METHODOLOGY **Open Access**

Enhancing credit card fraud detection: highly imbalanced data case

Dalia Breskuvienė[1*] and Gintautas Dzemyda[1]

*Correspondence:
dalia.breskuviene@mif.vu.lt

[1] Institute of Data Science and Digital Technologies, Vilnius University, Akademijos 4, 08412 Vilnius, Lithuania

Abstract

In the contemporary landscape, fraud is a widespread challenge in today's financial landscape, requiring innovative methods and technologies to detect and prevent losses from the sophisticated tactics used by fraudsters. This paper emphasizes the main issues in fraud detection and suggests a novel feature selection method called FID-SOM (feature selection for imbalanced data using SOM). Feature selection can significantly improve classification performance. Given the inherent imbalance in fraud detection data, feature selection must be done with an enhanced focus. To accomplish this task, we use Self-Organizing maps, which are a special type of artificial neural network. FID-SOM is designed to address the challenge of dimensionality reduction in scenarios characterized by highly imbalanced data. It has been specifically designed to efficiently process and analyze vast and complex datasets commonly encountered in the financial sector, showcasing adaptability to the dynamic nature of big data environments. The uniqueness of the proposed method is in forming a new dataset containing the Best-Matching Units of the trained SOM as vectors of attributes corresponding to the initial features. These attributes are sorted based on variance in descending order. By keeping the required number of attributes that hold the highest percentage of variability, we select features corresponding to those attributes for further analysis. The proposed FID-SOM method has demonstrated its ability to perform on par with, if not surpass, existing methodologies. It also shows innovative potential.

Keywords: Feature selection, SOM, Imbalanced data, Classification, Fraud detection

Introduction

In the dynamic landscape of financial crime detection, the ever-evolving tactics employed by fraudsters necessitate continuous advancements in analytical methodologies. Early efforts to combat fraud primarily relied on rule-based systems and manual inspection of transactions. Rule-based systems involve formulating predefined rules and heuristics to flag potentially fraudulent activities. However, these approaches are limited in their ability to adapt to new and evolving fraud patterns. Additionally, manual inspection is time-consuming and impractical for handling the vast volumes of digital transactions in real-time. Consequently, there is a demand for more sophisticated and automated methods.

In order to tackle the ever-growing sophistication of fraudsters, financial institutions need to harness the potential of artificial intelligence (AI) and machine learning (ML) algorithms. By leveraging advanced AI capabilities, these institutions can proactively detect and prevent fraudulent activities, safeguarding their customers and reputation. Machine learning algorithms emerge in fraud prevention area [1, 2]. However, there are several challenges when applying ML to financial fraud data, with the most significant obstacle being data imbalance. For instance, most transactions are legitimate when working with transactional data, while less than 1% of transactions are fraudulent.

Moreover, credit card fraud detection using machine learning suffers from concept drift [3], high-dimensional categorical features [4], lack of public databases, and even some performance measures can be misleading when used for imbalanced data [5]. To address the challenges above, implementing effective feature selection methods becomes crucial. Feature selection plays a pivotal role in enhancing the performance and efficiency of fraud detection models by selecting the most relevant and informative features from the dataset.

This paper introduces a novel method, FID-SOM (feature selection for imbalanced data using SOM), for feature selection utilizing the capabilities of a self-organizing-map (SOM) to generalize data. FID-SOM is designed to address the challenge of dimensionality reduction in scenarios characterized by highly imbalanced data. We compare our method with five feature selection methods: univariate feature selection utilizing F-test, χ^2 test and mutual information, recursive feature elimination (RFE), and XGBoost feature importance on different datasets. Our method outperforms comparison methods in the majority of cases.

The contributions of this study can be summarized as:

- Proposed method FID-SOM addresses the challenge of dimensionality reduction in scenarios characterized by highly imbalanced data.
- This paper identifies existing gaps in feature selection for imbalanced data, urging researchers to delve into this domain.
- The proposed method integrates a Self-Organizing Map, which can handle noise in features and identify patterns in the imbalanced data to improve feature selection for classification tasks.
- The proposed method significantly improves the classifier performance by finding the proper features for particular imbalanced datasets.

The rest of the paper is organized as follows. Section "Related work" provides an overview of existing research in the field. Following this, Section "A novel method FID-SOM of feature selection for imbalanced data using SOM" outlines the approach taken in this study. Detailed information about the data used for experiments is presented in Sect. "Experimental results", highlighting the datasets employed to validate the proposed method. The "Results" subsection in Section "Experimental results" showcases the outcomes of the experiments that were conducted. Lastly, the paper concludes with the "Discussions" and "Conclusions and Future Research" sections, summarizing the essential findings and implications of the research.

Related work

The evolution of technology and the increasing complexities of digital transactions have given rise to sophisticated fraudulent activities, necessitating novel and intelligent solutions for detecting and preventing such cyber threats. Here, we provide an overview of the feature selection methods when working with imbalanced datasets, especially in fraud detection applications.

Feature selection techniques are commonly categorized into three main groups: filters, wrappers, and embedded methods [6, 7]. The review paper [8] delves into the significance of feature selection in machine learning and data mining. It highlights contemporary challenges that are of particular importance. These challenges include feature selection for high-dimensional data with small sample sizes, dealing with large-scale data, and ensuring secure feature selection. Despite these challenges, several noteworthy trends in feature selection have surfaced, such as stable feature selection, multi-view feature selection, distributed feature selection, multi-label feature selection, online feature selection, and adversarial feature selection. The paper goes on to explore recent advancements in these areas. For each trend, it examines the current issues, presents existing solutions, and discusses them. Beyond these trends, the paper also introduces diverse applications of feature selection. These applications span fields including bioinformatics, social media analysis, and multimedia retrieval, showcasing practical relevance.

An alternative approach to arranging feature selection methods involves distinguishing between global and instance-wise feature selection strategies. The primary objective of global feature selection is to identify a singular feature selector applicable to all data samples, focusing on minimizing the number of features while retaining the capacity for discriminative predictions. On the other hand, instance-wise feature selection involves calculating distinct selectors for each instance, resulting in enhanced performance compared to the global feature selection approach. In ref. [9] suggests group-wise feature selection, which occupies an intermediate position between global feature selection and instance-wise feature selection.

The paper [10] highlights the importance of feature selection in reducing data processing complexity, particularly in the context of high-dimensional data. The study introduces the concept of fuzzy combination entropy (FCE) to address the limitations of classical combination entropy, especially in handling continuous features. The paper presents the development of FCE based on fuzzy λ-similarity relation, incorporating fuzzy rough sets and combination entropy. Furthermore, the concepts of global and local feature correlations are defined, leading to the design of a feature selection method, FSm-FCE. Experimental findings demonstrate the algorithm's ability to preferentially select a smaller feature set while maintaining commendable classification performance.

Feature selection for imbalanced data

When working with imbalanced datasets, where one class is significantly more prevalent than the other, feature selection becomes an even more complex task. Imbalanced datasets can introduce biases and negatively affect the performance of machine learning models.

The work by ref. [11] suggests a feature selection technique that centers around class decomposition. The suggested approach initially subdivides majority classes into more manageable pseudo-subclasses characterized by relatively balanced sizes. Subsequent feature selection operates on these newly decomposed data to calculate feature goodness metrics. Moreover, the study introduces a feature selection method reliant on the Hellinger distance. It measures distribution divergence, offering greater resilience to imbalanced class distributions [11].

Researchers propose many different approaches for feature selection when working with imbalanced data. For instance, neighborhood rough set theory is employed for feature selection [12]. The empirical findings showed the effectiveness of RSFSAID (Rough-Set-based Feature Selection Algorithm for Imbalanced Data) across binary and multiclass datasets. Nevertheless, in most scenarios, the information about the minority class holds greater significance. The noise within the minority class might impact the classifier's generalization ability when utilizing the chosen features.

The paper [13] introduces a feature selection technique for imbalanced data, utilizing a new regularization method called IR-LDA to enhance classification performance by emphasizing the minority class. The method employs cosine similarity to address feature redundancy issues and incorporates the regularization into the global feature selection framework, improving classifier performance and reducing feature redundancy.

Application of the self-organizing-map related with imbalanced data

SOM, being an unsupervised neural network, offers a promising solution by enabling the visualization and clustering of high-dimensional data while preserving its intrinsic structure. In cases of imbalanced data, SOM can benefit in identifying and understanding the distribution of minority classes, potentially uncovering hidden patterns and relationships. It is used in many applications like Cyber Intrusion and Anomaly [14–16] detection, investigation of energy demand [17], or even for text-independent speaker identification [18].

SOM is frequently employed for unsupervised dataset clustering [19–21]. Sometimes, it is used to cluster data into similar subsets and apply feature selection on each cluster [22]. By incorporating SOM as a preprocessing step or integrating them into ensemble methods, researchers and practitioners can enhance the robustness and accuracy of their models when dealing with imbalanced data, contributing to improved decision-making.

A novel method FID-SOM of feature selection for imbalanced data using SOM

In this section a novel method FID-SOM (feature selection for imbalanced data using SOM) for feature selection utilizing the capabilities of a self-organizing-map (SOM) to generalize data has been presented.

Consider a multidimensional dataset represented as an array X containing n data points, where each data point X_i $(i = 1, \ldots, n)$ is a vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ in $\mathbb{R}^m$. These data points are observations of objects or phenomena influenced by m different features $(x_1, x_2, \ldots, x_m)$. Some of these features are numerical, while others are categorical. Furthermore, each data point is associated with a class label y_i, where y_i indicates the category to which the sample X_i belongs.

In our specific context, these features describe various aspects of customers' financial behavior. We have categorized these data points into two classes where 0 represents Regular or Legitimate transactions, and 1 signifies Fraudulent transactions. Therefore, the target variable y assumes values $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$.

The Self-Organizing Map [23], often called SOM or Kohonen map, is a powerful unsupervised machine learning technique that falls under the category of artificial neural networks. Developed by Finnish professor Teuvo Kohonen in the 1980s, SOMs are used for dimensionality reduction, data visualization, clustering, and pattern recognition tasks. Despite the method being created at the end of the 20th century, it is still widely used for many actual applications. E.g. its combination with multidimensional scaling [24, 25] enlarged its possibilities to understand patterns in data. The fundamental concept behind SOM is to map high-dimensional input data onto a lower-dimensional grid while preserving the topological relationships between data points.

In our case, a SOM consists of a two-dimensional grid of nodes arranged in a rectangular pattern. Each node, also known as a neuron, is associated with a weight vector of the same dimension as the input data.

The dimensions of the map are evaluated by calculating the quantity of neurons based on the number of observations present in the training data, employing a formula [26]:

$$M \cong 5\sqrt{n}, \tag{1}$$

where M represents the number of neurons, approximating an integer value near the outcome derived from the right side of the equation, while n stands for the number of observations in training set of SOM. The number of rows and columns of SOM is $\cong \sqrt{M}$
.

The network learning is introduced briefly below. The weight vectors are initially assigned random values. In each step, an input point $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ is selected from the training data, and its Euclidean distance is computed with each of the neuron's weight vector (Eq. 2).

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - w_{jk})^2}, \tag{2}$$

where: d_{ij} is a distance between point X_i and weight vector $W_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$ of the j-th neuron.

The neuron that exhibits the smallest distance for a given input data point is identified as the best matching unit (BMU) for that data point. After identifying the best matching unit, the training process involves selecting the neighboring neurons of the BMU. These neighboring neurons are determined by a specific criterion, often based on their spatial proximity to the BMU within the neural network. Once the neighbors are established, the weight vectors associated with these neighboring neurons are updated using a neighborhood function.

Classical manner to update weights of neuron is as follows:

$$w_{jk}(t+1) = w_{jk}(t) + \eta(t) T_{j^*j}(t)(x_{ik} - w_{jk}(t)), \tag{3}$$

where t is a number of iteration,

$w_{jk}(t)$ is the k-th component of the weight vector of the j-th neuron at iteration t.

$\eta(t) = \eta_0 \exp\left(-\frac{t}{\lambda_\eta}\right)$ is the learning rate at the iteration t. It decreases over time to gradually reduce the influence of new input data on the weights.

$T_{j^*j}(t) = \exp\left(-\frac{\| W_{j^*} - W_j \|^2}{2\sigma(t)^2}\right)$ is the neighborhood function value between j^*-th (the best matching unit (BMU)) and the j-th neuron at iteration t.

$\| W_{j^*} - W_j \|$ is the lateral distance between neurons j^* and j, where W_{j^*} is a BMU (winning neuron).

x_{ik} is the k-th component of the input point X_i.

$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda_\sigma}\right)$ is neighborhood size.

Hyperparameters for SOM training are such:

η_0 is learning rate,

λ_η is a constant that determines the rate of decay,

σ_0 is neighborhood size,

λ_σ is a constant that determines the rate of decay for the neighborhood width.

The neighborhood function defines how much influence each neighbor should have on the BMU and its surrounding neurons. Typically, the influence decreases with distance from the BMU, effectively creating a decaying effect on the updates. In essence, the process of identifying the BMU and updating the weights of its neighbors through the neighborhood function forms the basis of a self-organizing map algorithm.

The example below could be employed to explain the SOM. Let us say we have a two-dimensional dataset which is visualized on the left side of Fig. 1. In the middle, we have SOM visualized on the grid/coordinate space. The graph on the right shows data points (grey dots) and neurons in weight space. The red dots are BMU, and the grey cross are neurons that never became BMU.

After collecting the weight vectors corresponding to the BMU, we obtain a data frame with dimensions of $n_{BMU} \times m$, where n_{BMU} represents the number of BMUs and m signifies the number of features or the length of each weight vector. Notably,

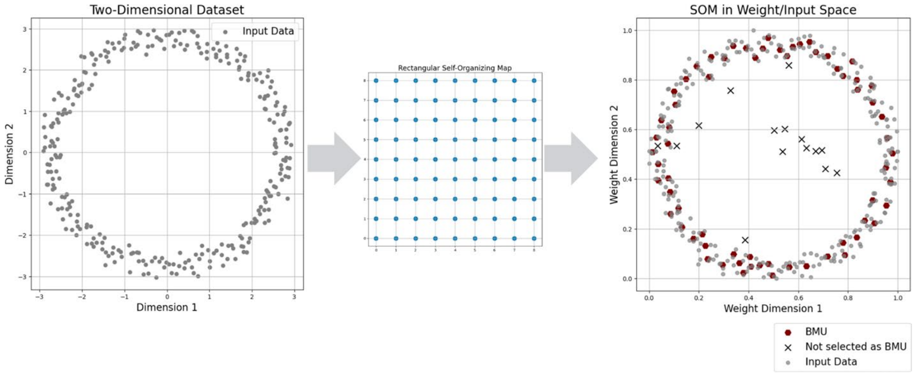

Fig. 1 Example of SOM

n_{BMU} remains equal to or less than the total number of neurons, denoted as M since not every individual neuron is selected for the role of a BMU.

Subsequently, this data frame serves as a foundation for the feature selection process, a pivotal step in refining the most relevant features from the original dataset, i.e., we try to decrease the number of features m significantly.

We propose to select a subset of features based on SOM weight variation. By normalizing the BMU data and calculating the variance of each attribute, we determine the importance of each feature in capturing the data's variability. The attributes are arranged in descending order according to their variance. This results in a list of features sorted by their significance. Subsequently, we can choose the desired number of features from the top of this ordered list.

Self-organizing-map is used for clustering tasks [27]. However, we employ the SOM's generalization capabilities to solve the dimensionality reduction problems for sharply imbalanced datasets. These ideas make the core of a novel method, FID-SOM, for feature selection for imbalanced data using SOM. The algorithm of the proposed method is presented by pseudo-code in Algorithm 1

Algorithm 1 FID-SOM (feature selection for imbalanced data using SOM)

```
Require: $X$: Dataset
Require: $params$: SOM parameters
Require: $d$: Desired number of features.
Ensure: features subset ensuring high classifier performance
procedure SELECTFEATURES
    train SOM using parameters $params$ with dataset $X$
    form a new dataset $W_{BMU}$ containing $n_{BMU}$ weight vectors of $m$ attributes
  corresponding to $m$ features of dataset $X$
    normalize $W_{BMU}$ dataset attributes to a scale of [0,1]
    calculate the variance of each attribute
    sort attributes based on variance in descending order
    select $d$ attributes from the top of the list
    select features for dataset $X$ corresponding to the kept attributes
end procedure
```

This method enables efficient feature selection for downstream analysis or visualization. This method dynamically adapts to the inherent characteristics of the data, ensuring an automatic and data-driven feature selection process. This attribute significantly enhances the method's suitability for diverse scientific applications, where datasets often vary in dimensionality and complexity.

The weight vector is critical for mapping high-dimensional transactional data into a lower-dimensional space, preserving the topological relationships of the input data. In the context of fraud detection, this enables the SOM to cluster similar transactions together while highlighting outliers, which often correspond to fraudulent behavior. The SOM provides a structured framework for analyzing complex transactional patterns and identifying anomalous activities by associating each node with a weight vector.

Classifiers and metrics for FID-SOM evaluation

The performance of FID-SOM was compared with five feature selection methods: univariate feature selection [28] utilizing the F-test, χ^2 test and mutual information, recursive feature elimination [29], and the XGB Importance method [30]. The baseline for performance evaluation is the model performance without a special selection of features, i.e., using all features for modeling.

The purpose of feature selection is to increase the performance of machine learning algorithms. The efficacy of the feature selection methods was evaluated using the XGBoost [31], CatBoost [32], and Random Forest [33] machine learning algorithms. The main reason for choosing Random Forest is its good performance on data related to financial fraud detection [34, 35]. Meanwhile, XGBoost and CatBoost usage is gaining popularity and demonstrating strong performance [36, 37]. We are aware that Logistic Regression is often employed to solve fraud detection tasks. However, our decision was not to use this algorithm for further experiments as it showed weak performance in scenarios where hyperparameters were not being optimized, or data was not balanced [38]. We did not use parameter hypertuning or data sampling in order to find the pure effect of feature selection methods.

To evaluate the goodness of the method, we use five metrics suitable for imbalanced datasets, namely F1 score, MCC, G-Mean, AUC-PR, and AUC-ROC.

The F1 score can be defined as the harmonic mean of precision and recall, effectively encapsulating precision and recall within a single metric in a symmetrical manner.

$$\mathrm{F1} = 2 \times \frac{precision \times recall}{precision + recall}, \tag{4}$$

where

$$\begin{aligned} Precision &= \frac{TP}{TP + FP}, \\ Recall &= \frac{TP}{TP + FN}. \end{aligned}$$

TP is True Positives (correctly predicted positive instances), *TN* is True Negatives (correctly predicted negative instances), *FP* is False Positives (incorrectly predicted positive instances), *FN* is False Negatives (incorrectly predicted negative instances).

The Matthews Correlation Coefficient (MCC) is a measure commonly used to assess the quality of binary classification models, especially when dealing with imbalanced datasets. It takes into account true positives, true negatives, false positives, and false negatives and provides a value that ranges from − 1 to + 1, with + 1 indicating a perfect prediction, 0 indicating random prediction, and − 1 indicating complete disagreement between prediction and observation.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{5}$$

The G-Mean, also known as the geometric mean or balanced accuracy, is a statistical measure used to evaluate the performance of classification models, particularly in situations where class imbalance exists. It offers a balanced perspective by considering sensitivity (recall) and specificity. It is defined as:

$$G\text{-}Mean = \sqrt{\text{sensitivity} \times \text{specificity}}, \tag{6}$$

where Sensitivity (True Positive Rate) = $\frac{TP}{TP+FN}$ and Specificity (True Negative Rate) = $\frac{TN}{TN+FP}$.

AUC-PR is a performance metric used to evaluate the effectiveness of classification models, especially in scenarios where class imbalance exists or when the focus is on positive instances. The Precision-Recall curve plots precision against recall as the classification threshold changes. Precision represents the proportion of correctly predicted positive instances among all instances predicted as positive, while recall is the proportion of correctly predicted positive instances among all actual positive instances.

AUC-ROC is another widely used performance metric for binary classification models. The ROC curve plots the true positive rate (recall) against the false positive rate as the classification threshold changes. The true positive rate is the proportion of correctly predicted positive instances among all actual positive instances, and the false positive rate is the proportion of incorrectly predicted positive instances among all actual negative instances.

Experimental results

In this section, we present the results obtained from our experimental study. Our experiments were designed to test the proposed method FID-SOM described in Section 3. We provide a detailed description of used datasets and decisions made in data preparation and data splitting, supported by quantitative and qualitative assessments, along with visual aids such as tables and figures.

Data used for experiments

For our experimental analysis, we employed three datasets. Among these, two datasets were derived from synthetic transactional payments data, while the third dataset represents a read transactional dataset.

Numerous real-world legal regulations govern the usage of private data, including prominent ones like the GDPR (General Data Protection Regulation), CCPA (California Consumer Privacy Act), and the "Act On Payment Services And Electronic Money." In this context, synthetic data emerges as a promising technological solution to address concerns related to privacy, fairness, data augmentation, and various other relevant issues.

Payment fraud represents a domain characterized by restricted access to data. Synthetic datasets like [39, 40] help overcome the abovementioned issues. Dataset [39] enables researchers and developers to study the purchasing habits of U.S. citizens within a virtual world featuring customers, merchants, and fraudsters. The data was meticulously designed to maintain key statistics, such as mean and standard deviation, resembling those of the actual population. Dataset developers employed stochastic sampling, generally from a Gaussian distribution, to select individual characteristic values. Unlike other synthetic datasets, this unique dataset ensures the interconnectedness of individuals' activities. For instance, spending behavior varies when individuals are in travel mode or during weekdays and weekends. The dataset also encompasses actual banking events, including the creation of chip-enabled cards, which were widely introduced in the U.S. in 2014, making "card-present" fraud more challenging for fraudsters.

Although the dataset includes transactions dating back to 1991, we exclusively utilize data from 2018 and 2019 for modeling purposes. The reason is that older transactions lack relevance in identifying contemporary fraud patterns.

The second data source employed in the experiments is also synthetic and was generated using the Sparkov Data Generation tool [40]. This dataset was published by its authors in two files—train and test. We merged the files so that we could do a proper split based on the timeline.

The third dataset contains credit card transactions conducted by European cardholders in September 2013. It encapsulates two-day transactions, revealing 492 instances of fraud out of 284 807 transactions. Notably, the dataset exhibits a substantial imbalance, with the positive class (frauds) constituting a mere 0.172% of all transactions. The dataset exclusively comprises numerical input variables resulting from a Principal Component Analysis (PCA) transformation. Regrettably, disclosure of the original features and additional contextual information is hidden due to confidentiality constraints. Principal components V1 through V28 are derived from PCA, while 'Time' and "Amount" are the only features unaffected by the transformation. "Time" denotes the seconds elapsed between each transaction and the initial transaction in the dataset, while 'Amount' represents the transaction amount. The "Time" feature is used for splitting purposes.

For the rest of the paper, Synthetic Credit Card data will be called DataSet-A, Sparkov-generated dataset will be called DataSet-B, Real Credit Card data will be called DataSet-C.

Table 1 represents the distribution between fraudulent and legitimate instances in each dataset. For the experiment, we used sharply imbalanced datasets. In size, those datasets are very different. In DataSet-A, which contains 3,445,553 cases and 25 attributes, the class distribution is 99.86% "non-fraud" and 0.14% "fraud". In DataSet-B, which contains 1,852,394 cases and 11 attributes, the level of fraud is slightly higher at 0.52%. DataSet-C, while maintaining a high majority of 99.83% "non-fraud," differs with a fraud rate of 0.17%, covering 284,807 cases and 30 attributes. These statistics reveal the imbalances and differences in attributes between each dataset, providing valuable insights for designing and evaluating robust fraud detection models.

Table 1 Description of the datasets

Category	DataSet-A	DataSet-B	DataSet-C
Not fraud (Percentage)	99.86%	99.48%	99.83%
Fraud (Percentage)	0.14%	0.52%	0.17%
# of instances	3,445,553	1,852,394	284,807
# of features	25	11	29

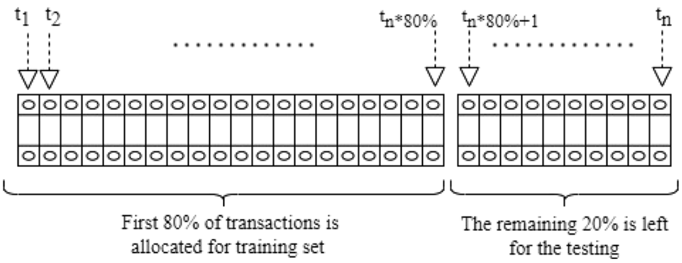
t1
t2
tn*80%
tn*80%+1
tn
First 80% of transactions is allocated for training set
The remaining 20% is left for the testing

Fig. 2 Dataset split based on time

Data preprocessing

In our comparison of feature selection methods, it is essential to train machine learning models. To facilitate this, we employ both training and testing datasets. Subsequently, we outline an appropriate approach for splitting the data into training and testing sets specifically tailored for fraud detection tasks.

Fraudulent data is inherently temporal, meaning that observations are dependent on previous observations in a sequential manner. This temporal dependence leads to correlations between data points that are close in time. The model might risk introducing temporal leaks if a standard train-test split is used on time series data, where data is randomly shuffled and partitioned. This can result in unrealistic correlations between the training and testing sets, leading to overly optimistic estimates of the model's performance [41].

Credit card, investment, or any other type of fraud data - has a concept drift property [42–44]. Concept drift refers to the phenomenon where the underlying statistical properties of the data distribution change over time. This can happen due to various reasons, such as changes in user behavior, fraud patterns, or market conditions.

The classical train-test split assumption of independent and identically distributed samples does not hold well for time series data, especially when dealing with concept drift in domains like fraud detection [45, 46]. When concept drift occurs, the assumption that the training and testing data are drawn from the same distribution is violated. To address this issue and create a more realistic evaluation setup for fraud detection, we suggest using *TimeSeriesSplit*. Instead of random shuffling, we suggest splitting the data chronologically, where the training data X_{train} comes from earlier periods, and the testing data X_{test} comes from later periods. This simulates the real-world scenario where the model is trained on historical data and tested on more recent data. FID-SOM and other feature selection methods used X_{train} to define the proper set of features.

Each dataset is split using a time-based approach. The dataset is divided so that the earliest 80% of instances would be for training, and the rest of the data, which has timestamps later than the training set, is left for testing (see Fig. 2). So, we are not setting up the split date, but instead, we are dynamically determining the split based on the chronological order of the data entries. This time-based approach ensures that the model is trained on historical data and then evaluated on more recent data, simulating a real-world scenario where the model makes predictions on new, unseen observations.

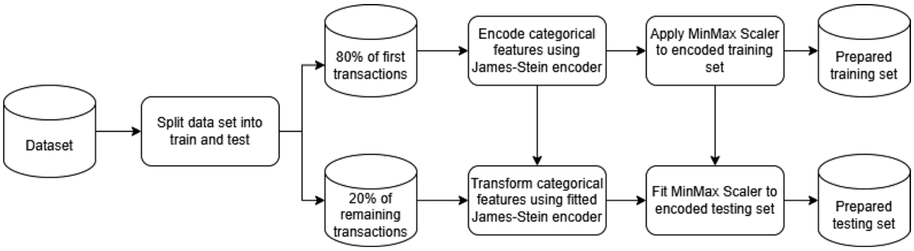
Dataset
Split data set into train and test
80% of first transactions
20% of remaining transactions
Encode categorical features using James-Stein encoder
Transform categorical features using fitted James-Stein encoder
Apply MinMax Scaler to encoded training set
Fit MinMax Scaler to encoded testing set
Prepared training set
Prepared testing set

Fig. 3 Data preprocessing steps which include data splitting, encoding, and normalization

We have used the categorical feature encoding method, James-Stein encoder, discovered as comparatively best for imbalanced data in the paper [38], where six feature encoders were compared.

Encoded data are scaled before training a Self-Organizing Map. This is essential because it ensures that all features contribute equally to the training process, regardless of their original units or magnitude. SOMs rely on the calculation of distances to map high-dimensional data onto a lower-dimensional grid. If features have vastly different ranges or units, for example, *"Transaction amount"* and *"Age"*, those with larger magnitudes can dominate the distance calculation, effectively overshadowing features with smaller scales. This imbalance can lead to a biased SOM, where the map primarily reflects variations in high-magnitude features, neglecting meaningful patterns in others. By scaling the data, usually through standardization (z-score scaling) or normalization (min-max scaling), we ensure that all features are on a comparable scale, allowing the SOM to identify and represent the intrinsic structure of the data more accurately. In our case, we use normalization which brings all features into the same range [0, 1], ensuring equal importance during training (see Fig. 3).

Results

The comprehensive overview of specifications of the SOM properties is presented in the Table 2. For DataSet-A, the SOM is characterized by a 90 × 90 grid structure, with an extensive training process involving 4,048,216 iterations. DataSet-B's SOM, slightly smaller with a 78 × 78 grid, underwent 3,043,349 iterations during training. Meanwhile, DataSet-C features a more compact 49 × 49 grid, with 1,193,330 iterations.

The upper part of Fig. 4 presents SOM in the grid space, where the dots represent BMUs and the size of the dots represents how many instances each BMU has. The lower part of Fig. 4 shows the dependency of a number of instances covered by a minimal number of BMU. The dashed horizontal line marks 95% of instances, and the dashed vertical line shows how many BMUs are required to cover these 95% of instances.

Table 2 SOM properties

Property	DataSet-A	DataSet-B	DataSet-C
Size	90 × 90	78 × 78	49 × 49
Iterations	4,048,216	3,043,349	1,193,330

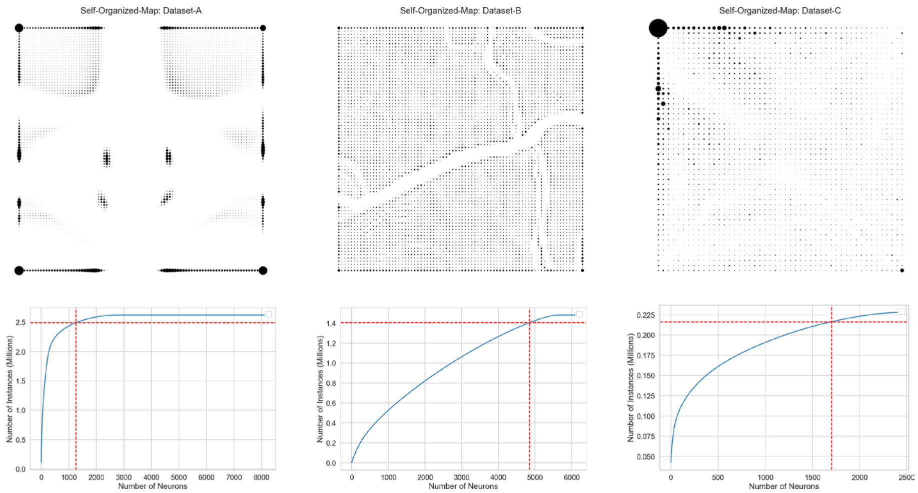

Fig. 4 Visualisation of the trained self-organized map for each dataset. The curves show the dependency of the number of instances covered by the number of Best-Matching-Units. The dashed horizontal line marks 95% of instances, and the dashed vertical line shows how many BMUs are required to cover these 95% of instances

Table 3 An example of how to identify the winning (best performing) feature selection methods

Method	F1	ROC	PR	MCC	G_MEAN
Baseline	0.82	**1.00**	0.95	0.83	0.85
FID-SOM	**0.95**	**1.00**	**0.98**	**0.95**	**1.00**
Uni_Chi2	0.82	**1.00**	0.94	0.83	0.85
Uni_F	0.83	**1.00**	0.95	0.84	0.85
Uni_MI	0.79	**1.00**	0.95	0.80	0.81
RFE	0.82	**1.00**	0.94	0.83	0.85
XGB_Imp	0.82	**1.00**	0.94	0.82	0.85

We observe that the SOM of each dataset is quite different. DataSet-A has many clusters, while DataSet-B is separated into two parts. The BMUs of SOM of DataSet-C have an almost uniform distribution with one very massive neuron.

The methods were evaluated 165 times, as there were three datasets with five metrics and three machine learning algorithms with different numbers of features selected. Each time, one or several feature selection methods were marked as the winning methods if they had the highest score of a particular metric.

An example of how to identify the winning (best performing) feature selection methods is shown in Table 3. The highest values in the result tables (Tables 3–8) for each metric are highlighted in bold.

The data in a Table 3 is a snapshot of our experiments. Here, the evaluation is performed by selecting the winning method for DataSet-A using the XGB classifier with 20 selected features for F1, MCC, and Geometric Mean. We mark that our proposed method, FID-SOM, became the best five times, and other methods became the best one time. The complete set of results is presented in Tables 5, 6 and 7.

Table 4 Comparison of feature selection methods

Method	# of winnings	Total	Percentage (%)
Baseline	33	165	20
FID-SOM	**73**	**165**	**44.24**
Uni_Chi2	33	165	20.00
Uni_F	18	165	10.91
Uni_MI	44	165	26.67
RFE	16	165	9.7
XGB_Imp	19	165	11.52

Table 4 presents a comparative analysis of various feature selection methods. The "Winning" column shows how often the particular method was selected as the best-performing method.

We calculated the average performance results of each method for five different random seed values to get a robust evaluation. In this case, results of FID-SOM are still outstanding (Table 4).

The proposed method FID-SOM is distinctive in its efficiency due to the utilization of a novel feature selection technique introduced in this study. It achieved a success rate of 44.24%, the highest among all methods considered. It almost outperforms the second-best method, Univariate_MI, which achieved a success rate of 26.67%, almost twice.

Different methods can have different optimal number of features. Considering this, we selected the best result for each metric/model/method from all compared feature sets. Results are shown in the Tables 5, 6 and 7. Each result represents the mean of five experiments conducted with different random seeds, along with the number of features that yielded the highest performance and the standard deviation across these five experiments.

In DataSet-A, using the CatBoostClassifier, the FID-SOM approach stands out with an impressive F1 score of 0.591, compared to the baseline's 0.40. Similarly, in DataSet-B with the RandomForestClassifier, FID-SOM excels with an F1 score of 0.835, outperforming the baseline's 0.622. Moving to DataSet-C, utilizing the XGBClassifier, FID-SOM maintains its robust performance with a notable F1 score of 0.801, surpassing the baseline's 0.796. Across all three datasets, FID-SOM consistently achieves superior results in various metrics, including ROC, PR, MCC, and G-MEAN, demonstrating its effectiveness as a classification method. These findings underscore the potential of FID-SOM in enhancing predictive capabilities and model performance across diverse datasets.

In these detailed tables, our proposed method, FID-SOM, has been marked 32 times as giving the best values for the selected metric.

The experimental results demonstrate the effectiveness of our proposed method. Notably, our proposed method works significantly better on the dataset structures when SOM can identify many homogeneous clusters and fewer neurons cover more data points. To get better results, one can vary the SOM architecture based on the dataset. In this study, the goal was to set up the same experimental environment rather than aiming for the highest performance metric for each dataset.

Table 5 Feature selection methods comparison with different machine learning models on DataSet-A

	F1-Score			**ROC-AUC**			**PR-AUC**			**MCC**			**G-MEAN**		
	Mean	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**
CatBoostClassifier															
Baseline	0.400	–	–	0.999	–	–	0.490	–	–	0.409	–	–	0.575	–	–
FID-SOM	**0.591**	**24**	**0.021**	**1.000**	**24**	**0.000**	**0.657**	**24**	**0.020**	**0.591**	**24**	**0.020**	**0.756**	**24**	**0.020**
RFE	0.403	24	0.030	0.999	22	0.007	0.489	24	0.070	0.412	24	0.019	0.575	24	0.042
Uni_Chi2	0.415	20	0.015	0.999	20	0.001	0.502	20	0.019	0.424	20	0.015	0.585	20	0.011
Uni_F	0.413	20	0.015	0.999	20	0.001	0.509	20	0.019	0.423	20	0.015	0.582	20	0.011
Uni_MI	0.403	24	0.016	0.999	24	0.000	0.489	24	0.025	0.412	24	0.017	0.575	24	0.015
XGB_Imp	0.415	22	0.019	0.999	24	0.000	0.492	22	0.017	0.424	22	0.018	0.586	22	0.019
RandomForestClassifier															
Baseline	0.844	–	–	**1.000**	–	–	0.967	–	–	0.851	–	–	0.864	–	–
FID-SOM	**0.957**	**24**	**0.003**	**1.000**	**24**	**0.000**	**0.980**	**24**	**0.006**	**0.958**	**24**	**0.003**	**0.998**	**24**	**0.001**
RFE	0.852	22	0.010	**1.000**	**22**	**0.000**	0.970	22	0.004	0.858	22	0.009	0.872	22	0.010
Uni_Chi2	0.851	22	0.019	**1.000**	**24**	**0.000**	0.973	24	0.004	0.857	22	0.017	0.873	22	0.018
Uni_F	0.851	22	0.013	**1.000**	**24**	**0.000**	0.973	24	0.004	0.857	22	0.012	0.873	22	0.012
Uni_MI	0.848	24	0.013	**1.000**	**24**	**0.000**	0.969	24	0.004	0.855	24	0.011	0.870	24	0.014
XGB_Imp	0.853	24	0.019	**1.000**	**24**	**0.000**	0.972	24	0.004	0.859	24	0.017	0.873	24	0.017
XGBClassifier															
Baseline	0.820	–	–	**1.000**	–	–	0.949	–	–	0.827	–	–	0.849	–	–
FID-SOM	**0.962**	**24**	**0.000**	**1.000**	**22**	**0.000**	**0.987**	**22**	**0.000**	**0.962**	**24**	**0.000**	**1.000**	**24**	**0.000**
RFE	0.825	22	0.000	**1.000**	**24**	**0.000**	0.949	24	0.000	0.832	22	0.000	0.853	22	0.000
Uni_Chi2	0.824	20	0.000	**1.000**	**24**	**0.000**	0.948	24	0.000	0.831	20	0.000	0.853	20	0.000
Uni_F	0.828	20	0.000	**1.000**	**20**	**0.000**	0.948	24	0.000	0.836	20	0.000	0.853	20	0.000
Uni_MI	0.820	24	0.007	**1.000**	**20**	**0.000**	0.949	24	0.002	0.827	24	0.006	0.849	24	0.006
XGB_Imp	0.820	24	0.000	**1.000**	**20**	**0.000**	0.945	24	0.000	0.827	24	0.000	0.849	24	0.000

Table 6 Feature selection methods comparison with different machine learning models on DataSet-B

	F1-Score			**ROC-AUC**			**PR-AUC**			**MCC**			**G-MEAN**		
	Mean	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**
CatBoostClassifier															
Baseline	0.750	–	–	0.991	–	–	0.769	–	–	0.765	–	–	0.791	–	–
FID-SOM	0.**828**	**7**	**0.014**	**0.997**	**7**	**0.002**	0.866	7	0.011	**0.831**	**7**	**0.013**	**0.870**	**7**	**0.011**
RFE	0.751	7	0.014	0.989	9	0.008	0.765	9	0.008	0.763	9	0.012	0.803	7	0.011
Uni_Chi2	0.711	8	0.022	0.987	9	0.002	0.722	8	0.014	0.727	8	0.019	0.765	8	0.018
Uni_F	0.705	9	0.014	0.987	9	0.002	0.722	9	0.010	0.723	9	0.012	0.759	9	0.011
Uni_MI	0.825	8	0.014	**0.997**	**8**	**0.001**	**0.867**	**8**	**0.013**	0.829	8	0.013	0.864	8	0.011
XGB_Imp	0.752	9	0.008	0.990	10	0.001	0.768	10	0.009	0.761	8	0.008	0.806	9	0.006
RandomForestClassifier															
Baseline	0.622	–	–	0.967	–	–	0.767	–	–	0.660	–	–	0.682	–	–
FID-SOM	**0.835**	**7**	**0.017**	0.978	7	0.002	**0.868**	**7**	**0.004**	**0.839**	**7**	**0.014**	**0.866**	**7**	**0.013**
RFE	0.707	7	0.025	0.966	10	0.002	0.774	7	0.006	0.725	7	0.021	0.760	7	0.021
Uni_Chi2	0.626	10	0.028	0.964	10	0.003	0.763	10	0.009	0.662	10	0.024	0.686	10	0.023
Uni_F	0.626	10	0.033	0.964	10	0.002	0.763	10	0.009	0.662	10	0.029	0.686	10	0.026
Uni_MI	0.826	8	0.020	**0.979**	**8**	**0.002**	0.863	8	0.006	0.832	8	0.017	0.856	8	0.016
XGB_Imp	0.706	7	0.030	0.963	10	0.003	0.776	7	0.008	0.724	7	0.025	0.759	7	0.025
XGBClassifier															
Baseline	0.543	–	–	0.973	–	–	0.641	–	–	0.593	–	–	0.622	–	–
FID-SOM	**0.855**	**7**	**0.000**	**0.998**	**7**	**0.000**	**0.895**	**7**	**0.000**	**0.857**	**7**	**0.000**	**0.889**	**7**	**0.000**
RFE	0.573	7	0.000	0.979	7	0.000	0.639	7	0.000	0.613	7	0.000	0.650	7	0.000
Uni_Chi2	0.606	8	0.000	0.983	8	0.000	0.646	8	0.000	0.640	8	0.000	0.676	8	0.000
Uni_F	0.537	9	0.000	0.978	10	0.000	0.603	9	0.000	0.582	9	0.000	0.621	9	0.000
Uni_MI	0.847	8	0.000	0.997	8	0.000	0.891	8	0.000	0.851	8	0.000	0.877	8	0.000
XGB_Imp	0.577	9	0.000	0.979	7	0.000	0.639	7	0.000	0.613	9	0.000	0.656	9	0.000

Table 7 Feature selection methods comparison with different machine learning models on DataSet-C

	F1-Score			**ROC-AUC**			**PR-AUC**			**MCC**			**G-MEAN**		
	Mean	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**
CatBoostClassifier															
Baseline	0.818	–	–	0.982	–	–	0.805	–	–	0.826	–	–	0.849	–	–
FID-SOM	0.813	21	0.018	**0.986**	**25**	**0.004**	**0.811**	**27**	**0.007**	0.822	21	0.018	0.842	21	0.011
RFE	0.818	23	0.017	0.982	25	0.004	0.803	27	0.007	0.824	23	0.016	**0.855**	**23**	**0.015**
Uni_Chi2	0.809	21	0.009	0.984	27	0.003	0.806	27	0.007	0.819	21	0.009	0.836	21	0.009
Uni_F	0.813	21	0.015	0.982	27	0.003	0.802	27	0.006	0.820	21	0.014	0.847	21	0.012
Uni_MI	0.814	23	0.012	0.982	27	0.004	0.794	27	0.008	0.821	23	0.012	0.848	23	0.010
XGB_Imp	**0.821**	**21**	**0.021**	0.982	25	0.004	0.805	27	0.006	**0.828**	**21**	**0.019**	**0.855**	**21**	**0.019**
RandomForestClassifier															
Baseline	0.815	–	–	0.938	–	–	0.792	–	–	0.825	–	–	0.841	–	–
FID-SOM	0.820	27	0.006	0.935	23	0.011	0.791	23	0.006	0.831	27	0.006	0.839	27	0.010
RFE	0.814	27	0.008	**0.942**	**27**	**0.014**	0.791	27	0.005	0.825	27	0.007	0.837	25	0.009
Uni_Chi2	**0.825**	**25**	**0.010**	**0.942**	**27**	**0.011**	0.789	23	0.006	**0.836**	**25**	**0.009**	**0.844**	**25**	**0.009**
Uni_F	0.819	21	0.009	0.939	23	0.013	**0.794**	**21**	**0.006**	0.794	21	0.006	0.839	21	0.009
Uni_MI	0.822	27	0.011	0.939	21	0.010	**0.794**	**21**	**0.006**	0.833	27	0.011	**0.844**	**25**	**0.009**
XGB_Imp	0.819	25	0.011	0.939	27	0.011	0.789	23	0.007	0.829	25	0.011	0.839	25	0.009
XGBClassifier															
Baseline	0.796	–	–	0.974	–	–	0.803	–	–	0.810	–	–	0.841	–	–
FID-SOM	0.801	21	0.000	**0.987**	**23**	**0.000**	0.824	23	0.000	0.833	23	0.000	**0.856**	**27**	**0.000**
RFE	0.800	27	0.000	0.984	21	0.000	0.827	27	0.000	**0.834**	**27**	**0.000**	**0.856**	**27**	**0.000**
Uni_Chi2	**0.806**	**23**	**0.000**	0.985	23	0.000	0.827	21	0.000	**0.834**	**21**	**0.000**	**0.856**	**21**	**0.000**
Uni_F	0.800	27	0.000	0.985	21	0.000	**0.833**	**27**	**0.000**	0.841	27	0.000	**0.856**	**27**	**0.000**
Uni_MI	0.799	27	0.000	0.980	27	0.000	0.827	25	0.000	**0.834**	**25**	**0.000**	**0.856**	**25**	**0.000**
XGB_Imp	0.796	27	0.000	0.984	27	0.000	0.821	27	0.000	0.827	27	0.000	**0.856**	**27**	**0.000**

Table 8 Performance comparison

Method	Number of times method performs the best	Percentage (%)
Baseline	2/45	4.44
FID-SOM	**32/45**	**71.11**
Uni_Chi2	8/45	17.78
Uni_F	6/45	13.33
Uni_MI	8/45	17.78
RFE	5/45	11.11
XGB_Imp	6/45	13.33

Graphical visualizations of the results are presented in Figs. 5, 6, 7, 8 and Fig. 9. Each figure shows the results for each measure: F1 (Fig. 5), MCC (Fig. 6), G-Mean (Fig. 7), AUCPR (Fig. 8), and AUCROC (Fig. 9). Visualization contains dependencies of criterion on a number of features. We can observe that the success of the feature selection strongly depends on data. Additionally, the selected machine learning algorithm has an impact on the feature selection performance as well.

FID-SOM method overcomes other feature selection methods. Figures 5, 6, 7, 8 and Fig. 9 visually show that FID-SOM outperforms other methods in many cases.

Discussions

Even though Dataset-C is very popular among researchers, it is difficult to compare our work with other papers. The primary challenges arise because some papers do not specify the splitting ratio or the type of split-random or time-based. In many cases, studies addressing the fraud-detection problem in credit card transactions unrealistically evaluate the performance of the proposed method by splitting the dataset into train and test using random split (see [47, 48]). This assumes that the data is independently and identically distributed over time. However, in real-world scenarios, credit card transaction data often exhibits temporal dependencies and non-stationarity, making this assumption flawed. As a result, models trained on one time period may not perform well on data from another due to shifts in transaction patterns, fraudulent activities, or changes in user behavior. This is important because in time-series data, observations are typically dependent on previous observations, and shuffling the data randomly could lead to data leakage. Comparison of the results when the data is split using time-based approach can be found in Table 9.

However, in order to compare FIDSOM performance against other published work, we did a split using stratified random split, selecting 80% of data points for the training set and 20% for testing on DataSet-C. For the comparison, we selected only those papers that clearly specified the splitting share (see Table 10). We did not include papers that use data balancing methods like oversampling or undersampling before splitting the data into training and testing datasets, e.g., in this way, technically removing imbalance problems, which is not possible in real-life scenarios. Applying sampling methods before splitting the dataset into train and test sets leads to deceptively high results.

The baseline performance of four widely recognized ensemble learning models, specifically focusing on their F1 scores, is presented in the paper [49]. The models evaluated include Random Forest, XGBoost, LightGBM, and CatBoost. No additional feature engineering or optimization steps were implemented for this baseline assessment, ensuring that the F1 scores reflect the models' classification abilities. The F1 scores are as follows:

- Random Forest achieved an F1 score of 0.846.
- XGBoost obtained a slightly lower score of 0.840.
- LightGBM trailed with a score of 0.749.
- CatBoost led the group with an F1 score of 0.853.

These results provide an initial benchmark for further model refinement.

Conclusions and future research

In this paper, we suggest a novel feature selection method called FID-SOM (feature selection for imbalanced data using SOM). The uniqueness of the proposed method is in forming a new dataset containing the best matching units of the trained SOM as vectors of attributes corresponding to the initial features. These attributes are sorted based on variance in descending order. By keeping the desired number of attributes holding the

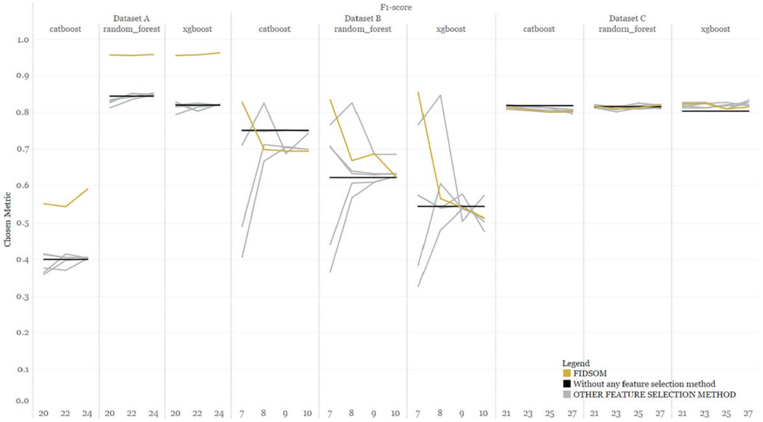

Fig. 5 Results with F1

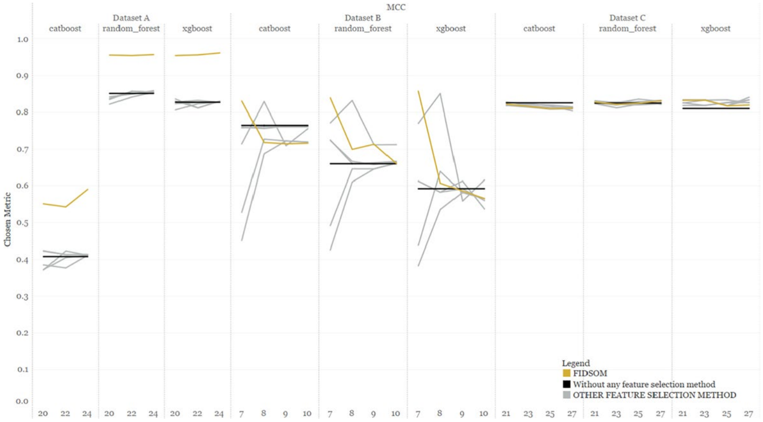

Fig. 6 Results with MCC

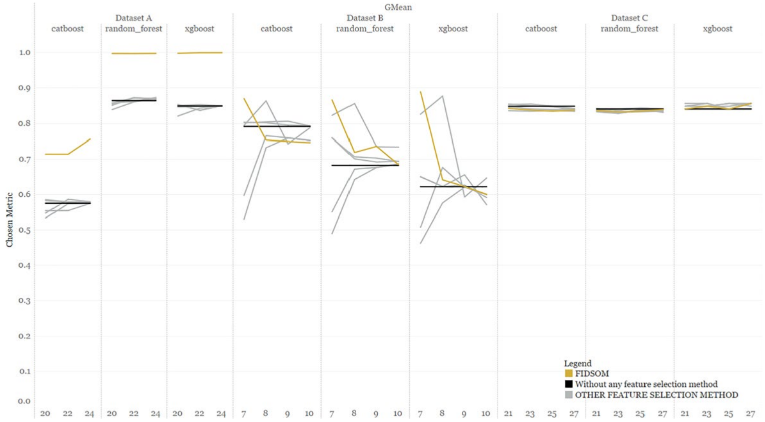

Fig. 7 Results with G-Mean

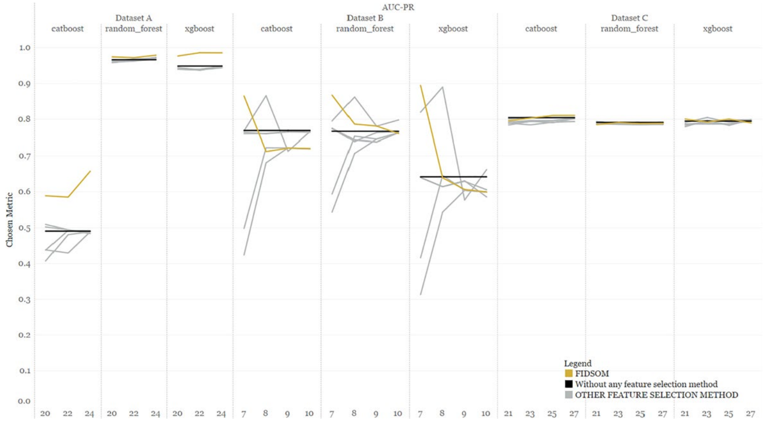

Fig. 8 Results with AUCPR

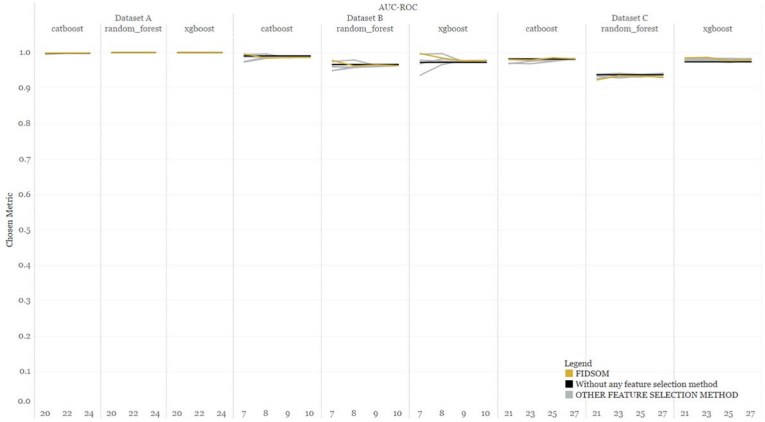

Fig. 9 Results with AUCROC

Table 9 Comparison with other papers splitting data in a time-based manner with a share of 70/30 for training and testing

Paper	Year	F1-Score	Recall	Precision
[50]	2019	0.82	0.73	0.93
[51]	2023	0.84	0.74	0.97
FIDSOM*	2024	0.85	0.76	0.97

*FIDSOM with XGB classifier selecting 23 features. Data split is done by selecting 70% of the first data points for training and 30% remaining data points for testing

Table 10 Comparison with other papers splitting data randomly with a share of 80/20 for training and testing

Paper	Year	F1-Score	Recall	Precision
[52]	2024	0.85	0.84	0.86
[53]	2023	0.85	0.76	0.98
FIDSOM**	2024	0.88	0.82	0.95

** FIDSOM with RF selecting 23 features. Data split is done by randomly selecting 80% of the data points for training and 20% of data points for testing

highest variability, we select a smaller number of features corresponding to those attributes for further analysis.

FID-SOM was compared with univariate feature selection methods utilizing the F-test, χ^2 test and mutual information, the recursive feature elimination method, and the XGB Importance method. The goodness of the feature selection methods was evaluated using F1 score, MCC, G-Mean, AUC-PR, and AUC-ROC metrics when performing XGBoost, CatBoost, and Random algorithms on three datasets.

The success of the method was evaluated by counting how many times the method was selected as the best-performing method. The proposed FID-SOM method has demonstrated noteworthy achievement by reaching a success rate of 71.11% (Table 8). This accomplishment is not only meaningful because of its ability to perform on par with, if not surpass, existing methodologies but also shows its innovative potential. Notably, the FID-SOM method is highlighted when compared with the performance of the second-best method, which yielded a success rate of 17.78% (Table 8).

FID-SOM is designed to address the challenge of dimensionality reduction in scenarios characterized by highly imbalanced data. Due to its discovered effectiveness, a novel FID-SOM method will become one of the often-used feature selection methods that allow fraud detection practitioners to solve complex classification problems successfully (Tables 9 and 10).

For future research, we are going to analyze the optimization algorithms of FID-SOM. While the current study applies the standard competitive learning mechanism for weight vector adaptation, future work could explore advanced optimization techniques, such as incorporating class-specific learning rates or integrating additional constraints on weight updates to further improve the SOM's performance on imbalanced datasets. These extensions could enhance the model's ability to detect fraud while maintaining computational efficiency.

Abbreviations

Baseline	Machine Learning without feature selection (using all available features)
FID-SOM	Proposed feature selection for imbalanced data using SOM
Uni_Chi2	Univariate feature selection utilizing χ^2
Uni_F	Univariate feature selection utilizing F-test
Uni_MI	Univariate feature selection utilizing mutual information
Uni_RFE	Recursive feature elimination
XGB_Imp	XGB importance method
PR	Area under the precision-recall (PR) curve
ROC	Area under the ROC curve

Author contributions

D.B.—Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing—Original Draft, Visualization. G.D.—Conceptualization, Writing—Review and Editing.

Funding

Not applicable.

Data availability

Synthesizing Credit Card Transactions [39] can be accessed at https://data.world/ealtman/synthetic-credit-card-transactions. Data generated with Sparkov Generation Tool [40] can be accessed at https://www.kaggle.com/datasets/kartik2112/fraud-detection. Real Data of the Credit Card Transactions can be found at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

Declarations

Competing interests

The authors declare no competing interests.

Received: 2 April 2024 Accepted: 17 December 2024
Published online: 28 December 2024

References

1. Jiang S, Dong R, Wang J, Xia M. Credit card fraud detection based on unsupervised attentional anomaly detection network. Systems. 2023. https://doi.org/10.3390/systems11060305.
2. Li W, Wu C-S, Ruan S-M. CUS-RF-based credit card fraud detection with imbalanced data. J Risk Anal Crisis Response. 2022;12(3):110–23. https://doi.org/10.54560/jracr.v12i3.332
3. Mai T-D, Hoang K, Baigutanova A, Alina G, Kim S. Customs fraud detection in the presence of concept drift. arXiv. arXiv:2109.14155. 2021.
4. Bourdonnaye F, Daniel F. Evaluating categorical encoding methods on a real credit card fraud detection database. 2021.
5. Breskuvienė D, Dzemyda G. Imbalanced data classification approach based on clustered training set. In: Dzemyda G, Bernatavičienė J, Kacprzyk J, editors. Data science in applications. Cham: Springer; 2023. p. 43–62. https://doi.org/10.1007/978-3-031-24453-7_3.
6. Chen Y, Ma L, Yu D, Zhang H, Feng K, Wang X, Song J. Comparison of feature selection methods for mapping soil organic matter in subtropical restored forests. Ecol Indic. 2022;135: 108545. https://doi.org/10.1016/j.ecolind.2022.108545.
7. Bashir S, Khattak IU, Khan A, Khan FH, Gani A, Shiraz M. A novel feature selection method for classification of medical data using filters, wrappers, and embedded approaches. Complexity. 2022. https://doi.org/10.1155/2022/8190814.
8. Li Y, Li T, Liu H. Recent advances in feature selection and its applications. Knowl Inform Syst. 2017;53(3):551–77. https://doi.org/10.1007/s10115-017-1059-8.
9. Xiao Q, Li H, Tian J, Wang Z. Group-wise feature selection for supervised learning. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2022. pp. 3149–53. https://doi.org/10.1109/ICASSP43922.2022.9746666.
10. Dai J, Liu Q, Zou X, Zhang C. Feature selection based on fuzzy combination entropy considering global and local feature correlation. Inform Sci. 2024;652: 119753. https://doi.org/10.1016/j.ins.2023.119753.
11. Yin L, Ge Y, Xiao K, Wang X, Quan X. Feature selection for high-dimensional imbalanced data. Neurocomputing. 2013;105:3–11. https://doi.org/10.1016/j.neucom.2012.04.039.
12. Chen H, Li T, Fan X, Luo C. Feature selection for imbalanced data based on neighborhood rough sets. Inform Sci. 2019;483:1–20. https://doi.org/10.1016/j.ins.2019.01.041.
13. Huang S, Chen H, Li T, Chen H, Luo C. Feature selection via minimizing global redundancy for imbalanced data. Appl Intell. 2022;52(8):8685–707. https://doi.org/10.1007/s10489-021-02855-9.

14. Aguayo L, Barreto GA. Detection of anomalies and novelties in time series with self-organizing networks. In: International Workshop on Self-Organizing Maps: Proceedings. 2007. https://doi.org/10.2390/biecoll-wsom2007-125.
15. McElwee S, Cannady J. Improving the performance of self-organizing maps for intrusion detection. In: SoutheastCon. 2016. pp. 1–6. https://doi.org/10.1109/SECON.2016.7506766.
16. Nair M, Cappello T, Dang S, Kalokidou V, Beach MA. RF fingerprinting of lora transmitters using machine learning with self-organizing maps for cyber intrusion detection. In: 2022 IEEE/MTT-S International Microwave Symposium—IMS. 2022. pp. 491–4. https://doi.org/10.1109/IMS37962.2022.9865441.
17. Sulaima MF, Saharani S, Ahmad A, Hassan EE, Bohari ZH. Investigation of energy demand correlation during pandemic using self-organizing map algorithm. Int J Artif Intell. 2022;11(4):1333–43. https://doi.org/10.11591/ijai.v11.i4.pp1333-1343.
18. Bouziane A, Kharroubi J, Zarghili A. Probabilistic self-organizing maps for text-independent speaker identification. TELKOMNIKA Telecommun Comput Electron Control. 2018;16(1):250–8. https://doi.org/10.12928/TELKOMNIKA.v16i1.7559.
19. Dorrer MG, Fomin AV, Loginov DA. Clustering of participants in the MaxBonus loyalty system using Kohonen's self-organizing maps. J Phys Conf Ser. 2020;1679(4): 042010. https://doi.org/10.1088/1742-6596/1679/4/042010.
20. Primandari AH, Ikasakti NA. Job applicants clustering using self-organizing map. Bull Soc Inform Theory Appl. 2017;1(2):60–71. https://doi.org/10.31763/businta.v1i2.28.
21. Widayanti R, Madenda S, Wibowo EP, Anwar K. SOM-SIS approach to auto summary of clustering results on university academic performance. TELKOMNIKA Telecommun Comput Electron Control. 2023;21(1):104–12. https://doi.org/10.12928/telkomnika.v21i1.24238.
22. Maiorana F. Feature selection with Kohonen self organizing classification algorithm. Zenodo. 2008. https://doi.org/10.5281/zenodo.1078959.
23. Kohonen T. The self-organizing map. Proc IEEE. 1990;78(9):1464–80. https://doi.org/10.1109/5.58325.
24. Dzemyda G, Sabaliauskas M, Medvedev V. Geometric MDS performance for large data dimensionality reduction and visualization. Informatica. 2022;33(2):299–320. https://doi.org/10.15388/22-INFOR491.
25. Dzemyda G, Kurasova O, Žilinskas J. Multidimensional data visualization. Springer Optimization and Its Applications. Berlin: Springer; 2013. https://doi.org/10.1007/978-1-4419-0236-8.
26. Tian J, Azarian MH, Pecht M. Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm. PHM Soc Eur Conf. 2014. https://doi.org/10.36001/phme.2014.v2i1.1554.
27. D'Urso P, Giovanni LD, Massari R. Smoothed self-organizing map for robust clustering. Inform Sci. 2020;512:381–401. https://doi.org/10.1016/j.ins.2019.06.038.
28. Jović A, Brkić K, Bogunović N. A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2015. pp. 1200–5. https://doi.org/10.1109/MIPRO.2015.7160458.
29. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;Mar(3):1157–82.
30. Tianqi C, Carlos G. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining. 2016. pp. 785–94.
31. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. pp. 785–94. https://doi.org/10.1145/2939672.2939785.
32. Dorogush AV, Ershov V, Gulin A. Catboost: gradient boosting with categorical features support. arXiv:1810.11363 [stat.ML]. 2018.
33. Breiman L. Random forests. Mach Learn. 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.
34. Dal Pozzolo A, Caelen O, Borgne Y-AL, Waterschoot S, Bontempi G. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl. 2014;41(10):4915–28.
35. Aung MH, Seluka PT, Fuata JTR, Tikoisuva MJ, Cabealawa MS, Nand R. Random forest classifier for detecting credit card fraud based on performance metrics. In: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). 2020. pp. 1–6. https://doi.org/10.1109/CSDE50874.2020.9411563.
36. Hajek P, Abedin MZ, Sivarajah U. Fraud detection in mobile payment systems using an XGBoost-based framework. Inform Syst Front. 2023;25:1985–2003. https://doi.org/10.1007/s10796-022-10346-6.
37. Chen Y, Han X. Catboost for fraud detection in financial transactions. In: 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE). 2021. pp. 176–9. https://doi.org/10.1109/ICCECE51280.2021.9342475.
38. Breskuvienė D, Dzemyda G. Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions. Int J Computers Commun Control. 2023. https://doi.org/10.15837/ijccc.2023.3.5433.
39. Altman ER. Synthesizing credit card transactions. 2019.
40. Harris B. Sparkov: synthetic data generation tool for Apache Spark. 2022. https://github.com/namebrandon/Sparkov. Accessed 30 July 2023.
41. Vorndran M, Schütz A, Bendix J, Thies B. Current training and validation weaknesses in classification-based radiation fog nowcast using machine learning algorithms. Artif Intell Earth Syst. 2022. https://doi.org/10.1175/AIES-D-21-0006.1.
42. Bullock H, Edwards M. Temporal constraints in online dating fraud classification. In: Proceedings of the 9th International Conference on Information Systems Security and Privacy, Lisbon, Portugal. SCITEPRESS - Science and Technology Publications; 2023. pp. 535–42. https://doi.org/10.5220/0011689000003405.
43. Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G. Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Trans Neural Netw Learn Syst. 2018;29(8):3784–97. https://doi.org/10.1109/TNNLS.2017.2736643.

44. Mai T-D, Hoang K, Baigutanova A, Alina G, Kim S. Customs fraud detection in the presence of concept drift. arXiv.2109.14155. 2021.
45. Assaad RH, Fayek S. Predicting the price of crude oil and its fluctuations using computational econometrics: deep learning, LSTM, and convolutional neural networks. Econom Res Financ. 2021;6:119–37. https://doi.org/10.2478/erfin-2021-0006.
46. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
47. Lei Y-T, Ma C-Q, Ren Y-S, Chen X-Q, Narayan S, Huynh ANQ. A distributed deep neural network model for credit card fraud detection. Financ Res Lett. 2023;58: 104547. https://doi.org/10.1016/j.frl.2023.104547.
48. Islam MA, Uddin MA, Aryal S, Stea G. An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes. J Inform Secur Appl. 2023;78: 103618. https://doi.org/10.1016/j.jisa.2023.103618.
49. Salekshahrezaee Z, Leevy JL, Khoshgoftaar TM. The effect of feature extraction and data sampling on credit card fraud detection. J Big Data. 2023;10(1):6.
50. Fiore U, De Santis A, Perla F, Zanetti P, Palmieri F. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Inform Sci. 2019;479:448–55.
51. Fanai H, Abbasimehr H. A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection. Expert Syst Appl. 2023;217: 119562.
52. Zhao C, Sun X, Wu M, Kang L. Advancing financial fraud detection: self-attention generative adversarial networks for precise and effective identification. Financ Res Lett. 2024;60: 104843.
53. Jiang S, Dong R, Wang J, Xia M. Credit card fraud detection based on unsupervised attentional anomaly detection network. Systems. 2023;11(6):305.

Publisher's Note