be clearly seen that there is no consensus in almost every aspect and parameter of ANN. But no one of the researchers analysed used more than 18 inputs (after selection from all inputs set). It seems that the most widely used methods to select input variables are common sense and genetic algorithms - it will be used in this research. Input data is very different in all researches but the most common are levels of the data which is forecasted and trading volume. In addition to these inputs, in this research various technical indicators made from price and volume will be used as it should increase ANN performance (if not, then it will be seen from inputs selection method results). Most researches sticks just to one hidden layer with a number of hidden neurons usually smaller than input neurons. So only one hidden layer will be used with a number of hidden neurons from a few to as much as number of input neurons (as seen in table 2 typically 35% to 100% of input neurons count is used as hidden neurons count in reviewed researches).

Two approaches seem to be the most popular to convert outputs to a trading strategy. One is to split output to less or more than something and buy or sell short accordingly (example: buy if output more than zero and sell short otherwise). Another is to forecast future price itself, forecast percentage of rise/fall of price or forecast excess return. Both of them seems to be appropriate. In this paper first approach will be used as it is more suitable for trading strategies.

It can be seen in overview table 2 that measures used in other researches to compare forecasting models are MSE, RMSE, MAD, MAE statistics, percentage of correct forecasts and trading profits (ROI). For ANN training MSE will be used. It is widely used statistic and does not have significant difference from others (MAE, RMSE, MAD). To determine prediction models parameter MSE and Sharpe ratio will be used. Sharpe ratio is popular measure of performance used by traders. A simple Sharpe ratio definition is: ratio of earned profit and the risk of investment. More information can be found in paper [Sha94]. To compare ANN and DT performance where is no point in using statistics like MSE, RMSE, MAD, MAE because errors are defined in different ways. Hence, other methods used in reviewed researches will be used to compare: percentage of correct forecasts, trading profits, Sharpe ratios.

Some of researchers use just raw data and some use normalized, transformed data. The benefit of preparing the data is arguable, however it is clear that it will not be worse if normalized data is used. In addition to that, SAND algorithm to select initial weights is chosen. These weights should work better for normalized data so normalization will be done (there seems to be no evidence of worse performance after data transformation/normalization).

Data frequency is another important parameter to be considered. Daily, weekly, monthly, quarterly and annually data is usually used as inputs. Even though a trader can aggregate higher frequency data to lower, the opposite should not be done as explained by [OM06]. This paper focuses more on higher frequency data, therefore daily data will be used as the highest popular frequency of data (the reason for the lack of higher frequency data usage in researches might be that most researchers do not have data of higher frequency than daily). Usually for daily frequency most of fundamental values are not suitable because of too low frequency. Because of that technical indicators are chosen to be used and fundamental indicators will be left out.

1.3. C5.0 decision tree

Most statistical methods require data to be normally distributed and independent. The superiority of decision trees is that it does not require such limiting assumptions which are rarely met in financial field. In addition, decision trees are easily interpreted, draws understandable graphical displays - it is a transparent classification and forecasting method. Even though decision trees are most commonly used in bioinformatics, medicine field, it can also be applied for financial markets forecast.

A decision tree is a flow chart just like a tree structure. Each internal node denotes a single test on an attribute. Each branch which comes from internal node represents an outcome of the test made in internal node. Each leaf node holds a class label - the decision.

A decision tree has two phases [HKP11]:

- Tree growth phase. The tree is built by recursively splitting the given training data set based on local optimal criterion until most or all of the records belonging in a specific class are classified correctly. During this phase the tree may over fit given data.

- Pruning phase. This phase handles the problem of tree over fit to the training data set. In this phase the tree is pruned (some nodes removed). The accuracy of tree for testing data sample is increased in this phase.

There are numerous of variations of decision trees which differs both in tree growing and tree pruning algorithms. In appendix table A2 [AS09] it can be seen that C4.5, ID3, and CART algorithms are the most popular. In addition, Delen et al. [DKU13] indicates ID3, C4.5, C5.0, CART, and CHAID and QUEST algorithms as the most commonly used ones. They concluded that C5.0 and CHAID decision tree algorithms gave the best prediction accuracy of forecasting Turkish companies' performance.

In short, C5.0 is a better and more advanced version of the most popular algorithms ID3 and C4.5. Also C5.0 algorithm usually gives one of the best accuracy of all popular decision tree, therefore it is chosen to use this decision tree in this paper as it should be one of the most representative algorithms of decision trees family.

C5.0 starts from ID3 - a supervised decision tree. A good description of ID3 algorithms is [Qui86]. This algorithm is based on the theory of information gain - information gain is determined by entropy. However, using information gain as a rule to select attributes for segmentation will result in bias over attributes of higher values [CC09]. To improve this drawback, C4.5 was introduced. This algorithm removed the biasness of information gain [Qui93]. C5.0 offered additional improvements to C4.5. It is significantly faster, more memory efficient, creates considerably smaller decision trees while giving similar results [DKU13]. Also C5.0 allows to weight different attributes that quantifies the importance of each case. This is extremely useful in finance as usually a mistake of predicting a low increase in price instead of true high increase in price is much better error than predicting small increase while the price actually slightly decreased. Weights allow more flexible errors so they will be tested in this research.

1.4. Discussion about C4.5/C5.0 in financial markets and other fields

Since C5.0 is the same as C4.5 just with some improvements, both C5.0 and C4.5 decision trees applications can be analysed. There are almost no scientific researches made which would apply these decision trees to forecast financial times series in order to make a profitable trading strategy. However, there are numerous researches in medicine, bioinformatics fields which shows that these decision trees work well for classification and predictability, therefore should also work with a financial data.

Kretschmann et al. [KFA01] successfully applied C4.5 algorithms and created over 11 000 rules which could be applied to unannotated (statement of an aspect that is relevant to describe the protein) protein sequences as the quantity of new protein data exceeds possibilities to manually annotate data. This research shows that decision trees might be suitable for extremely large set of financial instruments (ex. thousands of stocks) to be used to select a few most promising ones for trading.

An advantage of using decision trees against human experts is noted in Sebban et al. research [SMR+02]. They used C4.5 decision tree to generate knowledge rules for solving classification task from spoligotyping (a technique for the identification and analysis of polymorphism in certain types of repeat units in DNA) data. The rules which were generated by decision trees were simpler than those previously defines by the human expert. Decision trees uses tree pruning phase where noisy parameters are removed thus decision tree focuses only on main parameters whereas an expert takes into account all data as signals. In finance this is extremely important as there are lots of noise and not every parameter and indicator is relevant.

One of the few existing papers which used C4.5 or C5.0 decision tree to create a trading strategy is made by Wu et al. [WLL06]. They clustered trading points to favourable and unfavourable. Different thresholds were tested to classify trading point as favourable or not. 1-3% thresholds yielded best results in their tests. Trading points which were given to decision tree algorithm were at first filtered - only specific trading points were selected. There are numerous ways to filter data, however it is not a necessity as a trader might want to have a trading decision for every day. The task was to trade electronic Taiwan and NASDAQ technologies stocks. Only four parameters were used to generate decision trees which were derived from money supply, inflation rate, revenues and price of stock index futures data. From 39% up to 66% of trading points were filtered and classified correctly.

1.5. C4.5/C5.0 and neural networks combination

In scientific literature it is common method of combining decision trees with artificial neural networks to achieve better results. Chang and Chen [CC09] combined C4.5 decision tree with ANN to construct the best predictive model for six major skin diseases recognition. In their research stand-alone ANN gave best predictive accuracy (92.62%). Decision tree also was correct 90.89% so the difference is not very significant. However, combined model which used nodes produced from decision tree as inputs for ANN gave just 86.86% accuracy. This does not mean that combination of these two algorithms will always give worse results as there are different ways to combine them. This research also gives a good example of how important inputs are. ANN which used decision tree nodes as inputs gave almost 6% less accuracy than the simple ANN. The difference between them were just different inputs. This means that the ability of a researcher to prepare best data to work with affect algorithms results.

Zhou and Jiang [ZJ04] combined decision trees with ANN in opposite way - they proposed algorithm NeC4.5. At first neural network was trained on all data and outputs set was obtained. Then this output set was enlarged by generating some random input vectors and feeding them to ANN. Lastly, output set was used as input for C4.5 decision tree algorithm. The idea why NeC4.5 might be better than C4.5 is that original training set contains much noise and might not capture the whole target distribution. Twenty various data sets were used to compare C4.5 results with NeC4.5. It was shown that NeC4.5 could show better results than C4.5 if correct amount of decision tree input set enlargement is made, however it is a problem of how to determine the best amount.

Another approach to combine ANN with decision tree is to use both of these algorithms where they are the strongest. Pan et al. [PCH+03] used hybrid model to detect intrusion - deals with network security problem. They tried to detect DOS, Probing, U2L and R2L attacks. They concluded that ANN have high performance of detecting DOS and Probing attacks. On contracts, C4.5 decision tree detected R2L and U2R attacks more accurately than ANN. Based on these results a hybrid model was made. If ANN detected DOS or Probe attack then this hybrid model would stop, however if it was not detected or either R2L or U2L attack was detected, then C4.5 algorithm was used which took inputs both from initial ANN inputs set and from ANN outputs. Such hybrid model outperformed both ANN and C4.5.

Similar hybrid models were tested by Tsai and Wang [TW09] for Taiwan stocks price forecast. One of models trained ANN and then used only correct outputs of ANN for CART decision tree. Another hybrid model was to apply decision tree and then use correct outputs for a second decision tree. Decision tree hybrid with ANN gave 77.20% accuracy, decision tree with another decision tree - 66.85%, single decision tree 65.42%, single ANN 59.02%.

It can be seen that hybrid models look promising, however it is not clear which two models combination method should be used as there are various ways to combine them.

1.6. Artificial neural networks against decision trees

By this day there are almost no already made researches which would compare performance of ANN and decision trees in financial markets field. However, there are numerous researches made which compares ANN with statistical-econometrical models. Fadlalla and Lin [FL01] have analysed numerous papers about ANN performance in financial data forecast. In appendix table A1 a summary of many research papers about ANN vs. statistical-econometrical methods is presented. Topic fields covers financial data - bankruptcies, bond ratings, stock prices and other forecasts. From it a conclusion can be made that ANN performs better than discriminant analysis, linear regressions for forecast of financial data.

Tso and Yau [TY07] compared C5.0 decision tree, regression, and neural network for prediction of electricity energy consumption level. Winter and summer predictions were made. In both periods decision tree and ANN were superior to regression predictions measured in RMSE statistics. As for decision trees and ANN - one gave slightly better results in winter and another in summer period. So they seem to be similar, but better than regression models.

Curram and Mingers [CM94] made an empirical comparison of decision tree, ANN, and linear discriminant analysis using seven data sets of various fields (none financial). The research suggests that ANN and linear discriminant analysis is better than decision tree in classification of various data sets. However, it must be noted that decision tree which they used should be worse than the one, used in this research - C5.0.

1.7. Summary

It can be clearly seen that most researchers have a different understanding of how artificial neural networks and decision trees should be applied. ANN algorithm requires researcher to specify various parameters which highly affect the performance of the algorithm. Researchers uses their own set of inputs to their best knowledge or tries a small set of parameters to determine the best one. Another aspect which differs in empirical researches is a method, how these algorithms will be implemented. To this date there is not clear indication that some hybrid method is the best one. Inputs selection is another problem which does not have a common solution in most researches. In the end, a trader who wants to apply these algorithms does not know which ones to choose. This research is going to be the one which would use more commonly or very promising used data sets, algorithm parameters and methods to execute algorithms.

All papers which were analysed in this section lacks one important measure of performance - profitability of strategy by paying attention to the risk. The goal of a trader is to make high profit with low risk. Hence, Sharpe ratio values will be used to evaluate and compare prediction models along with other, popular methods used in common researches. Only a few papers tested their predictions by simulating a trading strategy made of models forecasts. And just one of few financial instruments were used - mainly stocks for unknown reason. This paper aims to fill this gap by testing prediction models on more than 40 financial instruments - futures, which covers a wide range of financial classes (stocks, bonds, agriculture, metals, energy, and currencies). Only by testing a wide range of financial instruments it can be seen if one method is superior to another. However, it must be noted that because of the problem of algorithms' parameters selection and numerous inputs which could be used, this research will not provide definite statement of which method is better as other researcher might get different results using other input sets and/or different algorithms modifications.

2. Data selection

For this research the data of 45 futures financial instruments in total are used. The purpose of using multiple data sets is to minimize results dependency on used data. Financial data tends to have lots of noise and have a specific tendency for even a long period of time. That is why researches, which use only one or just a few financial data sets can be very misleading.

All 45 financial instruments are listed in table 3. One of the reasons why futures were chosen is that they are simple and cheap to trade - this is very important for practitioners in trading. Another reason is that they cover different financial classes. Out of 45 futures, 5 are based on energy prices (oil, gas and similar), 11 on softs (commodities that are grown) prices, 8 on currency (currency exchange rates), 8 on stocks indexes prices, 5 on metal prices, 7 on interest rates based financial instrument, and one on US dollar index which is derived from various currency rates. Results of this research should be more precise as a wide range of financial classes will be covered and will not depend on solely stocks market, or currencies market tendencies and characteristics.

Daily prices will be used so it is necessary to have a long history of data. In any year trades happen and price changes in less than 270 days, therefore it is chosen to use at least 10 years of historical daily data. Data from 2006-01-01 to 2016-12-31 is used. 2006-01-01 is chosen not because more data could not be accessed but older data is not as much relevant as newer data. Markets tends to change over time, therefore trading strategies and tuned forecasted models which would work well for example in 2000's would probably not work well after more than 15 years as characteristics of market changes. A rule of thumb should be to take as less historical data as possible to efficiently learn forecasted system/model to avoid market changes as much as possible. However, it is not clear when market changes so 11 year of historical data or approximately 2800 days (it varies for different financial instruments) is taken based on educated guess of how many data will be needed to train, validate and test forecasting models in this research.

The first and the last minute of every future's trading session is cut off. This means that daily bars do not include those minutes. This is done because it gives more realistic data. The trading method simulation used in this research buys or short sells financial instrument on opening of daily bar. Opened position is closed (bought to cover or sold) on the last bar price - close price. Hence in simulations it would mean that a practitioner is always the first to start trading in the day and the last one to stop trading. Obviously it would not happen so some slippage might occur. A practitioner could sometimes take a lot more time to fill his buying or selling orders than in a few milliseconds. A practitioner might use some other orders filling method rather than instant market orders. Also, the first few trades might happen at a very different price than the latter as price does not settle for the first second of opened market. Therefore, a more realistic simulation is made when the first minute is subtracted. An example of vast price variation in first seconds of trading session is given in appendix figure A1.

Data used in this research is taken from *TradeStation* [Tra] broker. In total 126,361 days of prepared data for forecasting is used.

Table 3. Futures data used in the research.

Description (used by *TradeStation* [Tra])	Abbreviations used in this research	Category	Country	Exchange	Length of the data used for predictions (days)
Australian Dollar	AD	Currency	US	CME	2841
Soybean Oil	BO	Softs	US	CBOT	2767
British Pound	BP	Currency	US	CME	2841
Corn	C	Softs	US	NYSE	2771
Cocoa	CC	Softs	US	ICEUS	2763
Canadian Dollar	CD	Currency	US	CME	2841
Crude Oil	CL	Energy	US	NYMEX	2832
US Dollar Index	DX	Other	US	ICEUS	2831
Euro FX	EC	Currency	US	CME	2841
Eurodollar	ED	Currency	US	CME	2841
E-Mini S&P MidCap 400	EMD	Stocks index	US	CME	2841
E-mini S&P 500	ES	Stocks index	US	CME	2841
DAX	FDAX	Stocks index	DE	EUREX	2786
EURO STOXX 50 Index	FESX	Stocks index	DE	EUREX	2789
Euro Bund	FGBL	Interest	DE	EUREX	2789
Euro Bobl	FGBM	Interest	DE	EUREX	2755
Euro Schatz	FGBS	Interest	DE	EUREX	2788
Swiss Market	FSMI	Stocks index	CH	EUREX	2755
5 Yr U.S.Treasury Notes	FV	Interest	US	CBOT	2825
Gold	GC	Metal	US	COMEX	2832
Copper	HG	Metal	US	COMEX	2832
Heating Oil	HO	Energy	US	NYMEX	2830
Japanese Yen	JY	Currency	US	CME	2841
Coffee C	KC	Softs	US	ICEUS	2766
Live Cattle	LC	Softs	US	CME	2768
Lean Hog	LH	Softs	US	CME	2769
Mexican Peso	MP1	Currency	US	CME	2841
Natural Gas	NG	Energy	US	NYMEX	2831
E-Mini NASDAQ-100	NQ	Stocks index	US	CME	2841
Palladium	PA	Metal	US	NYMEX	2831
Platinum	PL	Metal	US	NYMEX	2819
E-mini Crude Oil	QM	Energy	US	NYMEX	2833
NYHarborBlendstock RBOB	RB	Energy	US	NYMEX	2831
Rough Rice	RR	Softs	US	CBOT	2667
Soybeans	S	Softs	US	CBOT	2770
Sugar No. 11	SB	Softs	US	ICEUS	2766
Swiss Franc	SF	Currency	US	CME	2841
Silver	SI	Metal	US	COMEX	2830
Soybean Meal	SM	Softs	US	CBOT	2770
Mini Russell 2000	TF	Stocks index	US	ICEUS	2841
10 Yr U.S. Treasury Notes	TY	Interest	US	CBOT	2825
2 Year U.S. Treasury Notes	TU	Interest	US	CBOT	2825
30 Yr U.S.Treasury Bonds	US	Interest	US	CBOT	2825
Wheat	W	Softs	US	CBOT	2770
E-mini Dow Futures ($5)	YM	Stocks index	US	CBOT	2828

Success of a trading strategy depends on a data which are used. Success in designing an ANN depends on a clear understanding of the problem which a researcher wants to solve [NI91]. It is critical to know which input variables are important in the market in order to forecast properly. It applies to both neural networks and decision trees.

A researcher who is interesting in predicting financial market prices, trading volume, price volatility, and similar, must decide whether to use a technical analysis data or a fundamental data (an economic data) or maybe both. As this research concentrates on short term trading (intraday trading), technical analysis suits better. Fundamental economic data usually has a frequency of weekly, monthly, or quarterly data (in example GDP figures). Such data would not be as significant when trading in a short period. This statement is not a strict one as a trader could for example use EONIA rates (overnights interest rates) whose frequency is daily and might be suitable in short term trading for some specific strategies.

Futures financial instrument has (as almost every other financial instrument) only two parameters - price and volume. Most of automatic strategies for forecasting uses only these two data sets (this research do not analyse more complex strategies like spread trading where various financial instruments are used - the goal of this paper is not to exploit sophisticated trading strategies). Most of the widely used technical indicators are derived solely from price and volume data sets, therefore only they will be used. In total 19 different technical indicators are applied. For some indicators various days ranges and levels are used, therefore from 19 completely different technical indicators 30 indicators are derived in total. These 30 indicators are used in this research.

A short description of the used technical indicators:

- MACD - calculates the Moving Average Convergence/Divergence (MACD) line. The MACD is calculated by subtracting the 26-period (7.5%) exponential moving average from the 12-period (15%) moving average. The 9-day (20%) exponential moving average of the MACD line is used as the signal line. For example, when the MACD and the 20% moving average line have just crossed and the MACD line falls below the other line, it is time to sell. Days' close price is used to get MACD values.

- RSI - Relative Strength Index calculated from last n days closing prices. It is a momentum oscillator that measures the speed and the change of the price movements. This indicator oscillates between zero and 100. Traditionally the RSI is considered overbought when above 70 and oversold when below 30. 14 and 25 days RSI are used in this research.

- A/D - Accumulation/Distribution (A/D) oscillator. Calculated based on the high, low, opening, and closing prices by adding up buying and selling power. It is normalized indicator by dividing by a number that is a multiple of a period's range. Each day is treated independently. This indicator should not be mixed with Accumulation/Distribution (A/D) line indicator.

- Chaikin oscilator - calculated by subtracting the 10-period exponential moving average of the Accumulation/Distribution (A/D) line from the three-period exponential moving average of the A/D line. Uses high, low, close prices and volume.

- Chaikin volatility - measures volatility as the trading range between price high and low for each day. This indicator helps to recognize those times when volatility is picking up as this offers the best opportunities to make trades. 5 and 10 days Chaikin volatilities are used.
- Fast stochastics - a momentum indicator that shows the location of the close relative to the high-low range over a set number of periods which is set to 10 days. Values above 80 indicate that the security is overbought and the values below 20 indicate that it is oversold.
- Slow stochastics - is an exponential 3 day average of Fast stochastics.
- Slow stochastics 2 - is an exponential 3 day average of Slow stochastics.
- Price rate of change - measures the percent change in price from one period to the next. Uses close prices. 5 and 12 days close price rate of changes are selected.
- Momentum - the difference between two prices (data points) separated by a number of times/-days. In this research 3, 5, and 10 day momentum of closing prices are used.
- Acceleration - the difference of two momentums separated by some number of periods. In this research 3, 5, and 10 day acceleration of closing prices are used.
- Range - days' high price minus low price. The range in which prices moves in a day.
- Real range - days' close price minus open price.
- Commodity Channel Index - measures the difference between a price change and its average price change. High positive readings indicate that prices are well above their average, which is a show of strength. Low negative readings indicate that prices are well below their average, which is a show of weakness. Uses high, low, and close prices.
- Aroon-Up indicator - identifies trends and the likelihood that the trends will reverse. The Aroon-Up measures the number of days since a n-day high. Uses high prices. Indicators calculated from 5 and 15 days are used.
- Aroon-Down indicator - identifies trends and the likelihood that the trends will reverse. The Aroon-Down measures the number of days since a n-day low. Uses low prices. Indicators calculated from 5 and 15 days are used.
- TSI - true strength index indicator. It measures the double smoothed price change relative to the double smoothed absolute price change. The market is bullish when True Strength Index is positive and the bearish when it's negative. Uses close prices. For smoothing these pairs are used: 25 and 13; 10 and 5.
- Volatility ratio - identifies price ranges and breakouts. It shows time periods when price has exceeded its most recent price range to an extent significant enough to constitute a breakout. If value of volatility ratio is greater than 0.5 it is the signal for breakout. Uses high, low, and close prices.

- ADX - the Average Directional Index. It is used to measure the strength or weakness of a trend, rather than the actual direction. Directional movement is defined by +DI (positive directional movement) and -DI (negative directional movement). In general, the bulls have the edge when +DI is greater than - DI, while the bears have the edge when - DI is greater. Uses high, low, and close prices. In this research 14 and 5 days ADX indicators are used.

It can be seen from indicators and oscillators definition that all of them are in some interval or fluctuates around zero. Therefore, they are suitable both for decision trees and neural networks. Indicators which would tend to rise or fall most of the time would not be suitable as decision tree rule's answer would be one for one part of data and another for another part - decision would change only a few times in all data set.

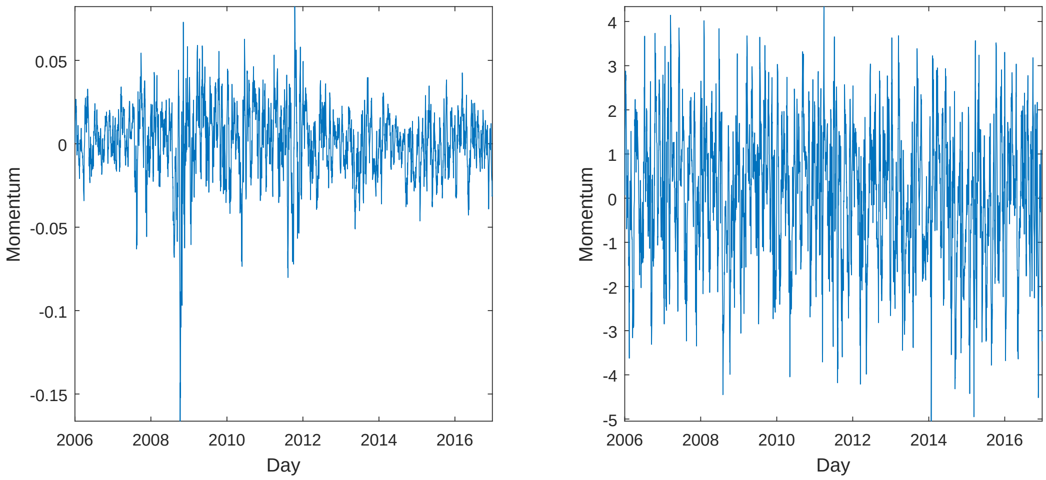

Figure 3. 10 days Momentum of AD financial instrument. Left chart shows the bare technical indicator. Right chart shows the technical indicator after division by trailing volume average.

Used data covers 2008 financial crisis period, therefore some serious changes in volatility and trading volume can be seen in the data. In the left graph of figure 3 close price 10 days momentum of AD financial instrument is displayed. It can be clearly seen, that momentum had significantly higher values in 2008. Some increase can also be seen in year 2011. Such changes in data set would make some troubles for decision tree and neural network to optimize its' parameters. In order to avoid this, momentum was divided by trailing volume average values. Trailing volume average values show a typical volume level at any time. By dividing momentum by trailing volume average a more suitable data set is made. For example, in 2008 momentum peaked but so did the volume. Therefore, by dividing momentum by trailing average volume we get more stationary data. Such data is more suitable for forecasting/classification methods which are used in this research. In the right graph of figure 3 momentum divided by trailing volatility average is presented. This modified data set has less variance and is distributed in a smaller range of momentum values.

Such method of dividing technical indicators by trailing volume average is applied to all Momentum, Acceleration, Range, Price rate of change, and Real range indicators because they all share this property of changing variance.

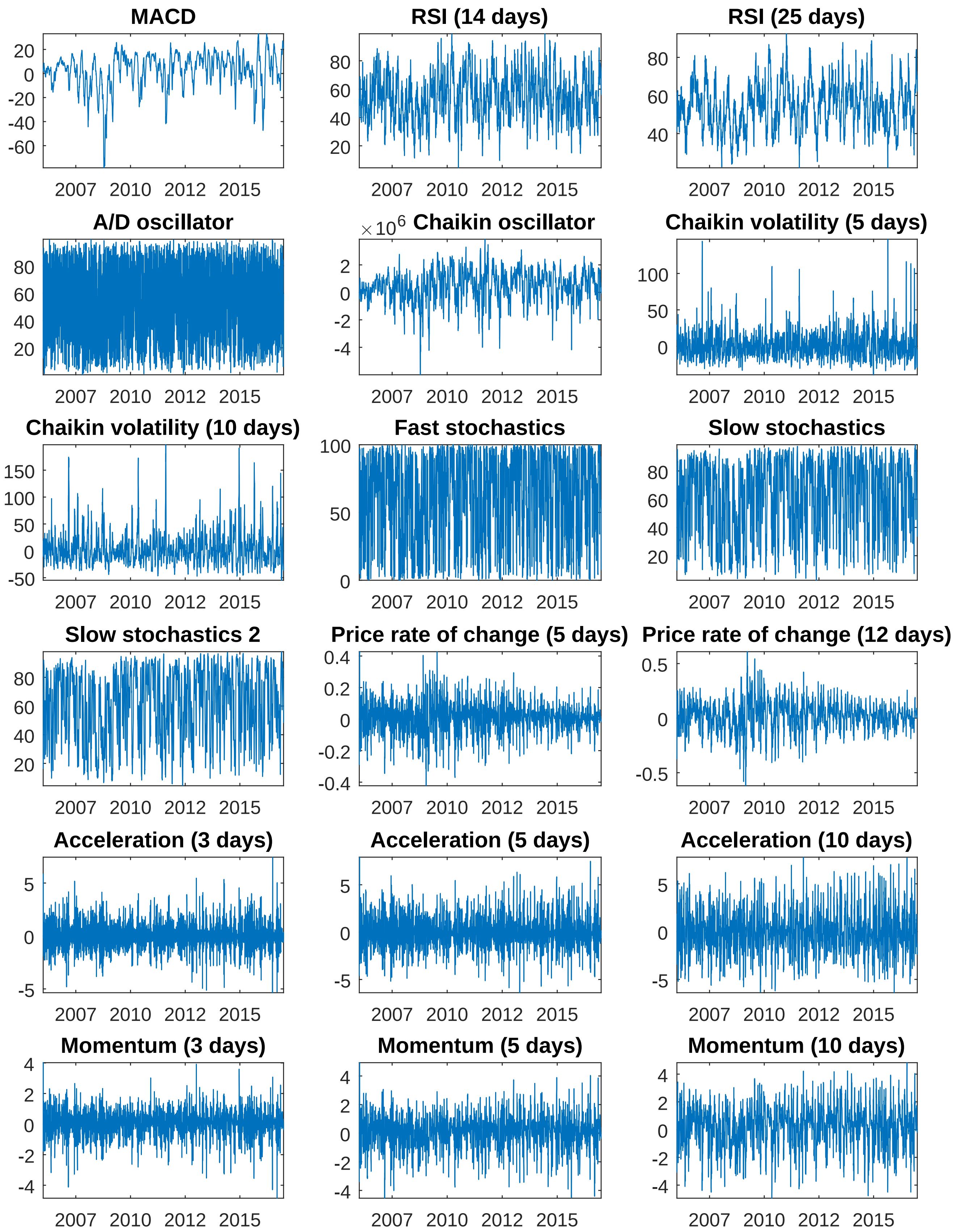

Figure 4. All indicators of ES future financial instrument data.

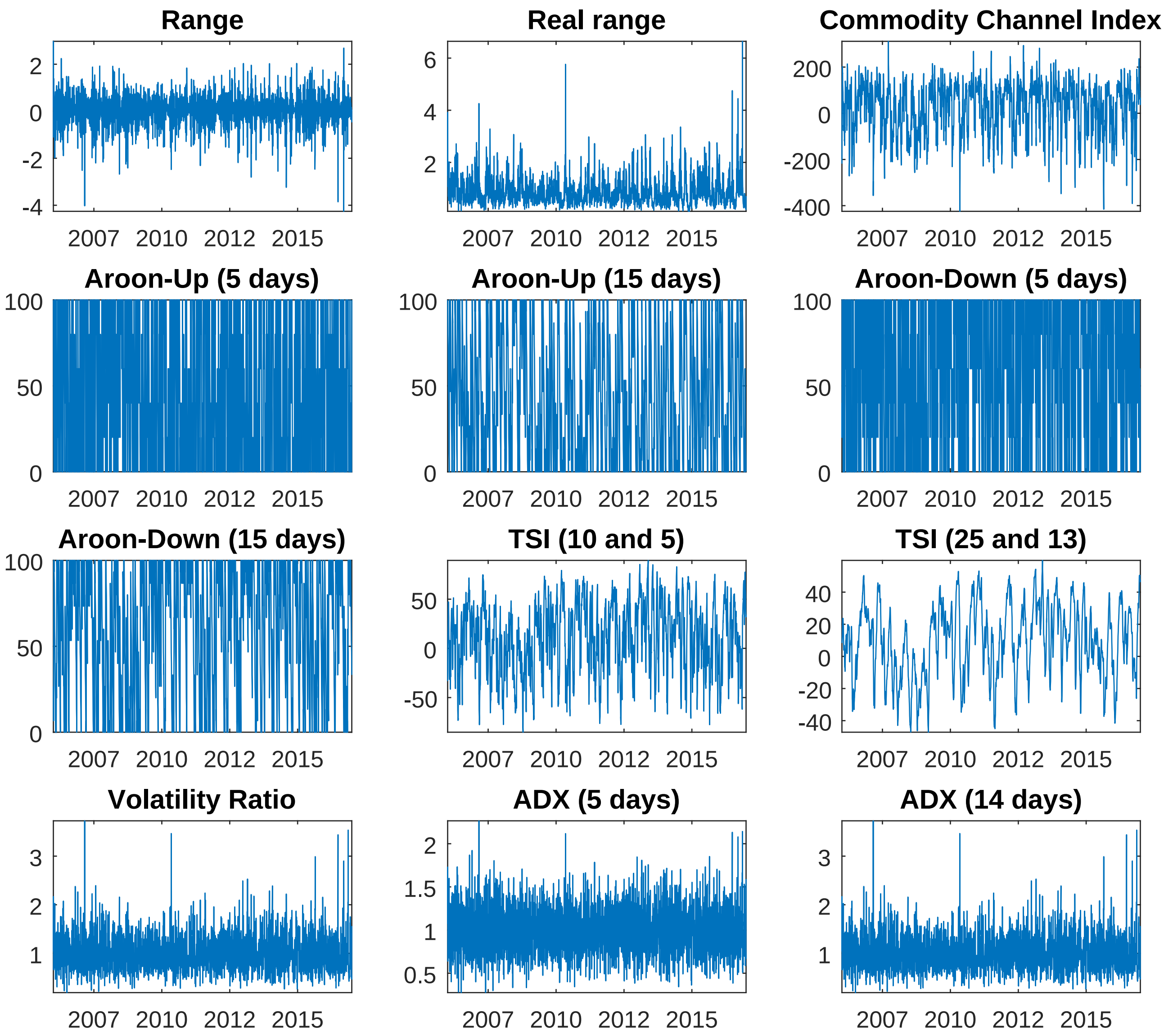

Figure 5. All indicators of ES future financial instrument data.

All 30 indicators for ES financial instrument data is shown in figures 4 and 5. Indicators are displayed after division by trailing volatility average where needed. As seen in figures 4 and 5, different indicators has different properties. Ranges of technical indicators differs a lot. Such differences between indicators are not a problem for C5.0 decision tree algorithm, however neural network would have to adapt weights between neurons accordingly to the range of movement of the indicator (at first iterations indicators with a higher range of movement would have more impact to the output). Therefore, it is decided to scale data to interval $[-1,1]$ before applying neural network.

3. Trading strategy

This paper focuses on higher frequency data, therefore daily data is used to forecast next day change in price. From a practitioner's point of view, it is more important to forecast a price movement direction and to decide whether it is worth to place a trade, than to forecast an exact price for tomorrow. Therefore, a forecast/classification of ”buy”, ”sell short”, or do nothing (”flat”) is enough. These 3 trading commands will be this papers' trading strategy.

A simulated trade is placed for a day only. There are three possibilities: future financial instrument will be bought on open of the day and sold on the close of the day (the buy day); sold short on the open of the day and bought to cover on the close of the day (the sell short day); no buys or sells in the day (the flat day).

Both ANN and DT goal is to tell whether a trader should make a ”buy”, a ”sell short”, or a ”flat”. So a prediction is converted to a classification problem. In figure 6 sorted price changes of every day for MP1 future are plotted. It can be seen that there are approximately equal number of positive and negative change days. This sorted data is divided into three intervals in a such way:

- Sorted data is divided into three intervals. ”Flat” is assigned to one of the intervals and consists of 30% of all days. The middle of the ”flat” interval is zero (day 1380 in figure 6). 15% from the middle is assigned to ”flat” for negative change dates (to the left) and 15% from the middle is assigned to ”flat” for positive change dates (to the right). Hence, ”flat” covers the smallest changes in a day when trading is not worthwhile.
- ”Sell short” is assigned to all days from day one to the first ”flat” day. Will be approximately $\frac{100\%-30\%}{2}$ of days.
- ”Long” is assigned to all days from the last ”flat” day to the last sorted day. Will be approximately $\frac{100\%-30\%}{2}$ of days.

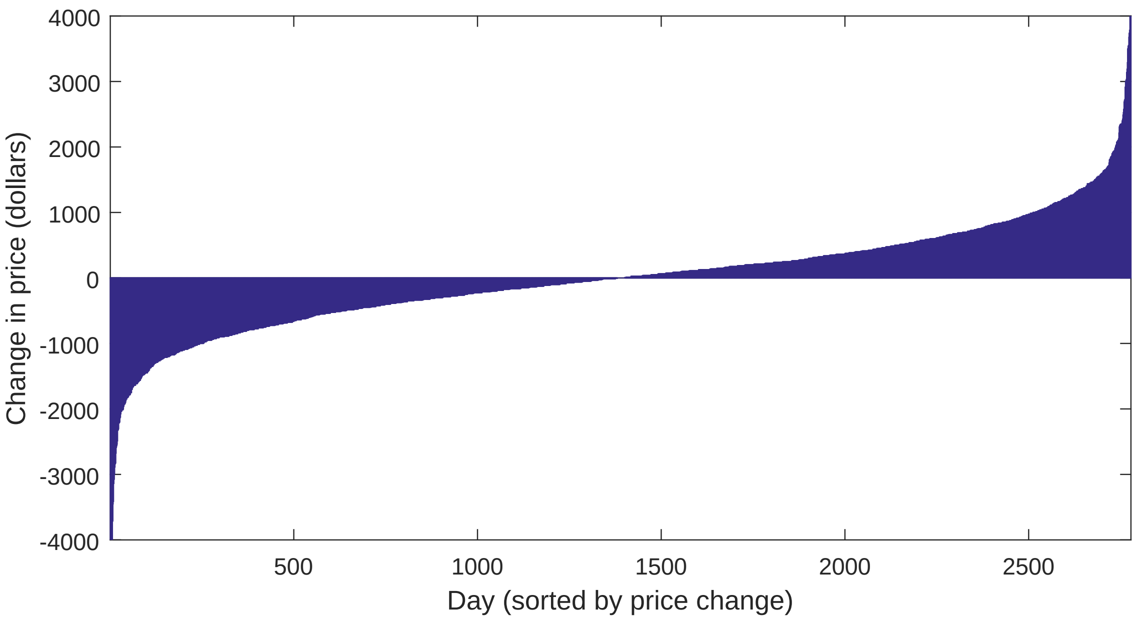

Figure 6. Sorted price changes in a day of MP1 future financial instrument.

This split is not symmetrical for "long" and "sell short" days. Suppose that there are 100 days and 55 of them have positive day change and 45 have negative. Then "flat" would still be assigned to 15 least negative and 15 least positive days (30% of all days). Then "sell short" would be assigned to $45 - 15 = 30$, whereas "long" would be assigned to $55 - 15 = 40$ days.

Usage of such method means that for example day change of -500 dollars could be assigned to "flat" class, but +500 could possibly be assigned to "long" class. However, usually there are no significant asymmetry of positive and negative day changes distribution. Also, other methods to specify the class ("long", "sell short", or "flat") for day change would also have its' own disadvantages. Moreover, the same data with the same method to assign the class to a day change is used both for neural network and decision tree, so both classification models will get the same asymmetrical data as their target, thus results will not be biased.

4. Testing period selection

After first tests of neural networks and decision trees, trading strategies produced from these algorithms did not perform well in testing data which spanned from 2016-01-01 to 2016-12-31. When comparing financial time series testing results with training results, a researcher must have in mind that differences occurs because of two factors. First of all, it is overfitting which occurs for most of data types - not only financial data. The second is the change of market conditions/characteristics which happens to some of time series data sets and is unavoidable in finance.

As for overfitting, both decision tree and neural network have their own methods to minimize it. Decision trees uses pruning whereas neural network applies various stopping criteria to stop learning before overfitting. So a practitioner can reduce overfitting by using some suitable methods. However, a practitioner cannot do anything against changing market conditions. If, for example, a trained neural network would be a very good trading strategy for training data because ANN would somehow exploit inefficiencies in market to make correct predictions and there would be no overfitting, then such ANN should predict in the same accuracy for testing data as it did for training data. Yet, because of changing market conditions and disappearance of some specific inefficiencies in market, such classification model would almost certainly work worse.

It is a very difficult task to separate performance decrease in testing data because of overfitting and because of changing markets. First tests of both neural networks and decision trees gave significantly worse results in testing data than in training data. Some of this decrease in performance might have been because of overfitting, however, as both algorithms use methods to prevent too much of overfitting, such hefty loss of performance might be caused of changing conditions in the market.

It must be noted, that conditions in the market might change only for some period of time when, for example, some higher uncertainty in markets would occur for some period of time. Therefore, even when predictive models would perform only marginally good for testing data, accuracy of prediction/classification might rise after the market would get back to a typical condition. Take for example changes of volatility for S&P 500 stocks index. In appendix figure A2 it can be seen that in 2004-2006 volatility was much lower than in 1997-2003 but eventually increased and stayed increased for 5 years. A decrease of volatility might have significantly reduced a performance of some specific trading strategies/predictive models for years 2004-2006. It was decided to make a research to answer question: are such losses in performance occurs because of changing market conditions in year 2016?

This research is motivated not only because of results which were made in this research, but also because of dropped profit performance of systematic funds which, same as in this research, uses technical indicators to invest and earn profit. A very good example of such funds performance is the Altegris 40 index which is derived from performance of top 40 commodity trading advisor (CTA) systematic trading funds. In appendix table A3 returns by year are listed. It shows that systematic trading funds struggled in 2016, because the index derived from their performance lost -3.13 % of its value. So it supports a concern that 2016 might not have been suitable for such prediction models as used in this research. Therefore, testing data results might be affected by the change of market.

Another reason for such research is to test whether MSE is suitable fitness value for DTs and ANNs. The goal of a practitioner is to make a best trading strategy which would make the most profit with as low risk as possible. For such goal Sharpe ratio is useful parameter. However, neural networks with backpropagation cannot use Sharpe ratio, but instead uses error parameter - MSE (in this research). A question, which should be answered when doing such kind of researches: is MSE a suitable fitness parameter when in the end what matters is not the lowest MSE (lowest mistakes), but the best trading strategy - maximum Sharpe ratio.

A research to test these questions for ANN was done in a such way: for every of 45 future financial instruments, 15 tests with different initial weights were done and MSE values with Sharpe ratio were saved. Also, 12 different testing date periods were selected. This means $45 \times 15 \times 12 = 8100$ tests in total, hence results should avoid randomness.

For neural network, Sharpe and MSE values were saved after every iteration to witness if decreasing MSE value causes an increase of Sharpe ratio. 11 years of data was split in a such way:

1. Testing data from 2006-01-01 to 2006-12-31. Remaining data is training data.
2. Testing data from 2007-01-01 to 2007-12-31. Remaining data is training data.
3. Testing data from 2008-01-01 to 2008-12-31. Remaining data is training data.
4. Testing data from 2009-01-01 to 2009-12-31. Remaining data is training data.
5. Testing data from 2010-01-01 to 2010-12-31. Remaining data is training data.
6. Testing data from 2011-01-01 to 2011-12-31. Remaining data is training data.
7. Testing data from 2012-01-01 to 2012-12-31. Remaining data is training data.
8. Testing data from 2013-01-01 to 2013-12-31. Remaining data is training data.
9. Testing data from 2014-01-01 to 2014-12-31. Remaining data is training data.
10. Testing data from 2015-01-01 to 2015-12-31. Remaining data is training data.
11. Testing data from 2016-01-01 to 2016-12-31. Remaining data is training data.
12. Testing data from 2007-01-01 to 2015-12-31 by randomly taking 30% of data. Same randomly selected days were used for all 15 tests for the same future financial instrument. Remaining 70% of data is training data.

Every year was taken as different testing data sets to measure whether markets changed. If trained neural network would work well with for example first year data, but would not work so well with a data after 10 years, that would mean that markets changed significantly in respect to our models. 30% of data in period 2006-01-01 to 2016-12-31 was taken as testing date to check whether it would be a suitable data set which would not be as much affected by market's situation for any particular time period.

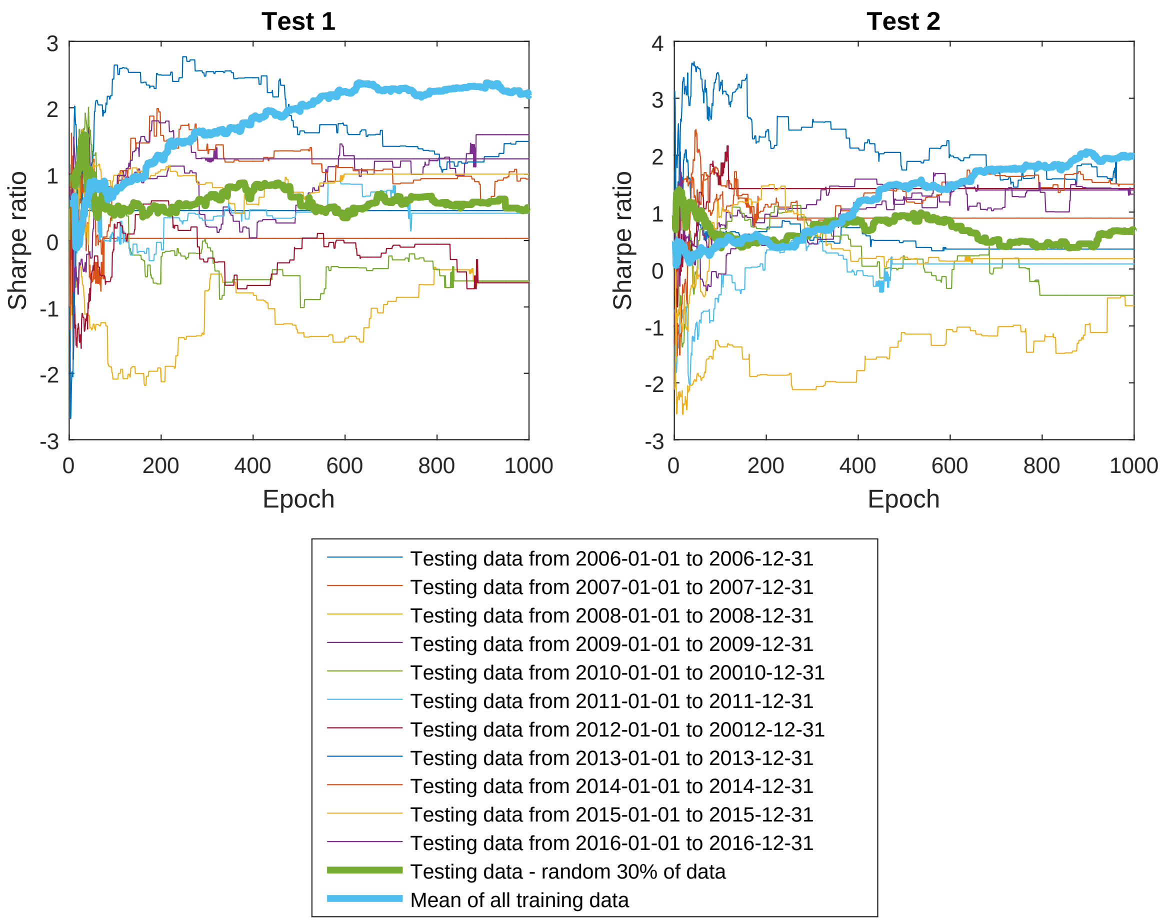

Figure 7. Two tests of NQ future financial instrument using ANN.

In figure 7 an example of two ANN test is presented for NQ financial instrument. The difference between them are just different initial neural network weights. Sharpe ratio is significantly different in both test for the same training and testing data. Also Sharpe ratio values are not consistent and varies highly when different days for training data and testing data are selected. It shows why numerous tests (8100 in this case) are necessary - only the averaged results of many tests can be trusted as results varies significantly for every test.

An average of all 8100 tests shows clearer information as seen in figure 8. First of all, it can be clearly seen that as MSE value decrease for training data, Sharpe ratio for training data increases and such relation holds for at least 1500 epochs (that much were done in tests). This result is very important because it tells us that MSE value can be used to train neural network even then the actual goal is to maximize Sharpe ratio. Correlation between MSE and Sharpe ratio is -0.7896 (high correlation).

Another important result is that testing data results differs significantly for most testing data periods. This means that there actually are some significant changes in the market. Hence, if testing data would be taken from 2016-01-01 to 2016-12-31, results would be biased. Green line denotes data set, where 30% of data are randomly split to testing date subset. It can be seen that this line is a good generalization of all results, therefore this randomly selected data set should be used in further researches.

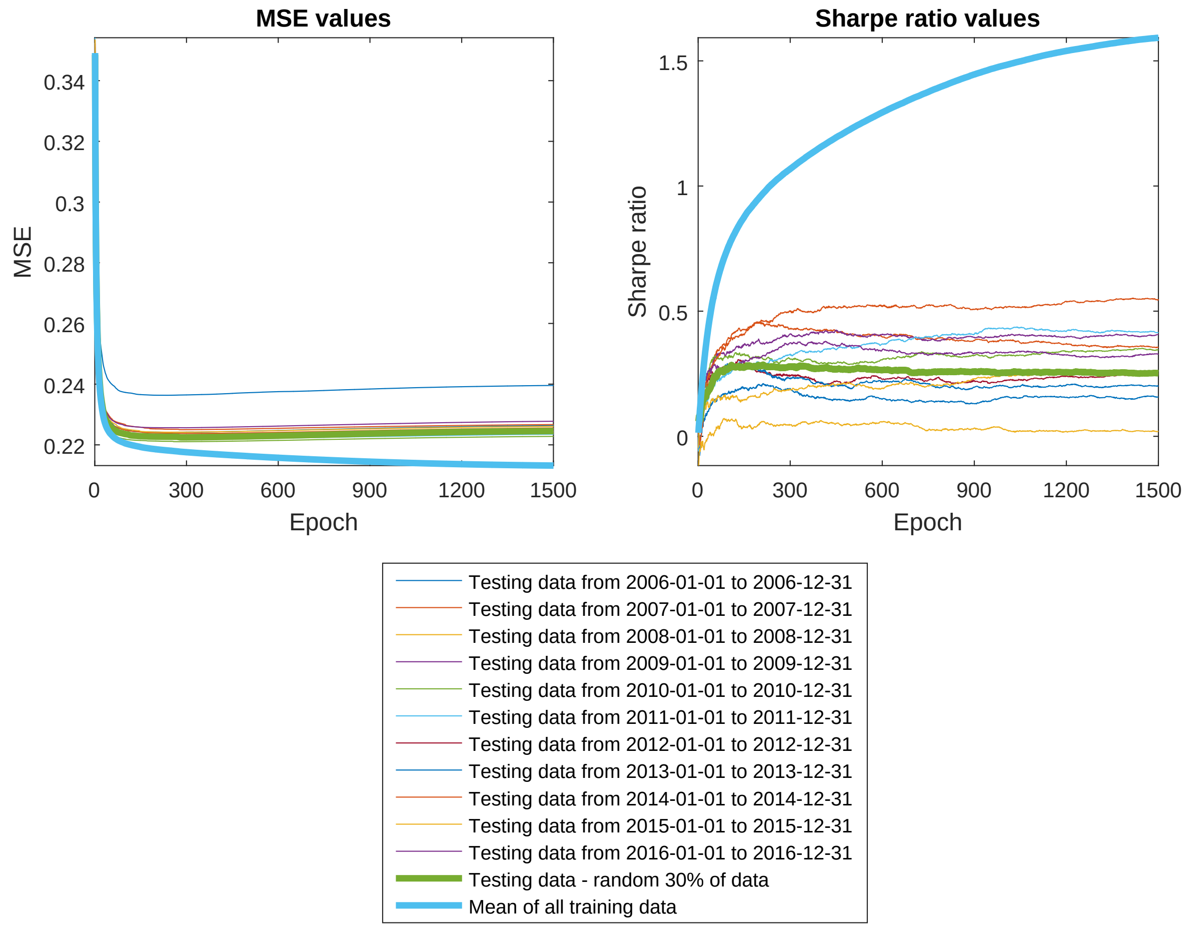

Figure 8. Average of 8100 tests using ANN.

To test decision tree results for every financial instrument only one test with same training and testing data sets were used instead of 15 because decision tree always returns the same results for the same input data as long as parameters are the same. So in total there were $45 \times 12 = 540$ tests. For decision tree, similar results were obtained. Testing data sets were taken in the same way as for ANN tests. In figure 9 MSE and Sharpe ratio scatter plots of all data sets are given. One circle denotes one of 540 tests results. It can be clearly seen that a decrease in MSE value causes an increase in Sharpe ratio. Keep in mind that for decision trees MSE is calculated differently than for neural networks. For decision tree MSE is calculated using errors which are defined like this:

$$\text{error} = \begin{cases} 0, & \text{if prediction (sell short/long/flat) is correct} \\ 1, & \text{if prediction is incorrect} \end{cases}$$

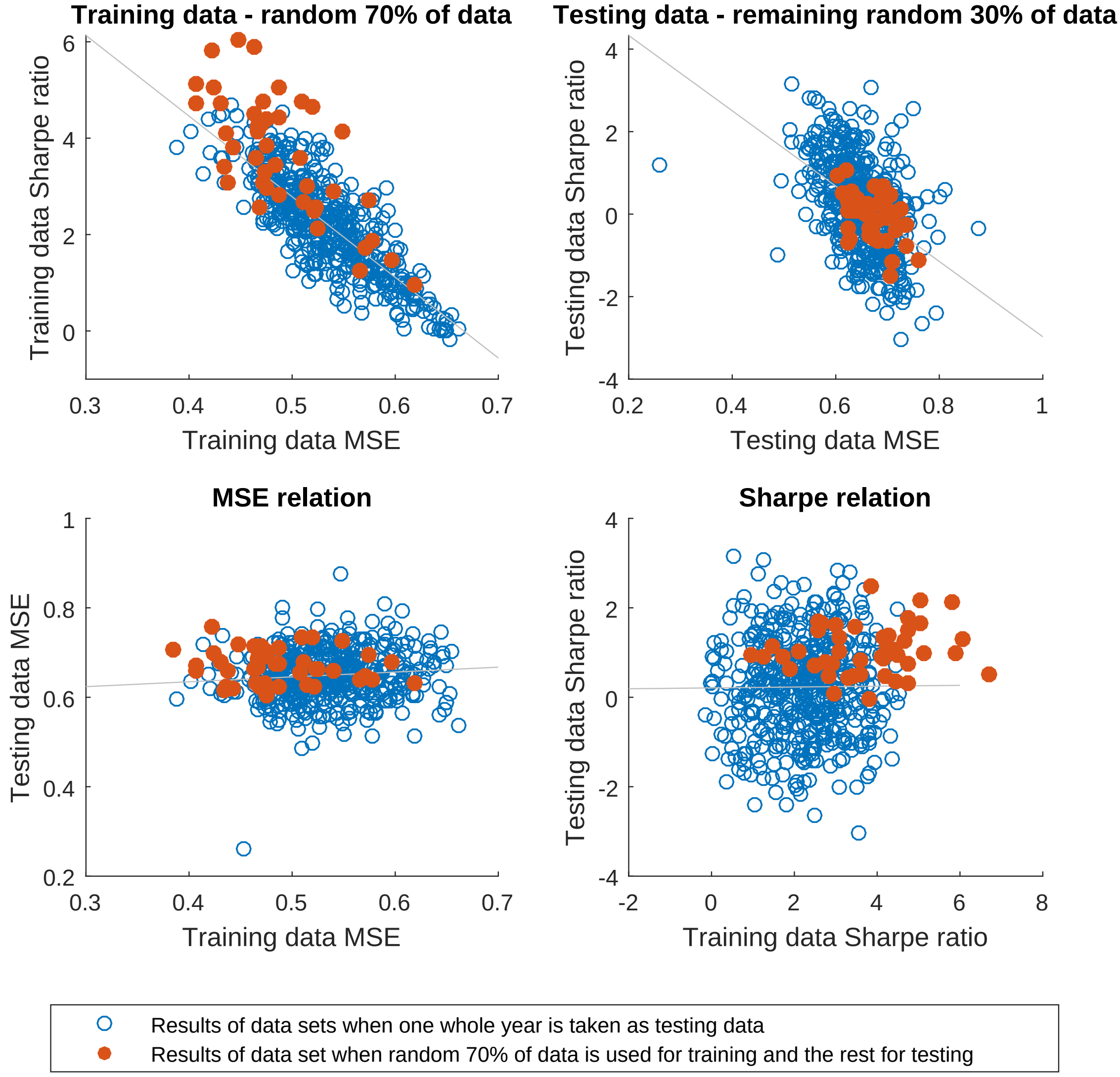

Figure 9. 540 tests results using decision tree.

Figure 9 also shows that taking 70% of random days is suitable data set. Reds dots are rather good generalization to all blue circles, even though not perfect. Keep in mind that with decision tree significantly less tests were made than with ANN so it has more randomness, hence some imperfections can be tolerated.

Both decision tree and ANN tests results indicates that the method of taking random 30% for testing data set is the best way to test prediction models. In such way, testing data results do not depend on specific time period. Yet, there are a few disadvantages:

- A practitioner in trading would care more about current performance (testing data for 2016) rather than about global testing data performance of randomly selected days as he/she would use prediction model for 2017. 2017 data should be more similar to 2016 data rather than to randomly selected 30% of all days.

- Then data set is split to randomly selecting data for training data and testing data sets, then similarity issue could potentially occur. Market data might remain similar for example 3 days in a row and 2 of these days might be taken as training day and 1 day might be taken as testing data. Then while learning algorithm trains with training data, the training of that 1 day from testing data sample might happen as well because of too similar data sets. It is not clear if this really occurs or is just a false hypothesis.

The main task of this paper is to compare two different artificial intelligence methods which will use the same data subsets for training, validation, and testing for the same future financial instrument. Based on the results it was decided to use 70% of randomly selected days for training (88,457 out of 126,361 days in total) and the rest for validation and testing (18,952 days in total for each). In a such way, results will not depend on specific market conditions for specific period of time so they will be better for comparison. However, they will be less useful for practitioners who would want to trade such models with new data in 2017.

5. Artificial neural network properties

5.1. Optimal parameters selection

After literature overview in section 1 and data analysis in section 2 it was decided to use such neural network learning procedure:

1. Input data sets are adjusted by volatility and scaled to interval $[-1,1]$.
2. SAND algorithm is applied to select all weights between network's neurons so that weights would cover n dimensional space as much as possible.
3. Genetic algorithm is used to determine which inputs should be used to learn neural network and which ones should be left out.
4. Neural network based on feed-forward and backpropagation with gradient descent is used to learn network. MSE is used as error parameter.

All parameters are optimized based on validation data results.

Epoch number determines how many times ANN will have its' weights changed. Averaged results of all 45 financial instruments simulations for validation data are presented in figure 10. MSE value is at its' lowest after 500-750 epochs. Sharpe ratio is at its' highest between 100 and 2000 epochs. Optimal epochs number is chosen 750. Validation data MSE value tends to constantly rise after 1000 epochs. Sharpe ratio starts to significantly fall after 2000 epochs. This is because neural network over-fits training data.

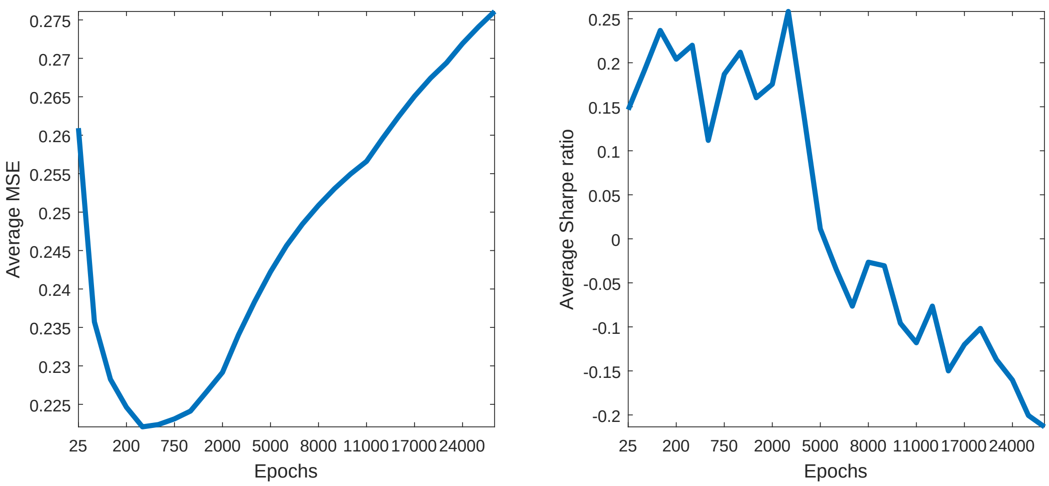

Figure 10. Results of different epochs number for validation data.

MSE and Sharpe ratio relation could change after more epochs than 30000 (that much is tested). When training data reaches extremely good results because of very precise training data fitting, then ANN not just over-fits data, but for some network branches could find even more precise weights values, which then might yield good results with testing or validation data. Disadvantage of overfitting some branches weights might be outweighed by advantage of more precise other branches weights. In figure 11 results of all 45 financial instruments with training data are plotted. Even after 5000 epochs Sharpe ratio keeps increasing with almost the same pace and continues at least until 30000 epochs. No more than 30000 epochs were made because for some financial instruments too much overfitting might occur and a variance of results in validation and testing data would increase, hence higher number of extremely poor models would be made. In order to avoid this, epochs number is limited and optimal value is taken 750, ignoring the idea that similar or even better averaged results might be achieved after more than 30000 epochs.

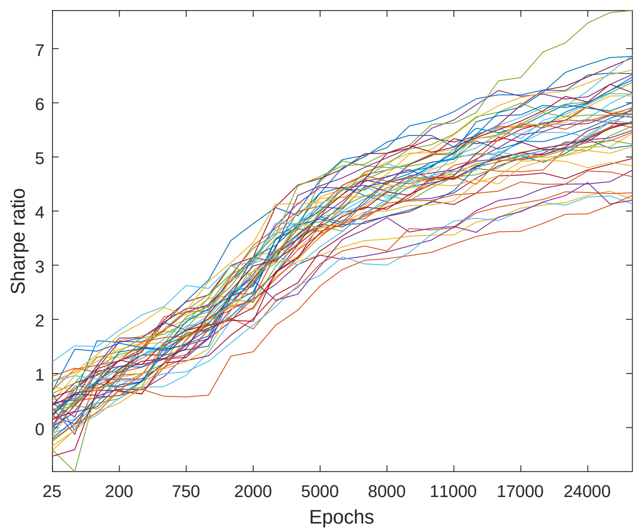

Figure 11. Results of different epochs number for training data.

When there are only 750 epochs, then a risk of over-fitting is low, so such stopping criterions as minimum gradient change, validation checks, and minimum performance change are not so important and would almost never be triggered, hence they are not used.

In figure 12 are results of different learning rate for validation data. 0.2 is optimal by Sharpe ratio results for validation data. MSE values are the best when learning rate is 0.1, 1.5, or 0.2. So 0.2 was taken as optimal value.

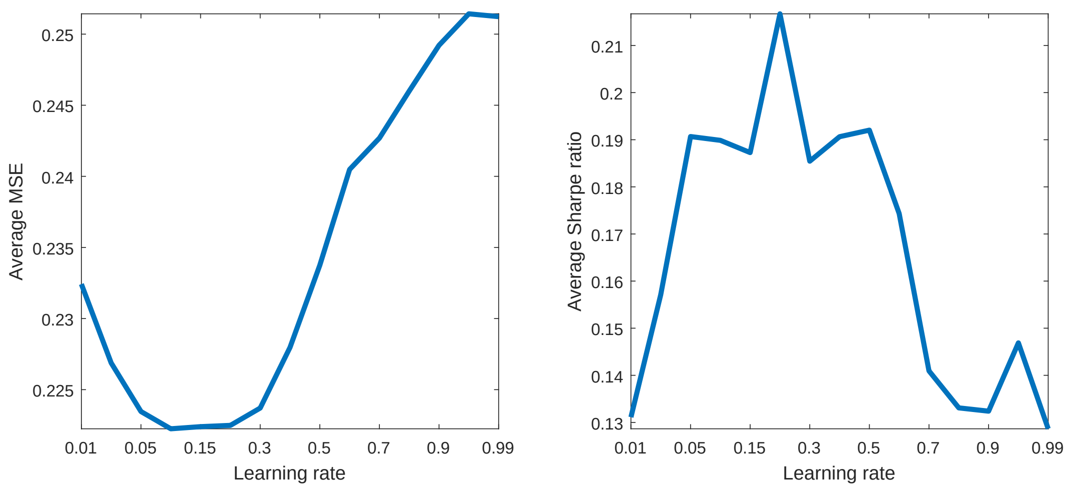

Figure 12. Results of different learning rates for validation data.

Results with different momentum constant is presented in figure 13. 0.7 is chosen as optimal momentum constant based on simulations. Average MSE are lowest between 0.4 and 0.85, whereas average Sharpe ratio peaks at 0.7.

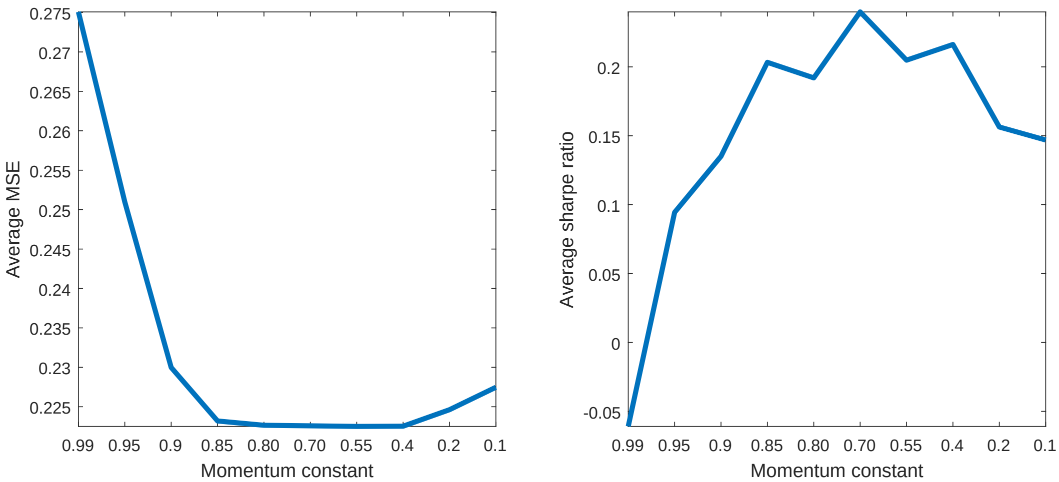

Figure 13. Results of different momentum rate for validation data.

Optimal value of neurons in a hidden layer is not clear. In figure 14 results of MSE values for validation data are presented. It seems that the more neurons after 8 are used, the worse is MSE value. However, keep in mind that difference between 6 and 50 neurons MSE values is less than 0.003, so it is almost non-existent. Results of Sharpe ratio gives different results (look at figure 15). 10 hidden neurons yield best Sharpe ratio for validation data and its' results distribute with one of the lowest variances. Therefore, 10 is chosen as optimal hidden neurons number.

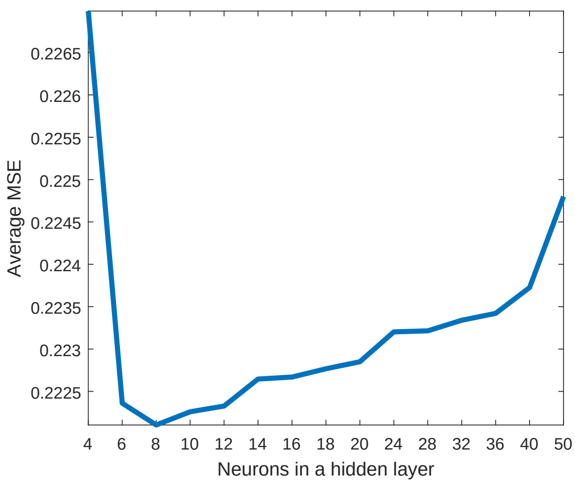

Figure 14. MSE results of different number of neurons in a hidden layer for validation data.

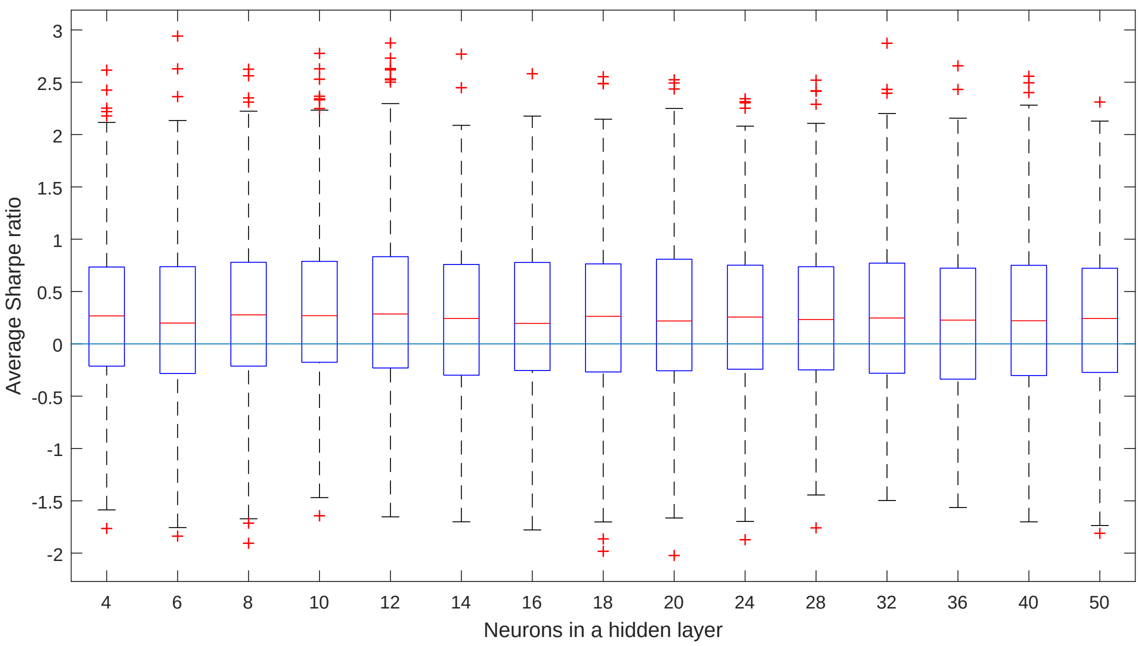

Figure 15. Sharpe ratio results of different number of neurons in a hidden layer for validation data.

Another parameter of neural network which should be tested is the number of outputs. One test is made using 3 output neurons (default) and another with only one output neuron. In a test with 3 output neurons (one for "long", "sell short", and "flat" predictions), target of correct class output is 1 and 0 for other two outputs. Prediction is made by choosing neuron with the maximum value. In a test with 1 output neuron, target for "long" is 1, for "flat" - 0, and for "sell short" - -1. Prediction is made by classifying 30% closest values to 0 as "flat", remaining positive as "long", and remaining negative to "sell short". Confusion matrices for validation data are given in tables 4 and 5. Presented data is calculated using all 45 financial instruments results.

		Prediction		
		Long	Flat	Short
Target	Long	46.07%	23.44%	30.49%
	Flat	38.70%	31.26%	30.04%
	Short	33.54%	23.58%	42.87%

Table 4. Results of 3 outputs neural network for validation data.

		Prediction		
		Long	Flat	Short
Target	Long	36.91%	29.31%	33.78%
	Flat	34.95%	30.28%	34.77%
	Short	35.11%	29.90%	34.99%

Table 5. Results of 1 output neural network for validation data.

Neural network with 3 outputs shows better results as it has higher percentage of correct predictions. Moreover, average Sharpe ratio value is 0.33 with 3 outputs network and 0.20 with 1 output network. Therefore, network with 3 outputs suits this financial data better.

5.2. Genetic algorithm

Genetic algorithm is applied to select inputs for ANN. There are 30 technical indicators (inputs) in total. ANN can get from 1 to 30 inputs, therefore in total there can be 1,073,741,823 ways to select inputs:

$$\binom{30}{1} + \binom{30}{2} + \cdots + \binom{30}{30} = 1{,}073{,}741{,}823$$

The idea is to create a population (pool) with different inputs subsets and use genetic algorithm to determine the best one. One chromosome of population is one of the possible inputs subsets. Initial population is made out of randomly selected technical indicators where every technical indicator has a 50% chance to be selected to a chromosome. Hence, initial population chromosome can contain any number of technical indicators from one to the maximum number of indicators. Based on the size of chromosome (number of inputs), hidden neurons number was increased or decreased proportionally.

Genetic algorithm parameters like elite count (set to 2), mutation rate (set to 15%), tournament function (chooses each parent by choosing 2 players at random and then choosing the best individual out of that set to be a parent), and crossover function (creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child) are typical parameters and would not drastically change results if were changed, so they are not optimized. More important parameters are generations count and population size. Population size and generations count are negatively correlated - if population size is bigger, then less generations are needed, and vice versa. Population size is set to 40.

To determine how many generations are needed to get good input subsets ”the false technical indicators method” is used. 15 random dummy technical indicators are generated uniformly in interval $[-1{,}1]$ and added to inputs set. So in total inputs set consists of 45 technical indicators (30 real and 15 dummy technical indicators). The idea is to see how many generations are needed for genetic algorithm to throw out dummy technical indicators from population’s input subsets. Such test should indicate how many generations are needed to throw out irrelevant technical indicators from real 30 technical indicators input set.

Inputs subset of population was saved after every generation. This method was applied to all 45 financial instruments. The fitness function is MSE value of predictions. All data is split to training (70%), validation (15%), and testing (15%). Genetic algorithm had to train and test 40 neural networks for every generation (set to 150) and it was done for all 45 financial instruments, thus $40 * 150 * 45 = 270{,}000$ networks were trained in total. In order to speed up this process, neural network is trained only up to 100 iterations. It is assumed that a network A which would be better after 100 iterations than network B, would also be better if both A and B networks would be fully trained (with no 100 iterations restriction).

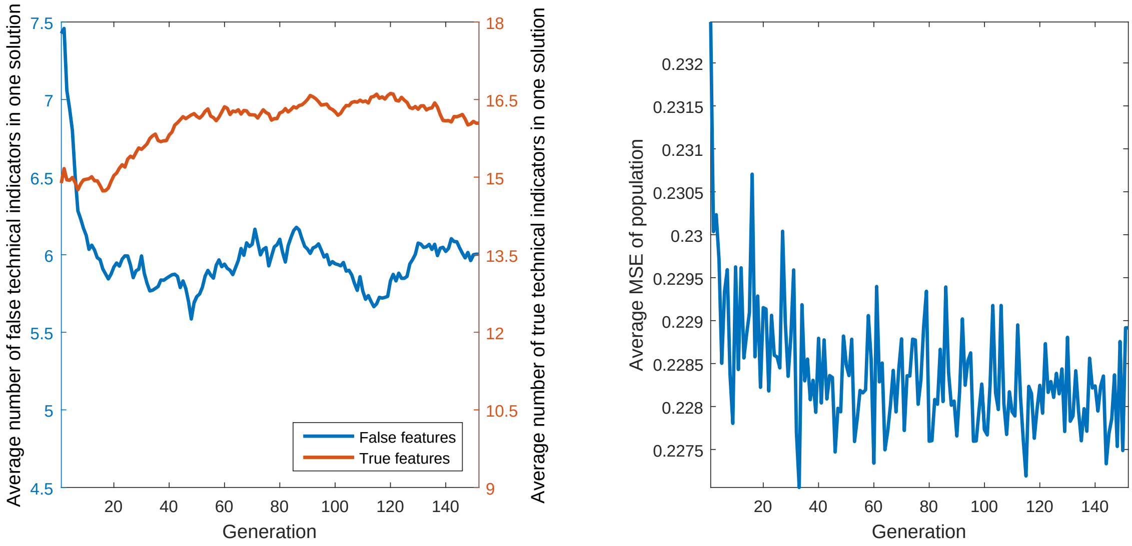

Figure 16. Population results average.

There are a lot of randomness in financial market, therefore only average results should be looked at. In figure 16 all tests average for validation data is presented. Average is made using all population solutions. In the right graph it can be seen that MSE falls only until 50'th generation, hence optimal generations count is 50. The left graph shows that the count of false technical indicators on average in a chromosome also falls only up to 50'th generation and later remains the same. MSE stops to decrease when the false technical indicator average settles just below 6 and true technical indicators average rises to 16. It must be noted, that because of huge randomness in financial data these 15 randomly generated technical indicators might not be very distinctive from the true technical indicators (technical indicators are noisy information about financial instrument price). Because of this randomness and the fact that neural network can nullify unnecessary inputs to some extent, genetic algorithm does not remove the false technical indicators from the population completely.

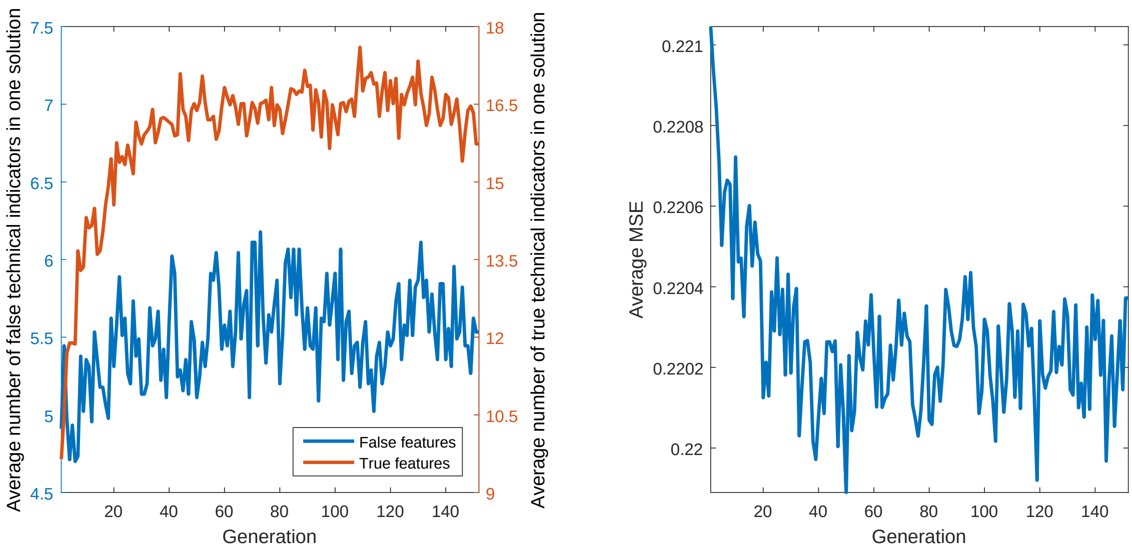

Figure 17. Best solution results average.

To get a clearer image of the genetic algorithm's performance, lets analyse only results of the best solutions (only best solution is relevant, because only one input subset is used for one financial instrument). In figure 17 results of the best solutions are shown. Notice that for higher clarity y-axis limits in figures 16 and 17 left graphs are made the same.

Best solutions MSE also falls until 50'th generation. However, the false technical indicators number in the best solutions stays the same (blue line). So in order to get as less false technical indicators in input subset as possible, only one generation is enough. MSE value falls only because more appropriate subset of true technical indicators is selected from all true technical indicators set, but not because the false technical indicators number is reduced.

The significance of benefits by applying genetic algorithm to select the best technical indicators subset is questionable, therefore other method will be tested.

5.3. Correlation

Another method to be tested for inputs selection is correlation analysis. Correlation between technical indicators and the desired outputs determines whether the input and the output value varies in the same or opposite direction. Using input sets which has high correlation with the outputs tent to generate good ANN models [WLM+08]. This method is realized by using only those technical indicators which correlate the most with the outputs for training data. Both positive and negative correlation indicates a relationship of the input and the output. Hence absolute correlation values are used to determine highest correlations. Obviously, smaller input subsets mean that less neurons in a hidden layer should be used. It is already determined in section 5.1 that with 30 inputs the optimal hidden neurons number is 10. This number was reduced as the number of input subset decreased. Seven input set types were tested in total: all 30 technical indicators used (with 10 hidden neurons); only top 20 highest correlating technical indicators used (with 9 hidden neurons); top 15 highest correlating technical indicators used (with 8 hidden neurons); top 10 used (with 7 hidden neurons); top 8 used (with 7 hidden neurons); top 5 used (with 6 hidden neurons); top 3 used (with 5 hidden neurons). For every financial instrument 10 tests with different initial weights were made to reduce randomness.

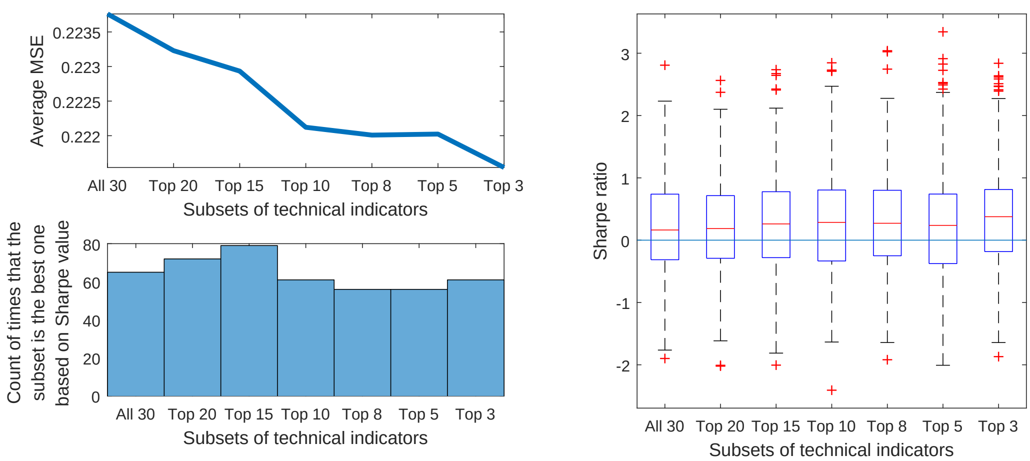

Figure 18. Results of different input subsets.

In figure 18 results for validation data are presented. Based on the results, it is clear that better results can be achieved then smaller number of inputs is used than 30. Tests shows that MSE decreases as less of the most correlated technical indicators are used but the change slows down when less than Top 15 the most correlated technical indicators are used. Based on Sharpe ratio optimal input set seems to be between Top 15 and Top 8 most correlated technical indicators selected.

In total 450 tests were made (10 for each 45 financial instruments with different initial weights). In the left bottom graph of figure 18 a histogram of the best input set for every test is shown (i.e. value 65 for ”All 30” means that out of 450 tests, 64 times the best input set is to use all 30 technical indicators). This histogram indicates that the optimal size of input set varies a lot. ANN used in this research can reduce the impact of unnecessary inputs to some extent, hence is not extremely important to select the smallest set. MSE values stops to significant decrease at Top 10 and its mean based on Sharpe ratio value is the highest, hence the method of selecting Top 10 most correlated technical indicators is chosen as the optimal method.

6. Decision tree properties

Parameter m in C5.0 decision tree sets the degree to which the initial tree can fit the data. At each branch point in the decision tree, the stated minimum number of training cases must follow at least two of the branches. This means, that the higher parameter m value is, the smaller tree will be made as many cases would need to follow to same branches. This can be seen in figure 19, where on x-axis are different tested parameter m values.

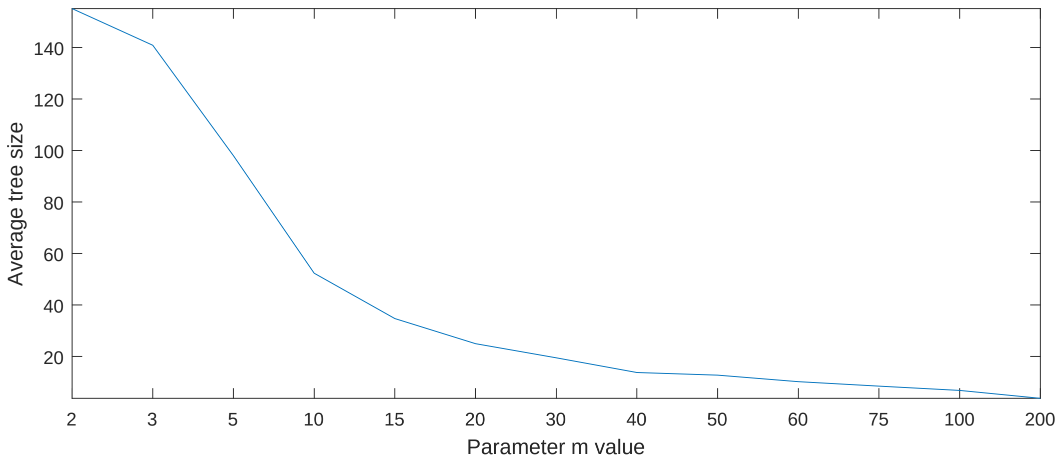

Figure 19. Tree size dependency on parameter m values.

Tree size is very important for overfitting. Using default parameters Sharpe ratio in training data reaches up to 4 (extremely good value) if tree size is around 100, however Sharpe ratio of validation data is around zero. Hence, tree size must be reduced to lower overfitting in training data and to increase validation and testing data performance.

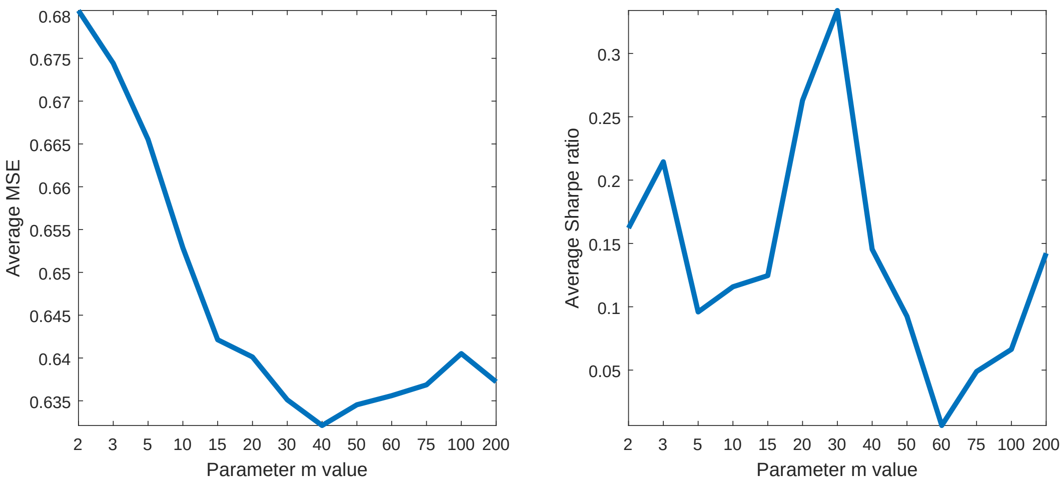

Figure 20. Results of different parameter m values for validation data.

Default m parameter is 2. In figure 20 tests results are presented. Based on MSE statistics best parameter value is between 30 and 50. Based on Sharpe ratio optimal value is 30. Therefore 30 is taken as optimal. As seen from figure 19, average tree size drops below 20, then parameter m value is 30.

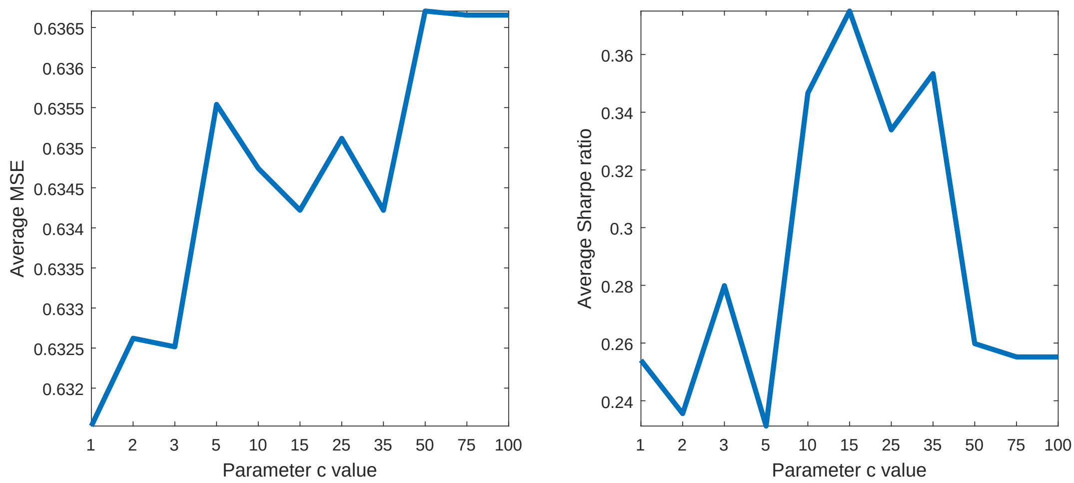

Figure 21. Results of different parameter c values for validation data.

Another parameter which affects tree size and reduces overfitting is severity of pruning. The parameter c affects the way that error rates are estimated and hence the severity of pruning; values smaller than the default (25%) cause more of the initial tree to be pruned, while larger values result in less pruning. In figure 21 results are presented. The optimal parameter is 1 based on MSE statistic, however based on Sharpe value optimal parameter value is 15. Notice that the MSE change between average results is less than 0.003 between parameter 1 and parameter 15 - almost non-existent. Therefore, optimal parameter is taken based on Sharpe value ($c = 15$).

Larger than default parameter m value and smaller than default parameter c value indicates, that tree size should be significantly smaller than the one that normal training for typical data would make as default values are usually tuned to be the best ones on average in various researches.

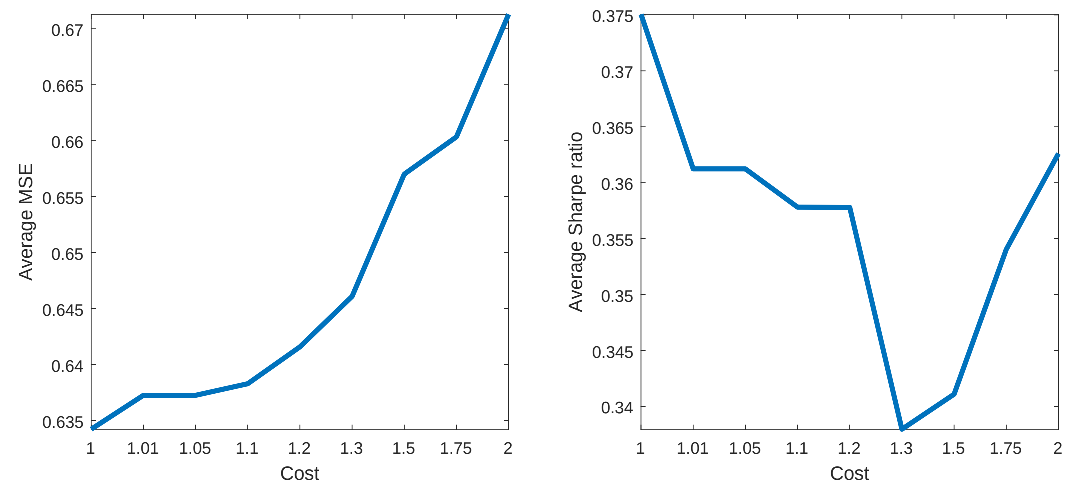

Figure 22. Results of different error costs for validation data.

One of the ways that C5.0 differs from C4.5 is that C5.0 can implement different costs to different mistakes. For example, it can be set that predicting ”sell short” would be costlier mistake than predicting ”flat” when the true prediction should be ”long”. This forces algorithm to avoid certain mistakes. Default cost is equal to one, whereas if prediction is correct, the cost is zero. The most important and costly in term of money losses from placing wrong trade are mistakes when ”sell short” is predicted but actual value is ”long” or vice versa. Hence, at first only the cost of those two mistakes is increased and tested. In figure 22 results of such test results for validation data are shown. Based on results optimal value is 1 (default - all errors costs are equal). If error is even slightly increased, then C5.0 decision tree performs worse on validation data. No other tests of error cost are done because if an increase of the most important errors do not improve results than any other less important error should not be made costlier.

7. Artificial neural network and decision tree results

In this section only results of testing data set are given. Testing data was not used in any way for neural network or decision tree parameter optimization in sections 5 and 6.

Out of 45 decision trees, one of them is just with one leave (size $= 1$). In such case only "sell short", or only "long", or always "flat" is predicted. Such tree is considered unsuccessful, because it does not use any technical indicators, therefore does not find any particular strategy to trade. The reason for such a small tree is that none of the given data (technical indicators) were informative enough about the desired output. The tree of size $= 1$ was generated for JY financial instrument. An ANN for JY is -0.11. This suggest that maybe the results of the decision tree can help to improve ANN. This idea will be exploited in the next section.

To get more accurate comparison JY financial instrument is left out.

For every financial instrument there is one decision tree grown. In contrast, ten neural networks are made with different initial weights. Then based on training data Sharpe ratio, the best of ten networks is selected and used to compare against decision tree result. In such way the risk that neural network results will be negatively affected by unsuccessfully selected initial weights using SAND algorithm should be greatly reduced.

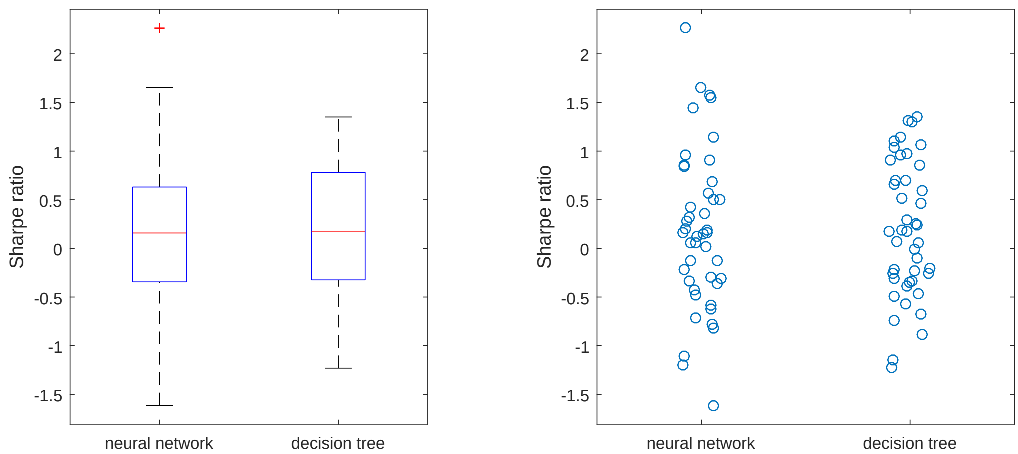

Figure 23. 44 out of 45 financial instruments results for testing data. Points in the right graph are jittered.

Results of remaining 44 financial instruments are shown in figure 23. Both median Sharpe ratio and mean Sharpe ratio are slightly better for decision tree but difference is minimal: mean in 0.1860 and median is 0.1768 with decision tree, whereas neural network mean is 0.1786, median is 0.1583.

A really good Sharpe value result is considered to be around or above 1. Out of 44 financial instruments for neural network Sharpe value for training data surpasses Sharpe ratio equal to one 6 times, whereas decision tree is above one 7 times. Sharpe ratio of 0.75 is surpassed 10 times by neural network and 11 times by decision tree.

These results show the importance of selecting the right financial instrument as only approximately 10 out of 45 gives satisfiable results for both ANN and DT. This is not a surprising result.

It is not expected that those 30 used technical indicators would be suitable for all 45 financial instruments. Neither decision tree, nor ANN can make a good predictive model if given input data is insufficient. Keep in mind that in this section testing results are analysed. It is incorrect to select the best financial instruments based on training results and compare them to determine which model is the best. A short example of prediction models selection is made in section 9.

The selection of the best predictive models to be used for trading is the portfolio construction problem. In real world, the ambiguity exists because of uncertainty and the lack of efficient information. Therefore, portfolio selection problem is a challenging problem for researchers [KA16]. No complex methods are used in this research to construct the best portfolio out of created predictive models as it is not the goal of this research. Only a simple approach of using all predictive models for portfolio construction is applied. It does not mean that comparison of two portfolios made using all prediction models is inadequate. Such comparison still indicates which method is the better one.

A portfolio must be formed by equally weighting predictive models share in portfolio. One futures financial instrument contract size differs for different futures financial instruments. For example, the contract size for a Canadian dollar (CD) futures contract is 100,000 Canadian dollars, while the size of a soybean (S) contract is 5,000 bushels. Hence, if portfolio consisted only of one CD and one S future then the asset allocation would not necessarily be 50% to 50%. Uneven asset allocation versus equal asset allocation has a very significant impact on the results of portfolio. In appendix 6 more details about how to construct a portfolio with an equal risk allocation is given; a significance of asset allocation is shown as well.

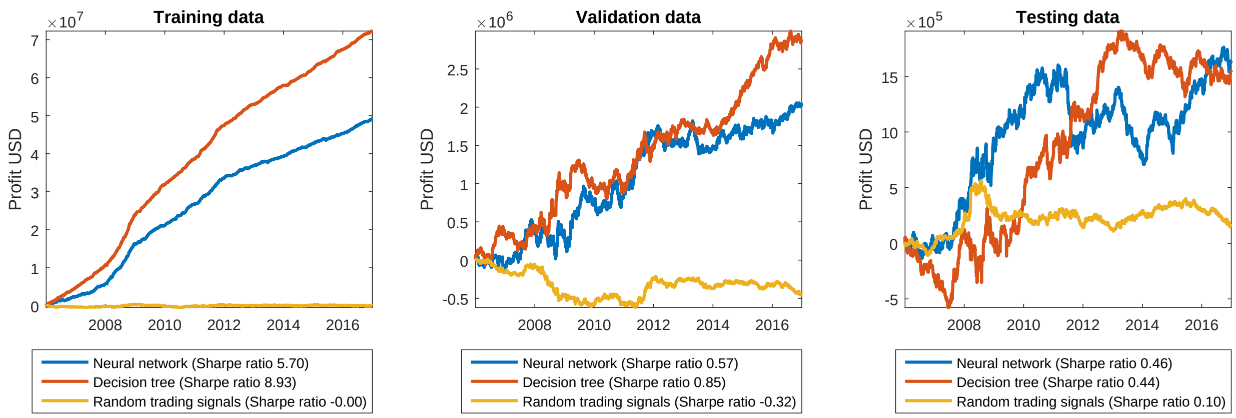

Figure 24. Portfolio results.

In figure 24 results are presented of equally weighted portfolios made from neural network prediction models, decision tree prediction models, and a portfolio where trades are generated randomly. Keep in mind that training data contains 70% of days, whereas validation and testing data has 15% each. Results shows the importance of using training, validation, and testing data sets. Even though decision tree portfolio does perform better with training and validation data, testing data results are almost the same as the portfolio with neural network models. Sharpe ratio for ANN and DT portfolios is 0.4607 and 0.4387 accordingly. They both earn almost the same, also performance volatility is similar, hence Sharpe ratio is similar. Both ANN and DT portfolios performs a lot better than the portfolio of randomly generated trading signals. It suggests that created models predictions are not random.

Table 6. Testing data results for every financial instrument.

Future abbreviation	AD	BO	BP	C	CC	CD	CL	DX	EC	ED	EMD
ANN Sharpe ratio	-0.3592	1.1452	-0.5861	0.1637	-1.1955	-1.1076	-1.6134	0.5076	0.9088	2.2642	-0.6194
DT Sharpe ratio	-0.3381	-1.1528	0.7048	-0.4950	-0.7406	-0.8907	0.5137	-0.0085	-0.2074	0.9765	0.0525
ANN accurate prediction %	47.55	54.24	50.00	53.79	45.10	48.91	46.06	52.26	54.91	60.20	51.47
DT accurate prediction %	48.07	49.22	55.75	50.00	47.99	49.80	57.35	52.05	48.13	53.29	53.64

Future abbreviation	ES	FDAX	FESX	FGBL	FGBM	FGBS	FSMI	FV	GC	HG	HO
ANN Sharpe ratio	0.9607	0.2028	0.1903	-0.4284	0.5712	0.6901	0.3560	1.5522	0.0555	-0.2943	-0.3120
DT Sharpe ratio	1.0674	0.1788	0.2354	1.0988	-0.2612	0.0678	1.1492	0.5943	-0.3505	0.4587	-0.5770
ANN accurate prediction %	58.37	51.08	53.01	53.36	49.54	50.47	52.48	53.17	51.15	53.43	48.74
DT accurate prediction %	58.56	48.95	50.46	54.66	48.06	48.09	52.40	53.95	54.75	51.68	46.04

Future abbreviation	KC	LC	LH	MP1	NG	NQ	PA	PL	QM	RB	RR
ANN Sharpe ratio	-0.1248	-0.2194	0.0589	-0.7137	1.5722	0.0175	-0.8233	-0.4759	0.3214	0.8596	-0.7860
DT Sharpe ratio	0.9625	0.9108	1.3506	-0.3937	1.3119	-0.2302	-1.2311	0.1944	-0.6717	-0.4682	-0.3095
ANN accurate prediction %	51.44	49.28	50.68	49.32	55.43	51.37	44.76	49.62	53.95	52.25	46.64
DT accurate prediction %	53.33	56.77	55.73	52.17	57.94	54.32	44.55	52.36	46.20	48.05	49.80

Future abbreviation	S	SB	SF	SI	SM	TF	TU	TY	US	W	YM
ANN Sharpe ratio	-0.3290	0.1192	0.1549	-0.4197	0.1616	1.6526	0.2854	0.4985	0.8433	1.4451	-0.1323
DT Sharpe ratio	-0.2197	-0.2615	0.1748	0.2974	-0.1032	0.8581	1.2973	0.6543	0.2478	1.0343	0.7046
ANN accurate prediction %	46.73	53.56	52.57	54.11	48.21	55.88	48.83	51.43	56.99	55.44	51.52
DT accurate prediction %	50.58	51.74	46.67	50.40	50.78	51.56	56.50	53.52	53.33	50.80	56.12

In table 6 results of every 44 futures financial instruments is given for testing data. Accurate prediction % is calculated only for those predictions, which do not predict ”flat” and where accurate prediction is not ”flat”. So, for example, if accurate prediction % value is 53% it means that 53 times out of a hundred ”sell short” is predicted when actual value is ”sell short” or ”long” is predicted when actual value is ”long”. In other hand, 47 times out of a hundred ”sell short” is predicted when actual value is ”long” or ”long” is predicted when actual value is ”sell short”.

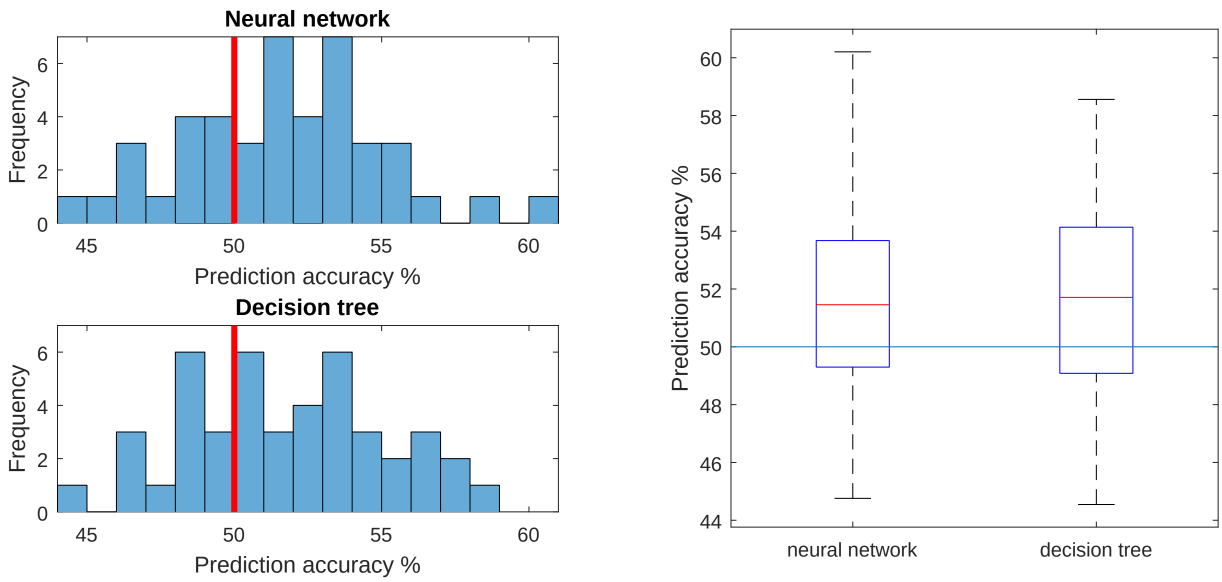

Figure 25. Prediction accuracy.

Distributions of accurate prediction % values are shown in figure 25. Distribution of accurate prediction % of random trading signals would be symmetrical with average 50%. Both neural network and decision tree distribution looks symmetrical, however in total they are more accurate than 50% (51.52% - neural network, 51.77% - decision tree). In table 6 green colour marks whether decision tree or neural network has a higher prediction accuracy. 23 times decision tree is more accurate, 21 times neural network has more precise prediction. Decision tree has a slightly heavier tail in the right. Prediction accuracy is higher than 56% 6 times in decision tree portfolio, whereas in neural network portfolio only 3 times.

Sharpe ratio has a high dependency on prediction accuracy. If accuracy is higher, then usually Sharpe ratio is also higher as seen from table 6 (blue colour marks which method has a higher Sharpe value). Both neural network and decision tree prediction models are superior to one another by Sharpe ratio 22 times each.

Prediction accuracy and Sharpe ratio results of ANN and DT models varies a lot for every financial instrument. Results are not always in favour for one of the methods. It shows the importance of using a large set of different financial instruments to draw conclusions. If for example, only TU, TY, YM futures were used in the research, when results would be misleading. For all of those futures DT predictions are clearly better than ANN predictions. Therefore, a conclusion would be made that DT is clearly better than ANN in general even though results of all 44 futures shows that it is not the case.

Based on all results, a conclusion can be made that both methods are quite similar. Constructed portfolio is slightly better of neural network predication models in terms of profit and Sharpe value. However, decision tree prediction models are more accurate on average and more times models prediction accuracy is higher than 56%. Decision tree is superior method because it can indicate that data might not be sufficient enough to make predictions. For JY futures financial instrument decision tree do not grow any tree. ANN for JY is also unsuccessful but ANN does not indicate from training data that results on testing data will be bad. Since results do not differ significantly, a decision tree should be used as this method is more informative to practitioners. Grown decision tree is easier to interpret than a neural network. Grown decision tree clearly shows how predictions are made. This information could help practitioners to create even better prediction models.

8. Artificial neural network and decision tree combinations

8.1. Inputs selection

C5.0 decision tree algorithm selects only those technical indicators which give the most information, hence ANN is not needed to select the best indicators from a 30 technical indicators set to be used with C5.0 algorithm. In contract, ANN should work better when more suitable technical indicators subset is selected from all the set. Results in section 5.3 shows that ANN gives better results when only the technical indicators which correlates with the output the most are selected. It supports this idea. In this section an input selection approach will be tested where ANN inputs will be selected based on learned decision tree results with the same training data.

44 out of 45 decision trees will be used to select inputs for ANN. Japanese Yen (JY) is excluded because none of the technical indicators were found to be suitable by C5.0 algorithm to classify outputs.

Every decision tree is grown using some subset of technical indicators. In average final 44 decision trees used 11.16 technical indicators (varies from 4 to 19). Let A_i be the subset of technical indicators which are included in the decision tree for i'th financial instrument. Test is made by learning artificial neural network only with inputs of the subset A_i. To get more accurate results 10 networks were learned for every of 44 financial instruments and the best one based on training data Sharpe ratio selected. Hidden neurons count was scaled based on the number of inputs.

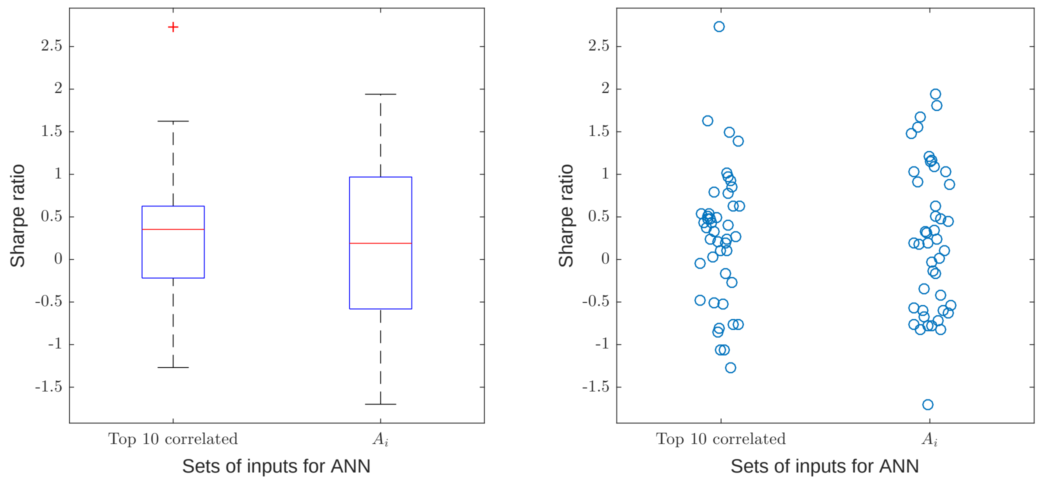

Figure 26. Result of different ANN input sets for validation data.

When deciding which inputs should be used, validation data results should be looked at instead of testing data results. In figure 26 Sharpe ratio distribution is shown. Even though third quarter is higher for ANN with A_i inputs, mean is better for neural network with top 10 correlated inputs (0.2651 against 0.2226).

A weighted portfolios results are given in figure 27. Total profit earned is higher when top 10 correlated inputs are used. In addition, variance is lower. Hence, Sharpe ratio is better.

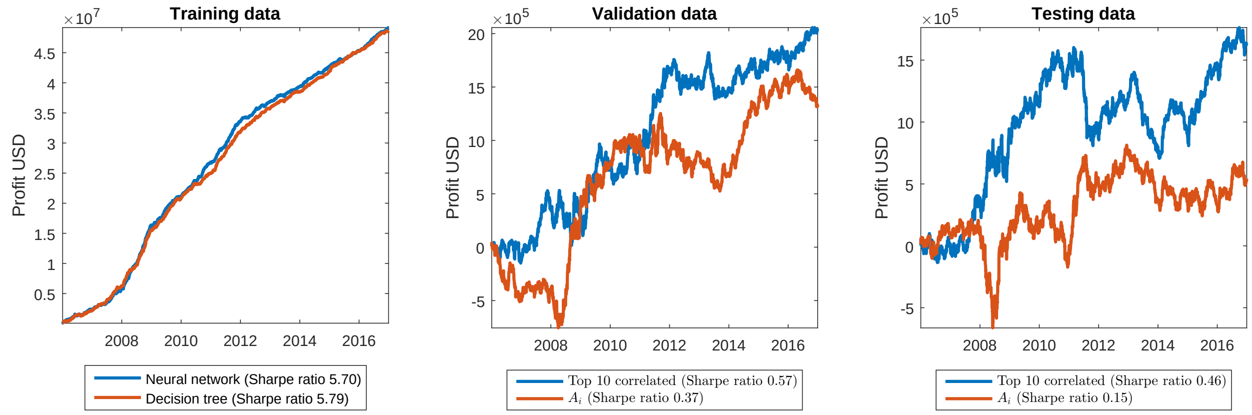

Figure 27. Portfolio results.

Prediction accuracies for validation data are given in figure 28 and table 7. Accurate prediction in percent is calculated only for those predictions, which do not predict "flat" and where the accurate prediction is not "flat". Total prediction accuracy of ANN with top 10 correlated inputs is 51.54%, whereas with inputs from A_i it is 51.67%. However, two times more ANN with top 10 correlated inputs has a higher Sharpe ratio and prediction accuracy than ANN with A_i inputs. Both method has the same number of prediction accuracies over 50% and over 56% - 29 and 5 accordingly. Even though results look similar on average, it differs a lot for every future financial instrument.

Table 7. Validation data results for every financial instrument.

Future abbreviation	AD	BO	BP	C	CC	CD	CL	DX	EC	ED	EMD
Top 10 correlated Sharpe ratio	0.4816	0.5342	0.9719	0.2732	-0.5098	-0.7644	0.4043	-0.4791	-0.1676	0.4947	1.3961
A_i Sharpe ratio	0.9115	-0.7230	0.1715	-0.5632	0.3440	-1.7004	0.1933	0.2441	0.4741	-0.3453	1.8017
Top 10 correlated accurate prediction %	55.64	51.16	51.61	51.08	50.10	49.81	54.86	51.54	49.82	48.39	54.90
A_i accurate prediction %	57.82	47.71	49.48	50.00	52.65	44.28	53.02	50.21	53.46	50.35	55.61

Future abbreviation	ES	FDAX	FESX	FGBL	FGBM	FGBS	FSMI	FV	GC	HG	HO
Top 10 correlated Sharpe ratio	-0.0394	-0.8147	-0.8566	2.7298	-0.2694	0.3269	0.6294	0.1008	0.5361	0.4847	-1.2693
A_i Sharpe ratio	0.3093	0.1029	-0.5985	1.6799	1.0853	-0.1319	1.1633	1.0298	-0.8185	-0.6048	1.0247
Top 10 correlated accurate prediction %	54.59	46.11	45.64	59.78	51.98	50.24	53.30	48.60	50.74	50.76	46.43
A_i accurate prediction %	55.21	51.09	46.92	58.80	57.07	48.29	55.61	54.65	43.40	50.22	53.49

Future abbreviation	KC	LC	LH	MP1	NG	NQ	PA	PL	QM	RB	RR
Top 10 correlated Sharpe ratio	-1.0695	0.3801	0.7974	0.1962	0.5088	0.9231	0.2399	-0.7656	0.4374	0.8530	1.6236
A_i Sharpe ratio	0.4481	0.6317	1.2154	-0.1649	-0.0265	1.9403	-0.7762	0.0073	1.1465	-0.5387	1.5498
Top 10 correlated accurate prediction %	48.56	53.33	52.74	56.25	48.93	60.67	50.20	43.92	49.77	53.40	57.21
A_i accurate prediction %	53.20	51.66	55.56	50.20	50.20	61.98	48.28	46.52	54.59	48.13	59.00

Future abbreviation	S	SB	SF	SI	SM	TF	TU	TY	US	W	YM
Top 10 correlated Sharpe ratio	0.0302	0.4376	-1.0684	1.5006	-0.5282	0.2027	0.1085	1.0172	0.7791	0.2432	0.6229
A_i Sharpe ratio	0.1864	-0.7627	1.4805	0.3331	-0.7741	-0.6706	-0.4263	0.5058	-0.8164	-0.6229	0.8780
Top 10 correlated accurate prediction %	52.38	52.48	46.44	54.55	47.45	56.86	47.10	54.08	54.10	52.09	54.21
A_i accurate prediction %	52.17	48.91	52.38	50.78	44.26	52.97	46.06	52.13	46.61	49.09	55.14

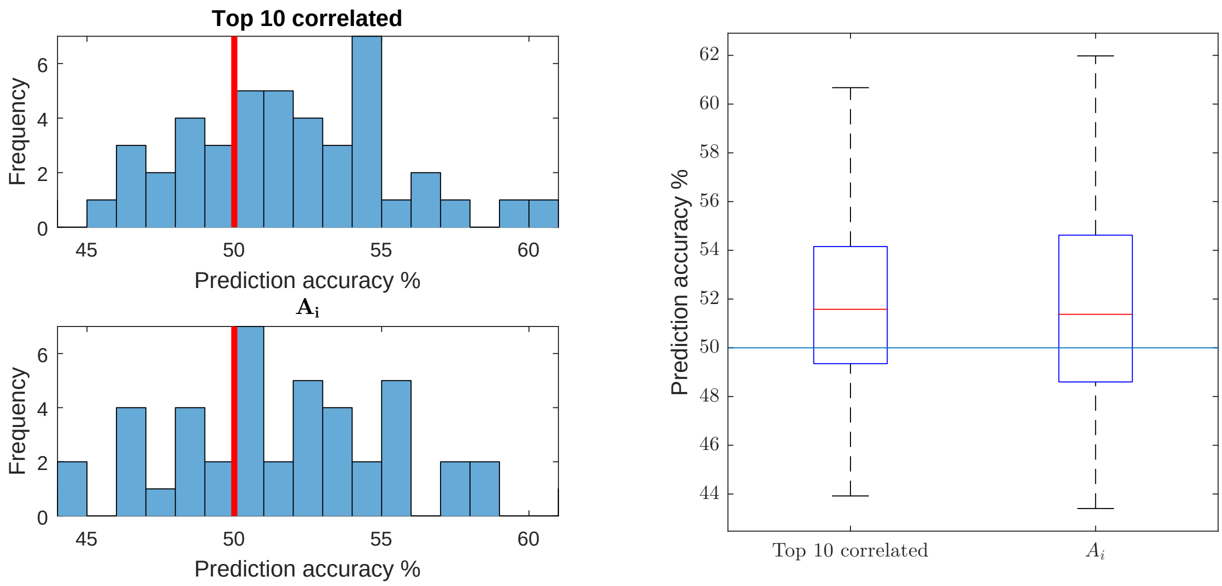

Figure 28. Prediction accuracy.

Different results of method of taking top 10 correlated technical indicator and method of using subset A_i suggest that different technical indicators were selected as input for ANN. If subset A_i is independent to the most correlated technical indicators then the probability that any of the technical indicators of subset A_i also belong to the subset of top 10 correlated technical indicators is equal to $\frac{1}{3}$. For example, if there are 9 technical indicators in subset A_i and if A_i is independent to subset of top 10 most correlated indicators, then 3 of them would belong to the both subsets on average. This means that in total for 44 financial instruments approximately 163 technical indicators would belong to the both subsets on average. However, in total 189 (more) technical indicators belongs to the both subsets. It indicates that A_i might not be independent to the subset of top 10 correlated technical indicators. Results show that technical indicator is more likely to be used in a decision tree if it is one of the top 10 most correlated technical indicators to the outputs. If every technical indicator of subset A_i (or exactly 10 if there are more or equal to 10 technical indicators in subset A_i) would also belong to top 10 most correlated technical indicators subset, then 378 technical indicators would belong to the both subsets if dependency would be ideal. Therefore, even though some dependency exists, it is not strong.

In conclusion, these methods use quite different inputs for ANN, however results are not so different when the whole portfolio results are compared. Method of using top 10 correlated technical indicators results gives a better portfolio in terms of Sharpe value. Also prediction accuracy is rather similar to A_i. There is no evidence that method of using technical indicators subset A_i is superior to method of using top 10 correlated technical indicators. Therefore, a simpler method should be used (ANN with top 10 correlated technical indicators). This method does not require a decision tree to be grown, hence is easier to implement.

8.2. Portfolio with both predictive methods combinations

In section 7 different portfolio for both prediction models were made. In this section a portfolio using both predictive methods is constructed. Daily returns correlation between neural network portfolio and decision tree portfolio for validation data is only 0.2334. The portfolio construction/diversification approach is based on the notion of asset segmentation, which can be based on: similarity of asset types, risk arguments mainly based on security risk and correlation, or similarity of asset dependency on macroeconomic factors [Pol14]. Low correlation between two portfolios returns means that a portfolio with lower volatility can be constructed. It results in a higher Sharpe value, hence is more appealing for practitioners.

In section 7 1,100,000 US dollars where used as investment asset (more information in appendix 6). A portfolio of combined methods is constructed in a such way: half of assets are used to trade neural network models and half to trade decision tree models. Let's denote this new portfolio as COMB1.

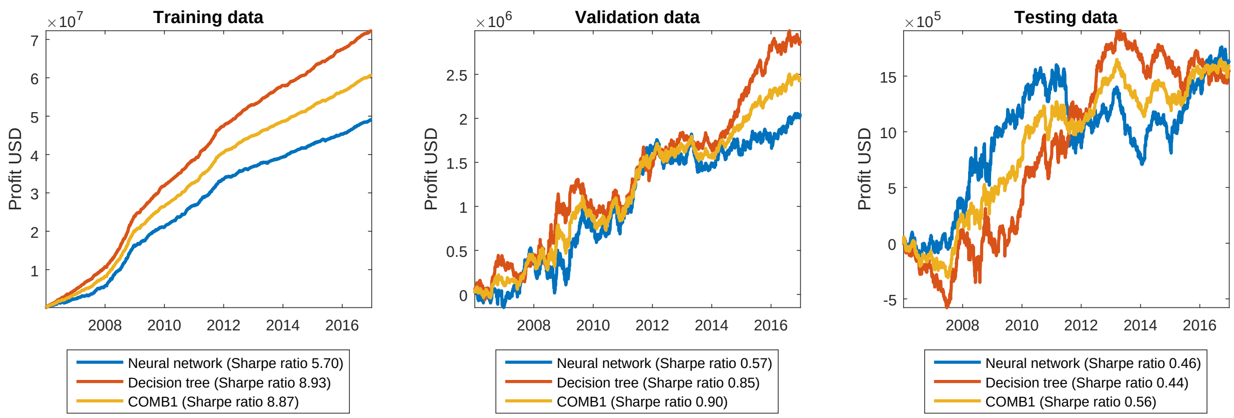

Figure 29. Portfolio results.

Results are presented in figure 29. As expected, a combination of two weakly correlated portfolios gives a better risk to return ratio (Sharpe ratio). Even though total profit earned does not improve, volatility decreases, hence Sharpe ratio is higher.

The fact that neural network and decision tree portfolios returns has only a weak correlation allows to build a more effective portfolio. Hence, if possible, portfolio of both neural network and decision tree prediction models should be used.

Prediction accuracy of the portfolio COMB1 changes because some predictions changes. Let's say that in the n'th day the neural network would buy gold future financial instrument. Let's assume that in the same n'th day the decision tree would short sell gold future financial instrument. In a such case, COMB1 would decide to buy and sell short gold at the same time. Hence, neither buy nor short sell would happen. In other words, combination of portfolios would predict flat. Because of such cases, prediction accuracies changes. In similar situation, COMB1 could for example buy gold future financial instrument when decision tree predicts flat, but neural network predicts long.

COMB1 predicts "buy", "sell short", or "flat" for every financial instrument in a such way (in case when one future financial instrument is bought/short sold per predictive model):

- If DT and ANN predicts ”flat”, then COMB1 does nothing.
- If DT predicts ”flat”, but ANN predicts ”long”, then COMB1 buys 1. If DT predicts ”flat”, but ANN predicts ”sell short”, then COMB1 sell shorts 1. If DT predicts ”long”, but ANN predicts ”flat”, then COMB1 buys 1. If DT predicts ”sell short”, but ANN predicts ”flat”, then COMB1 sell shorts 1.
- If DT predicts ”sell short”, but ANN predicts ”long”, then COMB1 does nothing. If DT predicts ”long”, but ANN predicts ”sell short”, then COMB1 does nothing.
- If DT and ANN predicts ”long”, then COMB1 buys 2. If DT and ANN predicts ”sell short”, then COMB1 sell shorts 2.

Results for validation data for ANN, DT, and COMB1 portfolios are given in appendix table A4. COMB1 is only slightly better in terms of total prediction accuracy of portfolio. COMB1 prediction accuracy is 51.87%, ANN - 51.59%, DT - 51.67%. Also, Sharpe ratios are the best for COMB1 only 5 out of 44 times. So, even though prediction accuracies change a bit, no clear improvement over neural network or decision tree portfolio in terms of prediction accuracy or Sharpe ratio can be seen when futures results are compared one by one. However, portfolio COMB1 is still better because of decreased volatility of the portfolio.

Let’s try another portfolio construction method using both methods. Let’s denote it as COMB2. COMB2 is constructed in the same way as COMB1 with only one difference: when DT and ANN prediction mismatch (”long” versus ”sell short” or ”long” versus ”flat” or ”sell short” versus ”flat”) then COMB2 predicts ”flat” and does nothing. The idea is to buy or sell short only if the prediction is strong - both ANN and DT agrees on the prediction. Keep in mind that this method decreases the number of days when either buy or sell short happens. However, it is not an issue as more future financial instruments can be bought or sold short in order to maintain similar trading activity on average.

Results of COMB1 and COMB2 portfolio for validation data are given in table 8. COMB2 gives a much better prediction accuracy than COMB1. 27 out of 44 times COMB2 accuracy is higher than the one of COMB1. Also, Sharpe ratio is higher with portfolio COMB2 30 out of 44 times. For portfolio COMB2 the number of trades decreases 40.49% compared to COMB1, 48.96% compared to ANN portfolio, and 44.90% compared to DT portfolio for validation data. However, for COMB2 total prediction accuracy is significantly better compared to other portfolios. COMB2 prediction accuracy is 53.02%, whereas COMB1 accuracy is 51.87%, ANN portfolio accuracy is 51.59%, and DT portfolio accuracy is 51.67%. Based on validation data results COMB2 is superior to COMB1. Therefore, COMB2 is selected for comparison against ANN and DT portfolios for testing data.

In table 9 COMB2 portfolio results for testing data are given. These results can be compared against ANN and DT portfolios results for testing data given in table 6 (section 7). For more clarity, table cells are coloured in a such way: cell is blue if COMB2 Sharpe ratio is higher than ANN and DT Sharpe ratio; cell is green if COMB2 prediction accuracy is higher than ANN and DT accuracy; cell is orange if either Sharpe ratio or prediction accuracy is higher than the one of either ANN or DT result; cell is white if COMB2 result is worse than both ANN and DT results.

Table 8. Validation data results for every financial instrument.

Future abbreviation	AD	BO	BP	C	CC	CD	CL	DX	EC	ED	EMD
COMB1 Sharpe ratio	0.6593	0.0901	-0.1859	0.8567	-1.0977	-0.5987	0.7560	-0.0879	0.1702	0.0485	1.6356
COMB2 Sharpe ratio	0.7421	0.6172	0.0613	0.9167	-0.8791	-0.4847	0.7755	0.2182	0.5473	-0.1035	1.6071
COMB1 accurate prediction %	57.71	47.26	46.76	52.28	48.03	51.17	54.15	52.49	51.89	50.60	57.09
COMB2 accurate prediction %	58.29	46.77	49.17	54.09	47.87	50.76	55.83	54.43	56.41	46.94	57.39

Future abbreviation	ES	FDAX	FESX	FGBL	FGBM	FGBS	FSMI	FV	GC	HG	HO
COMB1 Sharpe ratio	-0.0593	-0.3505	-0.7705	2.3897	0.1945	0.0397	-0.0744	0.1614	0.1470	-0.1266	-0.3533
COMB2 Sharpe ratio	0.3105	-0.0304	-0.9121	2.5672	0.6427	1.6010	-0.5932	0.6409	-0.3009	0.2759	-0.1231
COMB1 accurate prediction %	54.97	44.30	47.31	60.53	52.43	48.53	50.00	49.03	48.98	50.26	48.88
COMB2 accurate prediction %	57.25	45.45	45.38	63.16	54.29	57.14	47.13	52.54	42.86	49.57	47.47

Future abbreviation	KC	LC	LH	MP1	NG	NQ	PA	PL	QM	RB	RR
COMB1 Sharpe ratio	-0.7386	0.8761	0.8870	0.0407	0.4733	1.9876	0.1944	-1.1426	-0.0536	0.9291	2.1107
COMB2 Sharpe ratio	-0.1341	0.7804	0.6968	0.0831	0.1579	2.1762	0.2019	-0.7981	1.2615	0.9451	1.6492
COMB1 accurate prediction %	47.87	54.40	53.98	53.85	51.50	61.17	50.90	42.33	47.20	56.47	57.66
COMB2 accurate prediction %	50.88	53.85	55.00	54.94	50.00	65.29	55.45	44.55	54.26	57.14	58.64

Future abbreviation	S	SB	SF	SI	SM	TF	TU	TY	US	W	YM
COMB1 Sharpe ratio	-0.0169	0.0799	-1.2411	1.3459	0.3560	0.6076	-0.5042	1.5255	1.2452	0.4838	0.4376
COMB2 Sharpe ratio	-0.2732	0.4101	-1.4681	1.6217	0.3300	0.3393	0.0569	1.2760	1.4069	0.6619	0.9568
COMB1 accurate prediction %	52.11	50.24	45.97	54.97	50.79	58.25	44.89	55.79	55.43	53.69	53.85
COMB2 accurate prediction %	50.38	50.74	42.31	56.60	50.75	56.64	42.86	54.34	55.98	54.55	56.60

Prediction accuracy of COMB2 is higher than the one of ANN or DT or both 38 times out of 44. 17 times accuracy surpasses both ANN and DT models accuracy. In total for testing data predictions are the most accurate with COMB2 portfolio (52.48%). ANN accuracy is 51.52%, DT accuracy is 51.77%. Predictions over 56% accuracy are made 10 times with COMB2, whereas with ANN it is only 6, DT - only 3. However, accuracies below 45% are also more common. COMB2 models accuracy is below 45% 5 times, whereas ANN and DT models only 2 times each. In appendix figure A4 it is shown how COMB2 accuracy compares to ANN and DT accuracy by distribution.

Table 9. Testing data results for every financial instrument.

Future abbreviation	AD	BO	BP	C	CC	CD	CL	DX	EC	ED	EMD
COMB2 Sharpe ratio	-0.22	-0.10	-0.37	-0.20	-0.74	-1.22	-0.72	0.77	0.48	2.10	-0.30
COMB2 accurate prediction %	48.58	51.90	51.47	54.07	47.15	48.70	50.91	57.32	53.00	62.75	52.60

Future abbreviation	ES	FDAX	FESX	FGBL	FGBM	FGBS	FSMI	FV	GC	HG	HO
COMB2 Sharpe ratio	0.99	0.23	0.18	0.82	0.30	0.49	1.16	1.16	0.15	0.19	-0.79
COMB2 accurate prediction %	61.90	52.24	52.42	58.99	50.43	48.09	56.25	56.41	56.20	54.03	44.33

Future abbreviation	KC	LC	LH	MP1	NG	NQ	PA	PL	QM	RB	RR
COMB2 Sharpe ratio	0.27	-0.17	0.90	-0.56	1.14	-0.08	-1.15	-0.48	0.10	0.38	-0.63
COMB2 accurate prediction %	50.43	53.21	56.31	51.72	56.62	53.78	41.09	48.62	49.48	50.65	47.90

Future abbreviation	S	SB	SF	SI	SM	TF	TU	TY	US	W	YM
COMB2 Sharpe ratio	0.21	-0.16	0.38	0.56	-0.45	1.27	1.24	0.39	0.50	1.53	0.48
COMB2 accurate prediction %	51.49	54.29	52.38	54.13	45.00	53.73	55.37	53.25	55.77	55.91	56.03

Sharpe ratio data distribution is shown in appendix figure A5. Sharpe ratios of COMB2 prediction models are higher than ANN or DT or both 37 times out of 44. 10 times Sharpe ratio surpasses both ANN and DT models Sharpe ratios.

To compare COMB2 portfolio profit results with ANN and DT portfolios results, COMB2 should be scaled at first. As mentioned before, COMB2 trading frequency is significantly lower than the one of ANN and DT portfolios. In order to level out trading frequencies, COMB2 portfolio should trade more of the same future financial instruments. Let's say that the trading level of COMB2 on average is half of the ANN and DT trading level. Let's say that all gold future financial instrument predictive models decide to buy 5 gold futures. Then, COMB2 portfolio should buy 10 gold futures financial instruments instead of 5, because COMB2 is twice as less likely to buy gold futures financial instruments in general because of lower average trading level. Note, that prediction accuracies and Sharpe ratios given in table 8 for every single future financial instrument do not change if more or less future contracts are bought or sold short. Training data trading levels are used to determine quantities of trading in order to even out trading levels.

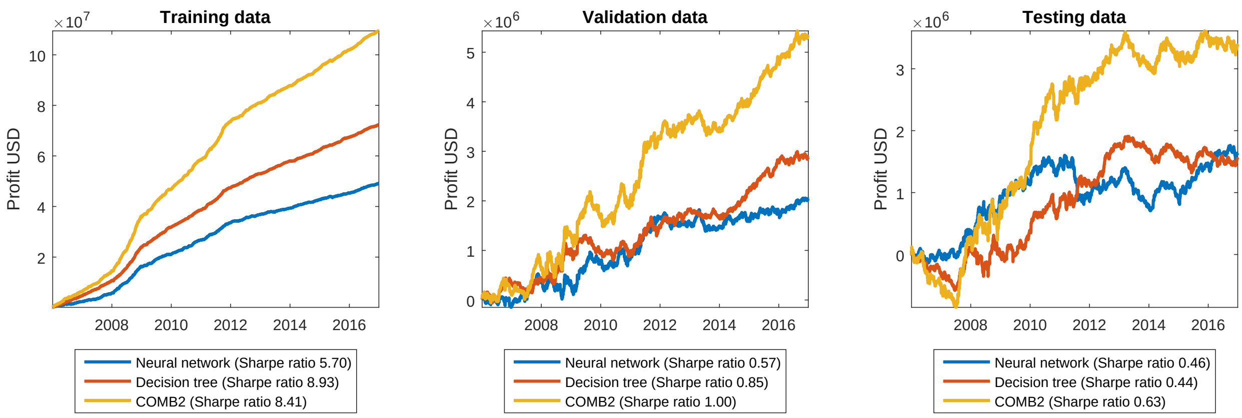

Figure 30. Portfolio results.

Portfolios results are shown in figure 30. Sharpe ratio of COMB2 portfolio for testing data is the highest (0.63).

Based on prediction accuracies, Sharpe ratios for every single future financial instrument, and portfolio results, COMB2 prediction models shows better results than solely neural network prediction models using top 10 correlated technical indicators and solely decision tree prediction models. Hence, COMB2 should be used by practitioners.

9. Selection of future financial instruments

In this section an example of futures selection for portfolio construction is given. The goal is to show that some improvements for portfolios can be made. Only one example is shown as the main goal of this paper is to compare different prediction methods, not to construct the best portfolio possible for trading.

Results presented in section 7 suggests that ANNs and DTs does not give similar results for all 45 future financial instruments used in this research. With some futures, results are significantly worse than with the others. This means that if a practitioner could select only those futures with which results are the best, then a portfolio would be better in terms of profit and prediction accuracy.

A few selection rules were tested. One is to select top 5 future financial instruments for which Sharpe ratio for training data is the best. Another rule is to select it based on the best prediction accuracies. Validation data is used to determine which rule to use. Based on validation data results, for neural network it is best to select top 5 future financial instruments using prediction accuracy. For decision tree - Sharpe ratio.

In order to compare newly constructed portfolio with the ones from section 7, trading levels are evened up. Results of portfolios are shown in figure 31.

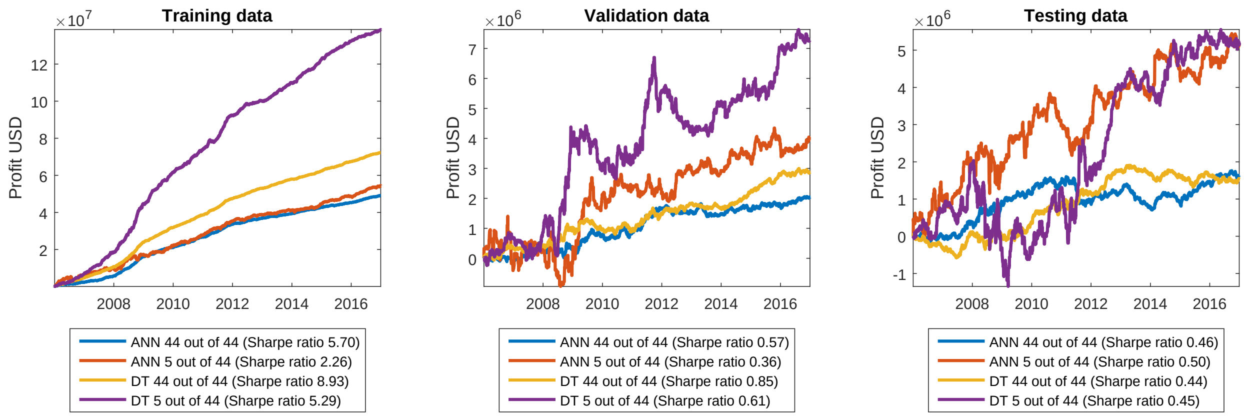

Figure 31. Portfolio results.

Diversification of the portfolio decreases, if only 5 out of 44 financial instruments are used. Hence, volatility increases. Because of it, Sharpe ratios are lower as seen in figure 31. In other hand, portfolios with only 5 filtered futures financial instruments are more profitable for the same trading level. Because of it, these new portfolios Sharpe ratios are similar to the ones with all 44 futures for testing data. New portfolios are superior because they give the same risk to reward ratio (Sharpe ratio) but with the same trading level are more profitable. Also decision tree portfolios with selected 5 futures predicts much more accurately than the one with 44 futures for testing data (53.16% against 51.77% accordingly).

An example given in this section shows that a further improvement is possible in order to get better and more profitable portfolios. A more detailed research of portfolio construction/optimization methods is planned as a future work.

10. Conclusion

This paper summarizes massive amount of 16,895 experiments. Predictions using testing data were performed on 45 different futures financial instruments to determine whether artificial neural network or C5.0 decision tree is more suitable for financial data prediction of price movement for one day ahead. It is shown that a performance of artificial neural network and C5.0 decision tree is quite similar. Such conclusion is made based on prediction accuracies and Sharpe ratios of all 45 futures as well as on performance of the portfolios constructed with artificial neural network and C5.0 decision tree prediction models. An important observation for practitioners is that constructed portfolios has a rather low correlation with one another.

If only one of the prediction methods must be chosen, then the C5.0 decision tree algorithm should be used. This method can indicate if the data might not be sufficient enough to make accurate predictions.

Two combination types of both prediction methods are also proposed. It is shown that better results in term of profitability and prediction accuracy can be achieved using combined approach. Proposed method COMB2 should be used instead of solely neural network or C5.0 decision tree if possible. In this paper it is also shown that results of testing data sample heavily depend on the time period of testing data sample. Such dependency arises from changing market conditions. Tests suggest that a method of taking randomly selected testing data sample makes a good generalization of various testing data samples and should be used to get unbiased results. This conclusion should hold to any research using financial data and not only when neural networks or decision trees are used.

In addition, it is shown that MSE value can be used to train neural network even then the actual goal is to maximize Sharpe ratio of the trading strategy that uses trained neural network model. MSE and Sharpe ratio has rather high correlation.

Also it is shown that genetic algorithm might not be always suitable to select which features to use as inputs for artificial neural networks. Even though there are numerous papers, which shows that genetic algorithm is successful in selecting a subset of all possible inputs for artificial neural networks, tests in this paper suggest that it might not be the case when noisy data is used as the input set.

Moreover, this paper shows that in analysing financial data it is extremely important to analyse a large set of data samples and make conclusions only out of all of them. Simulations results varies a lot from instrument to instrument. It is shown that if only a few of them would be analysed, results might be heavily misleading.

As a future work, more effort should be made to construct portfolios. An example is given on how performance can be increased if only some of the future financial instruments are added to the portfolio. More analysis should be made to determine the best way to filter future financial instruments. In addition, in order to prepare trading strategies to be used in real life, trailing stops, stop losses, take profits, and similar traders tools should be tested to improve profitability of the portfolio.

References

[Alt] Altegris. The altegris group. `http://www.managedfutures.com/managed_futures_index.aspx`. Accessed: 2016-12.

[AS09] Matthew N Anyanwu and Sajjan G Shiva. Comparative analysis of serial decision tree classification algorithms. *International journal of computer science and security*, 3(3):230–240, 2009.

[Bar] Barclay. Barclay hedge, ltd. `https://www.barclayhedge.com/research/indices/cta/mum/Systematic_Traders.html`. Accessed: 2017-02.

[BK92] Egbert JW Boers and Herman Kuiper. Biological metaphors and the design of modular artificial neural networks, 1992.

[CC09] Chun-Lang Chang and Chih-Hao Chen. Applying decision tree and neural network to increase quality of dermatologic diagnosis. *Expert systems with applications*, 36(2):4035–4041, 2009.

[CM94] Stephen P Curram and John Mingers. Neural networks, decision tree induction and discriminant analysis: an empirical comparison. *Journal of the operational research society*, 45(4):440–450, 1994.

[CME] CME. Countdown to the fomc. `http://www.cmegroup.com/trading/interest-rates/countdown-to-fomc.html`. Accessed: 2016-05-25.

[CV01] Leandro Nunes de Castro and Fernando J Von Zuben. An immunological approach to initialize feedforward neural network weights. In *Artificial neural nets and genetic algorithms*. Springer, 2001, pp. 126–129.

[CV98] Leandro Nunes de Castro and Fernando José Von Zuben. A hybrid paradigm for weight initialization in supervised feedforward neural network learning. *Proc. of the ics'98*:30–37, 1998.

[DKU13] Dursun Delen, Cemil Kuzey, and Ali Uyar. Measuring firm performance using financial ratios: a decision tree approach. *Expert systems with applications*, 40(10):3970–3983, 2013.

[FL01] Adam Fadlalla and Chien-Hua Lin. An analysis of the applications of neural networks in finance. *Interfaces*, 31(4):112–122, 2001.

[GKD11] Erkam Guresen, Gulgun Kayakutlu, and Tugrul U Daim. Using artificial neural network models in stock market index prediction. *Expert systems with applications*, 38(8):10389–10397, 2011.

[HDY+12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal processing magazine, ieee*, 29(6):82–97, 2012.

[HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Elsevier, 2011.

[HW00] Lonnie Hamm and B Wade Brorsen. Trading futures markets based on signals from a neural network. *Applied economics letters*, 7(2):137–140, 2000.

[KA16] Fusun Kucukbay and Ceyhun Araz. Portfolio selection problem: a comparison of fuzzy goal programming and linear physical programming. *An international journal of optimization and control: theories & applications (ijocta)*, 6(2):121–128, 2016.

[KB96] Iebeling Kaastra and Milton Boyd. Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3):215–236, 1996.

[KC00] Cheng-Wen Ko and Hsiao-Wen Chung. Automatic spike detection via an artificial neural network using raw eeg data: effects of data preparation and implications in the limitations of online recognition. *Clinical neurophysiology*, 111(3):477–481, 2000.

[KFA01] Ernst Kretschmann, Wolfgang Fleischmann, and Rolf Apweiler. Automatic rule generation for protein annotation with the c4. 5 data mining algorithm applied on swiss-prot. *Bioinformatics*, 17(10):920–926, 2001.

[KR91] YK Kim and JB Ra. Weight value initialization for improving training speed in the backpropagation network. In *Neural networks, 1991. 1991 ieee international joint conference on*. IEEE, 1991, pp. 2396–2401.

[KSP01] Samuel Kaski, Janne Sinkkonen, and Jaakko Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *Neural networks, ieee transactions on*, 12(4):936–947, 2001.

[LCC96] Yinghua Lin, George A Cunningham, and Stephen V Coggeshall. Input variable identification—fuzzy curves and fuzzy surfaces. *Fuzzy sets and systems*, 82(1):65–71, 1996.

[LPR02] William Leigh, Russell Purvis, and James M Ragusa. Forecasting the nyse composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decision support systems*, 32(4):361–377, 2002.

[LSK+95] Mikko Lehtokangas, Jukka Saarinen, Kimmo Kaski, and Pentti Huuhtanen. Initializing weights of a multilayer perceptron network by using the orthogonal least squares algorithm. *Neural computation*, 7(5):982–999, 1995.

[Mas93] Timothy Masters. *Practical neural network recipes in c++*. Morgan Kaufmann, 1993.

[MP43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[Ngu90] D Nguyen. E widrow b. improving the learning speed of 2-layer neural networks by choosing initial values of the adaptatives weights. 1990.

[NI91] Marilyn McCord Nelson and William T Illingworth. *A practical guide to neural nets*. Vol. 1. Addison-Wesley Reading, MA, 1991.

[OM03] Dennis Olson and Charles Mossman. Neural network forecasts of canadian stock returns using accounting ratios. *International journal of forecasting*, 19(3):453–465, 2003.

[OM06] Niall O'Connor and Michael G Madden. A neural network approach to predicting stock exchange movements using external factors. *Knowledge-based systems*, 19(5):371–378, 2006.

[PCH+03] Zhi-Song Pan, Song-Can Chen, Gen-Bao Hu, and Dao-Qiang Zhang. Hybrid neural network and c4. 5 for misuse detection. In *Machine learning and cybernetics, 2003 international conference on*. Vol. 4. IEEE, 2003, pp. 2463–2467.

[PEL99] Jose C Principe, Neil R Euliano, and W Curt Lefebvre. *Neural and adaptive systems: fundamentals through simulations with cd-rom*. John Wiley & Sons, Inc., 1999.

[Pol14] Gianni Pola. Is your portfolio effectively diversified? various perspectives on portfolio diversification. Tech. rep. Amundi Working Paper WP-040-2014, 2014.

[Pol90] Jordan B Pollack. Backpropagation is sensitive to initial conditions. *Complex systems*, 4:269–80, 1990.

[QS99] Tong-Seng Quah and Bobby Srinivasan. Improving returns on stock investment through neural network selection. *Expert systems with applications*, 17(4):295–301, 1999.

[QSA16] Mingyue Qiu, Yu Song, and Fumio Akagi. Application of artificial neural network for the prediction of stock market returns: the case of the japanese stock market. *Chaos, solitons & fractals*, 85:1–7, 2016.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[Qui93] J Ross Quinlan. C4. 5: programming for machine learning. *Morgan kauffmann*, 1993.

[Ree93] Russell Reed. Pruning algorithms-a survey. *Neural networks, ieee transactions on*, 4(5):740–747, 1993.

[RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Tech. rep. DTIC Document, 1985.

[Ros58] Frank Rosenblatt. *Two theorems of statistical separability in the perceptron*. United States Department of Commerce, 1958.

[Sha94] William F Sharpe. The sharpe ratio. *The journal of portfolio management*, 21(1):49–58, 1994.

[SHH93] Venkat Subramanian, Ming S Hung, and Michael Y Hu. An experimental evaluation of neural networks for classification. *Computers & operations research*, 20(7):769–782, 1993.

[SHH96] Murali Shanker, Michael Y Hu, and Ming S Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, 1996.

[SMR+02] Marc Sebban, I Mokrousov, N Rastogi, and C Sola. A data-mining approach to spacer oligonucleotide typing of mycobacterium tuberculosis. *Bioinformatics*, 18(2):235–243, 2002.

[TD92] Robert R Trippi and Duane DeSieno. Trading equity index futures with a neural network. *The journal of portfolio management*, 19(1):27–33, 1992.

[Tra] TradeStation. Tradestation group, inc. `http://www.tradestation.com/`. Accessed: 2016-12.

[TW09] CF Tsai and SP Wang. Stock price forecasting by hybrid machine learning techniques. In *Proceedings of the international multiconference of engineers and computer scientists*. Vol. 1. (755), 2009, p. 60.

[TY07] Geoffrey KF Tso and Kelvin KW Yau. Predicting electricity energy consumption: a comparison of regression analysis, decision tree and neural networks. *Energy*, 32(9):1761–1768, 2007.

[VLV99] Alfredo Vellido, Paulo JG Lisboa, and J Vaughan. Neural networks in business: a survey of applications (1992–1998). *Expert systems with applications*, 17(1):51–70, 1999.

[Wan09] Yi-Hsien Wang. Nonlinear neural network forecasting model for stock index option price: hybrid gjr–garch approach. *Expert systems with applications*, 36(1):564–570, 2009.

[Wer90] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the ieee*, 78(10):1550–1560, 1990.

[Whi89] Halbert White. Learning in artificial neural networks: a statistical perspective. *Neural computation*, 1(4):425–464, 1989.

[WLL06] Muh-Cherng Wu, Sheng-Yu Lin, and Chia-Hsin Lin. An effective application of decision tree to stock trading. *Expert systems with applications*, 31(2):270–274, 2006.

[WLM+08] Gary R Weckman, Sriram Lakshminarayanan, Jon H Marvel, and Andy Snow. An integrated stock market forecasting model using neural networks. *International journal of business forecasting and marketing intelligence*, 1(1):30–49, 2008.

[ZH98] Gioqinang Zhang and Michael Y Hu. Neural network forecasting of the british pound/us dollar exchange rate. *Omega*, 26(4):495–506, 1998.

[Zha03] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[ZJ04] Zhi-Hua Zhou and Yuan Jiang. Nec4. 5: neural ensemble based c4. 5. *Knowledge and data engineering, ieee transactions on*, 16(6):770–773, 2004.

Abbreviations

ANN - Artificial Neural Network.

ANNs - Artificial Neural Networks.

DT - Decision tree.

DTs - Decision trees.

MAE - Mean Absolute Error.

MAD - Mean Absolute Deviation.

MAPE - Mean Absolute Percentage Error.

MSE - Mean Square Error.

OHLC - Open, High, Low ,Close. This is price information of a data bar. Bars' open price, bars' highest price, bars' lowest price and bars' close price.

OLS - Ordinary Least Squares.

RMSE - Root Mean Squared Error.

ROI - Return On Investment.

USD - United States dollar.

Appendix 1
ANN comparison to statistical-econometric models

Table A1. Most comparative studies show that neural networks outperform statistical-econometric models. This table is taken from [FL01]. References to studies listed in this table can be found in paper [FL01].

Study	Domain	Statistical model	Performance: Statisticalmodel vs. neuralnetworks (ANN)	Conclusion
Altman, Marco and Varetto, 1994	Corporate distress diagnosis	Linear discriminant analysis	88.4% vs. 87.8% diagnosis accuracy on training data. 94.7% vs. 93.6% diagnosis accuracy on testing data	Similar performance
Barr and Mani, 1994	Investment management	Linear regression	Similar on root-mean-square error. 38% vs. 116% total return on investment	ANNs outperform linear regression in trading profits
Berry and Trigueiros, 1993	Extraction of knowledge from accounting reports	Discriminant analysis 30 vs. 45 correct conclusions	ANNs perform better	
Chiang, Urban and Baldridge, 1996	Mutualfund net asset value forecasting	Regression	15.17% vs. 8.76% mean-absolute percent forecasting error	ANNs outperform both linear and nonlinear regressions
Dutta and Shekhar, 1988	Bond rating	Regression	67.7% vs. 92.4% rating accuracy on training data. 82.4% vs. 64.7% rating accuracy on testing data	ANNs outperform regression
Odom and Sharda, 1990	Bankruptcy prediction	Discriminant analysis	59.26% vs. 81.48% prediction accuracy	ANNs perform better
Rahimian et al., 1993	Bankruptcy prediction	Discriminant analysis 74.5% vs. 81.8% prediction accuracy	ANNs perform better	
Salchenberger, Cinar and Lash, 1992	Predicting thrift failures	Logit	92.3% vs. 95.8% prediction accuracy	ANNs are better
Tam and Kiang, 1992	Bank failure predictions	Discriminant analysis	11% vs. 3.8% misclassification rate on training data. 15.9% vs. 14.8% misclassification rate on testing data	ANNs offer better predictive accuracy
Wilson and Sharda, 1994	Bankruptcy prediction	Discriminant analysis	88.65% vs. 100% prediction accuracy on training data. 88.25% vs. 97.5% prediction accuracy on testing data	ANNs perform better
Yoon, Swales and Margavio, 1993	Predicting stock price performance	Multiple discriminant analysis (MDA)	74% vs. 91% prediction accuracy on training data. 65% vs. 77% prediction accuracy on testing data	ANNs outperform MDA

Appendix 2

Frequency usage of decision tree algorithms

Table A2. Most commonly used decision trees in percentage. This table is taken from [AS09].

Decision tree algorithm	Usage frequency in percent
C4.5	54.55
C5.0	9
CART	40.9
CLOUDS	4.5
CLS	9
IDE	68
IDE3+	4.5
OCI	4.5
PUBLIC	13.6
Random Forest	9
Random Tree	4.5
SLIQ	27.27
SPRINT	31.84

Appendix 3

Price variation example in first seconds after session opening

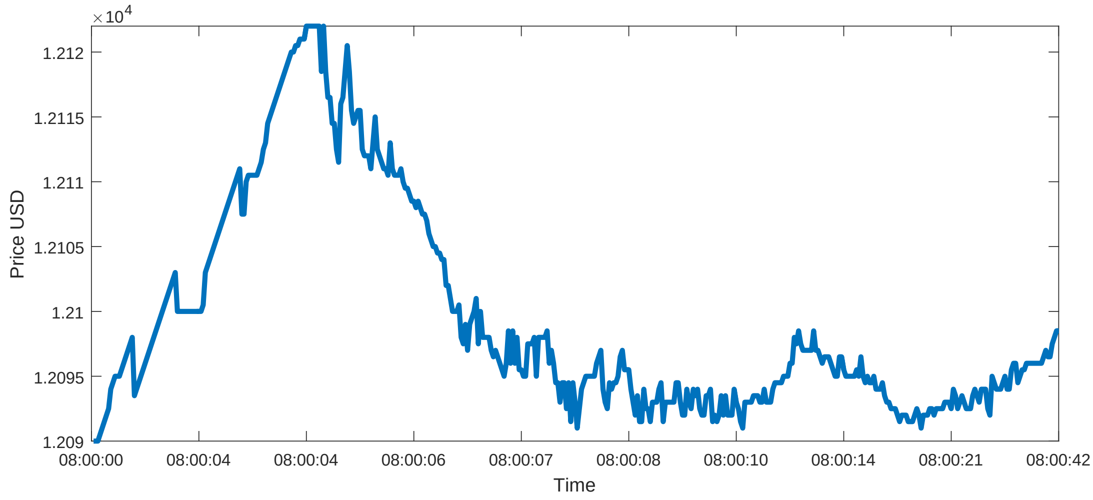

Figure A1. DAX price variates a lot in the first seconds of opened session.

Appendix 4
Changes of volatility for S&P 500

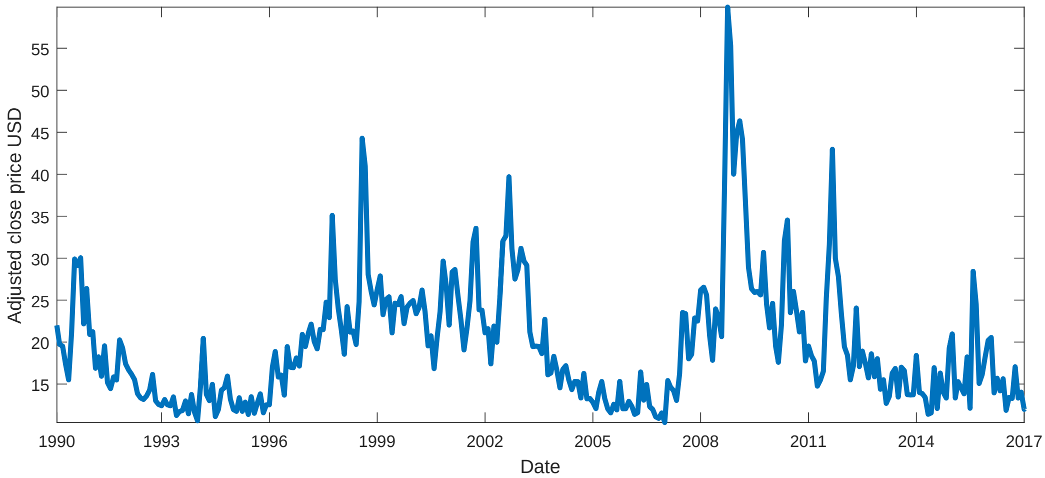

Figure A2. Changes of volatility for S&P 500 stocks index.

Appendix 5

Performance of funds which exploit systematic strategies based on technical analysis

Table A3. Altegris 40 index returns. This table is taken from [Alt].

Year	Return	Year	Return
1990	37.15 %	2004	2.57 %
1991	15.12 %	2005	4.51 %
1992	0.89 %	2006	6.70 %
1993	14.66 %	2007	7.18 %
1994	-5.46 %	2008	15.47 %
1995	13.16 %	2009	-7.98 %
1996	16.04 %	2010	11.33 %
1997	10.22 %	2011	-3.23 %
1998	12.61 %	2012	-4.75 %
1999	0.87 %	2013	-2.45 %
2000	10.63 %	2014	15.75 %
2001	5.39 %	2015	0.09 %
2002	15.22 %	2016	-3.13 %
2003	15.99 %		

Appendix 6
Portfolio asset allocation

To compare neural network and decision tree prediction results a portfolio is constructed using prediction models of 44 futures financial instruments for both neural network and decision tree. It is assumed that for every future financial instrument 25,000 US dollars are used as an investment asset. So in total there are $25{,}000 \times 45 = 1{,}100{,}000$ US dollars. If a practitioner wants to buy or sell short a futures financial instrument he/she only uses as much asset as the initial margin level is. So the trading happens with a leverage. The initial margin is the minimum amount required to make a futures deal. Initial margin is determined by a broker thru which the buy or sell short trades are made. Brokers determines initial margins based on the risk (volatility) of a futures financial instrument. Therefore, initial margin is a good measure to determine what amount of contracts should be bought so that potential winnings and losses would be equal for all futures financial instruments. For example, bonds futures usually are not volatile, whereas stock index futures are usually much more volatile. Therefore, initial margin of one bond future is usually much lower than the one of stock index future. Lower volatility means that the potential losses and winning are also lower. Hence, in order to have an equal risk/potential winning for all futures financial instruments, different size of contracts must be bought or sold short.

Initial margins which are used in this research are taken from *TradeStation* [Tra] broker.

Contract sizes of the portfolio are calculated like this: if initial margin of a future is 5,000 US dollars, then $\frac{25{,}000}{5} = 5$ future contracts are bought or sold short.

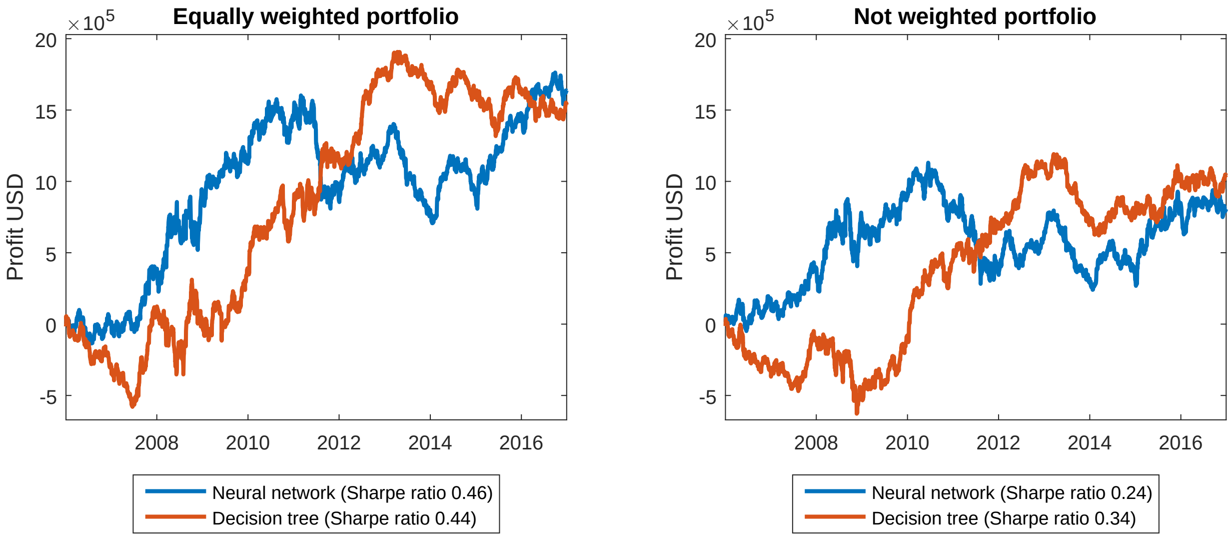

Figure A3. Significance of asset allocation to the portfolio results.

A significance of asset allocation in the portfolio is shown in figure A3. In this figure results of portfolios for testing data are shown. In the left graph portfolio is constructed using initial margin to determine the size of contracts. In the right graph contract sizes are the same for every of 44 futures financial instruments. In this case the size of contracts is such, that on average initial margin would be 25,000. Such contract sizes are used so that results of different portfolios could be compared. It is clear that results are affected a lot based on which asset allocation method is used. The goal of this paper is to compare different methods, hence equal weighted portfolio should be used. In such portfolio, every future financial instrument has the same impact on the results of the whole portfolio.

Appendix 7

COMB1 results comparison for validation data

Table A4. Validation data results for every financial instrument.

Future abbreviation	AD	BO	BP	C	CC	CD	CL	DX	EC	ED	EMD
ANN Sharpe ratio	0.4816	0.5342	0.9719	0.2732	-0.5098	-0.7644	0.4043	-0.4791	-0.1676	0.4947	1.3961
DT Sharpe ratio	0.7530	-0.0352	-0.9987	1.0830	-1.2591	-0.1016	0.7288	0.6440	0.6826	-0.2063	1.7238
COMB 1 Sharpe ratio	0.6593	0.0901	-0.1859	0.8567	-1.0977	-0.5987	0.7560	-0.0879	0.1702	0.0485	1.6356
ANN accurate prediction %	55.64	51.16	51.61	51.09	50.00	49.81	54.86	51.55	49.82	48.39	54.90
DT accurate prediction %	58.17	44.96	45.05	52.80	46.86	51.75	51.25	52.69	54.59	50.60	57.95
COMB1 accurate prediction %	57.71	47.26	46.76	52.28	48.03	51.17	54.15	52.49	51.89	50.60	57.09

Future abbreviation	ES	FDAX	FESX	FGBL	FGBM	FGBS	FSMI	FV	GC	HG	HO
ANN Sharpe ratio	-0.0394	-0.8147	-0.8566	2.7298	-0.2694	0.3269	0.6294	0.1008	0.5361	0.4847	-1.2693
DT Sharpe ratio	0.2533	0.5211	-0.5240	1.7132	0.9024	0.8202	-1.0223	0.5341	-0.7421	-0.4017	0.8837
COMB1 Sharpe ratio	-0.0593	-0.3505	-0.7705	2.3897	0.1945	0.0397	-0.0744	0.1614	0.1470	-0.1266	-0.3533
ANN accurate prediction %	54.59	46.12	45.64	59.79	51.98	50.24	53.30	48.60	50.75	50.76	46.43
DT accurate prediction %	55.00	47.69	49.14	57.79	52.82	51.89	46.19	52.38	42.29	49.12	51.79
COMB1 accurate prediction %	54.97	44.30	47.31	60.53	52.43	48.53	50.00	49.03	48.98	50.26	48.88

Future abbreviation	KC	LC	LH	MP1	NG	NQ	PA	PL	QM	RB	RR
ANN Sharpe ratio	-1.0695	0.3801	0.7974	0.1962	0.5088	0.9231	0.2399	-0.7656	0.4374	0.8530	1.6236
DT Sharpe ratio	0.3886	1.0551	0.6033	-0.1038	-0.0353	2.1826	0.0968	-0.9697	0.4149	0.7013	1.8758
COMB1 Sharpe ratio	-0.7386	0.8761	0.8870	0.0407	0.4733	1.9876	0.1944	-1.1426	-0.0536	0.9291	2.1107
ANN accurate prediction %	48.56	53.33	52.74	56.25	48.93	60.67	50.20	43.92	49.77	53.40	57.21
DT accurate prediction %	50.23	52.97	54.36	50.20	52.48	59.03	53.57	47.47	49.17	56.34	56.63
COMB1 accurate prediction %	47.87	54.40	53.98	53.85	51.50	61.17	50.90	42.33	47.20	56.47	57.66

Future abbreviation	S	SB	SF	SI	SM	TF	TU	TY	US	W	YM
ANN Sharpe ratio	0.0302	0.4376	-1.0684	1.5006	-0.5282	0.2027	0.1085	1.0172	0.7791	0.2432	0.6229
DT Sharpe ratio	-0.2226	-0.0656	-1.4285	0.7278	1.1424	0.6827	-0.5082	1.1872	1.6531	0.6911	0.5348
COMB1 Sharpe ratio	-0.0169	0.0799	-1.2411	1.3459	0.3560	0.6076	-0.5042	1.5255	1.2452	0.4838	0.4376
ANN accurate prediction %	52.38	52.48	46.44	54.55	47.45	56.86	47.10	54.08	54.10	52.09	54.21
DT accurate prediction %	50.00	47.47	45.92	51.09	53.85	55.87	44.94	52.33	55.26	51.77	52.82
COMB1 accurate prediction %	52.11	50.24	45.97	54.97	50.79	58.25	44.89	55.79	55.43	53.69	53.85

Appendix 8

COMB2 results comparison for testing data

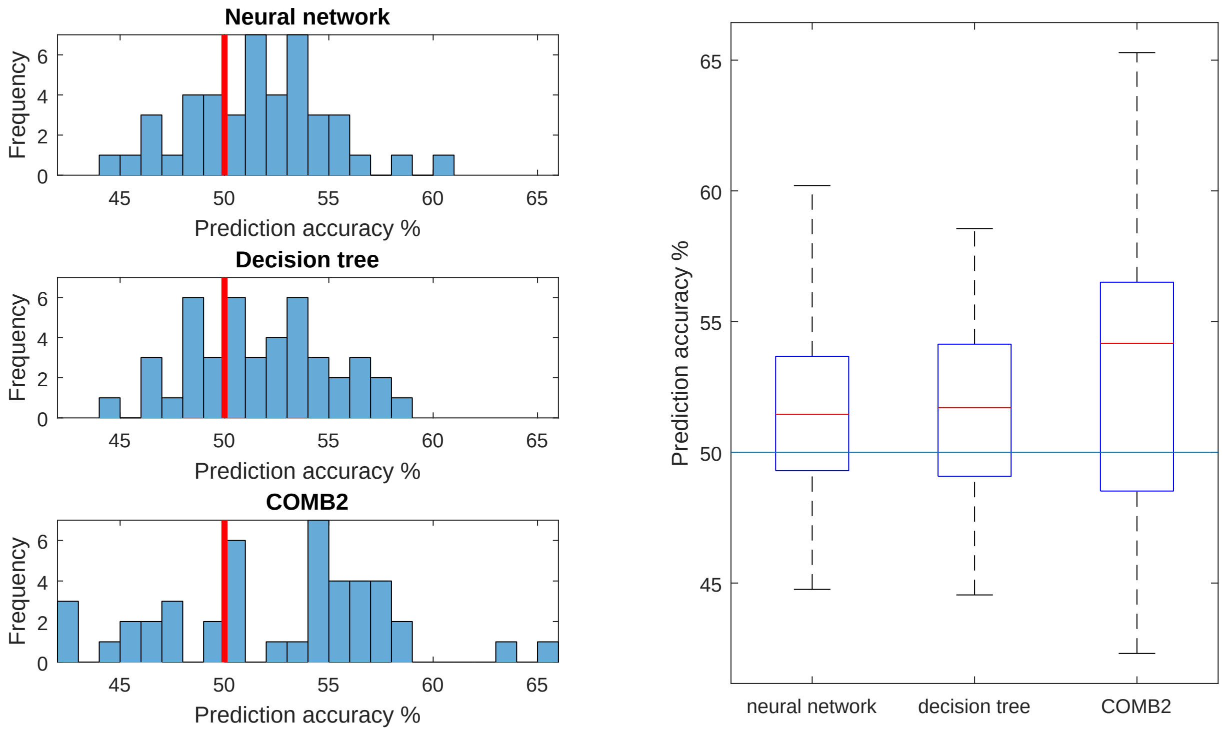

Figure A4. Prediction accuracy.

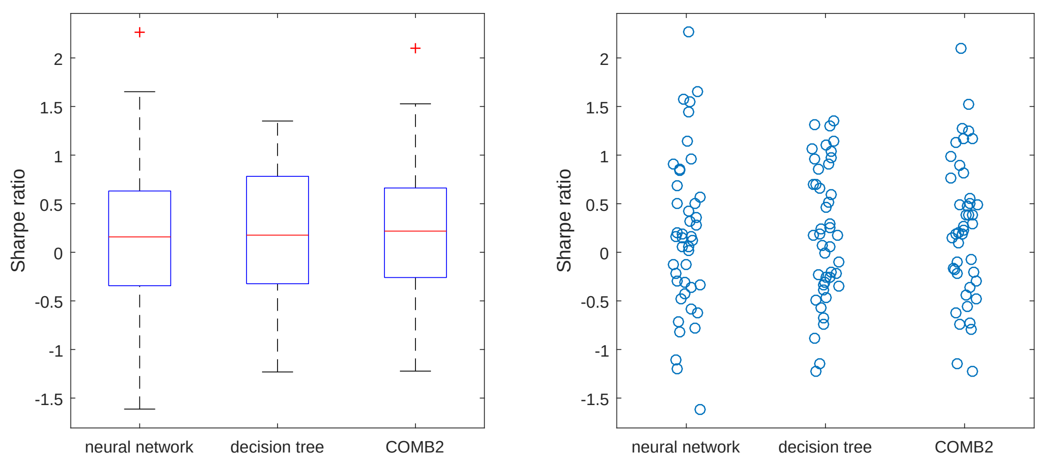

Figure A5. Sharpe ratio.