Learning suitable data representation for credit card fraud detection algorithms

Sylvain Lejamble, Ilham Alloui, Sébastien Monnet, Flavien Vernier

▶ **To cite this version:**

Sylvain Lejamble, Ilham Alloui, Sébastien Monnet, Flavien Vernier. Learning suitable data representation for credit card fraud detection algorithms. 2023 6th International Conference on Data Mining and Big Data Analytics (DMBDA 2023), Jul 2023, Shanghai, China. hal-04446399

HAL Id: hal-04446399
https://hal.science/hal-04446399

Submitted on 26 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning suitable data representation for credit card fraud detection algorithms

Sylvain Lejamble*†, Flavien Vernier†, Ilham Alloui† and Sebastien Monnet†
*Research & Development Department
SaGa Corp, Paris, France
† Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance (LISTIC)
Université Savoie Mont Blanc (USMB), Annecy, France
{sylvain.lejamble,flavien.vernier,ilham.alloui,sebastien.monnet}@univ-smb.fr

***Abstract*—With the recent pandemic, credit card and even contactless payment have gained significant popularity. The elevated frequency of card usage, along with the lack of diligence among customers, has resulted in an increase in stolen or counterfeit cards, often leading to fraudulent activities. This emphasized the importance of real-time detection of abnormal banking transactions for card issuers. Automated analysis of transaction logs is a prevalent approach to address this challenge, often involving the comparison of incoming transactions to a database containing genuine and fraudulent transactions. Methods such as the Mahalanobis distance have proven to be efficient when seeking for similarities between high dimensional data. However, the challenge lies in the fact that credit card logs contain both categorical and tabular data, which poses compatibility issues with the Mahalanobis algorithm. This study explores the effectiveness of finding alternative data representations as a pre-processing step to enable the utilization of algorithms that were previously unsuitable for such data types. The research conducted in this work is established on an exclusive credit card logs dataset.**

***Keywords*-fraud detection; credit card; Mahalanobis distance, Auto-Encoder, Principal Component Analysis, multi-modality**

I. Introduction

According to the Nilson Report [1], the estimated cost of fraud reached $28.8 billion in 2020. Experts predict that this figure will exceed $400 billion over the next decade. Consequently, the issue is significant, with the primary entities affected being credit institutions and banks. Fraud can occur in numerous ways (such as insurance fraud, government aid fraud, etc.), and it is an ongoing race between fraudsters and detection/prevention systems that has no end in sight. Credit card fraud is one of the most bothersome types of fraud for bank clients, as it is highly prevalent and it directly targets their funds.

Machine learning (ML) is expected to be a useful tool to assist banks in detecting abnormal phenomena. However, in many financial institutions, such methods are considered too innovative to be investigated and are often not even considered. Banks often favor manual analysis of logs, which necessitates large teams due to the extensive volume of data requiring analysis. Typically, this processing involves algorithms extracting fraud reports from all transactions. These algorithms, which are employed by traditional banks, consist of very basic rules (designed by an expert) that incorporate thresholds and statistics. These rules have a number of shortcomings. They are not frequently updated, making them vulnerable to data drift. Moreover, their simplicity makes them insufficient to identify complex or dynamic patterns.

Machine Learning has demonstrated its effectiveness in solving complex problems with often improved results compared to traditional methods. When it comes to fraud detection, two prominent families of methods stand out: supervised learning and anomaly detection (unsupervised). We position our work at the intersection of these two methods by performing semi-supervised learning. In [2] the authors used the Mahalanobis distance to calculate an anomaly score for an incoming transaction to transactions of similar users (grouped using Peer Group Analysis) based on the assumption that an anomalous behavior would deviate strongly from its peer group. However, this work only considers continuous peer group features and disregards binary-encoded categorical features such as transaction type, payment type, and card presence, which our compliance team has found to be valuable information (binary columns are unusable in the Mahalanobis algorithm because of an non invertible matrix problem). To address this limitation, we propose a method that preserves the information in binary columns by compressing them and using them in a Mahalanobis classifier.

This study aims to assess the efficiency of different compression algorithms in order to use a Mahalanobis based classifier that cannot be fitted on multi modal (binary + tabular) data. Our approach compare not only genuine but also past instance of fraud to incoming transaction to reduce the occurrence of false positives (instances classified as anomalies but not actual fraud). Simultaneously, this inclusion improves the algorithm's ability to identify complex patterns that pose challenges for existing rule-based models reducing false negatives. The rationale behind is that the cost associated with a false positive is expected to be significantly lower than that of a false negative.

The paper is organized as follows. The first part provides an overview of existing methods; in the second part we introduce our algorithm, discuss the challenges associated with multimodality and present our proposed solutions. Finally, we discuss about the results of our experiments conducted on an exclusive credit card fraud dataset, along with their respective outcomes.

II. RELATED WORK

There are two main sets of techniques for fraud detection: supervised and unsupervised anomaly detection. Supervised approaches require a history of fraudulent transactions and often result in fewer false positives than anomaly detection models. Unsupervised methods seek to find outliers (i.e., anomaly) among the data, assuming that fraudulent transactions are sufficiently different from other data to be detected. This can result in detecting outlier data but not necessarily fraud, leading to a high false positive rate. Anomaly detection techniques also have advantages, as they are often less sensitive to data drift and can still produce good results even if the patterns of fraud are not the same as those in the past.

A. Supervised fraud detection techniques

For years, data mining has been widely used to detect suspicious changes in user behavior [3], [4].

Decision Trees(DTs) [5] have demonstrated their efficiency in classification, even with small training sets. They have been widely used for credit card fraud detection [6]. A very large portion of fraud classification is also using DTs in boosting algorithm such as AdaBoost [7] or LightGBM [8].

Neural Networks (NNs) are highly efficient in learning features for classification, as demonstrated by [9], who employed a Multi-Layer Perceptron to perform Non-Linear Discriminant Analysis. Long short-term memory models are highly effective for sequential data, such as credit card payment flows [10]. NNs are frequently utilized in ensemble learning [11], and have become one of the most effective approaches for credit card fraud detection due to the varying strengths of different classifiers [12].

B. Anomaly detection for fraud detection

Profiling involves the collection of user information to construct a database that captures the patterns of user behavior. This information can be used to check if current behavior is different from the expected(learned) behavior [13], [14], [15]. More recently, user profiles are sent as features for a third party classifier [16]. Break Point Analysis is the detection of an anomalous sequence from past data. It has been used for money laundering [17] and is now used implicitly in decision tree or neural networks algorithms.

Most of unsupervised techniques chose to compute an anomaly score from the new transaction with respect to the spending profile of the user or user-associated cluster [2], [18].

C. Alternative representation for better classification

Learning an alternative representation of the data can help a lot before sending data to a third party classifier. For example, Convolutional Neural Networks delivers impressive results in image classification by virtue of their convolutional layers, which act as feature extractors that alter the structure of the data.

The Self-Organizing Map (SOM) can be employed to learn a two-dimensional representation of the training data. In previous studies [19], [20], a SOM was trained on historical genuine and fraud transactions. The SOM then assigns labels (e.g., genuine or fraud) to new transactions based on their positions on the map.

Auto-Encoders (AEs) [21] are highly effective in learning a compressed representation that captures the most valuable information in the data. If a new data significantly deviates from the learned latent distribution, the reconstruction quality will be affected, resulting in a high reconstruction loss. Either the reconstruction loss or the latent representation itself can be employed in unsupervised anomaly detection [22]. AEs can also be employed in semi-supervised learning [23], [24] to improve the result of the binary classifier.

Principal Component Analysis (PCA) [25] is a powerful method for data compression that retains as much information as possible, and it is also well-suited for handling sensitive data. For instance, the Université Libre de Bruxelles (ULB) employed PCA to process data for fraud detection, which allowed them to share the data without compromising its sensitivity [26]. In another study [27], PCA was utilized to reduce processing time and yielded to better results than AEs models.

There exist various methods for altering feature representation. This paper aims to compare these methods and determine their efficiency as a preprocessing step for an other algorithm.

D. Data level techniques

The imbalanced nature of the dataset poses a significant challenge in fraud detection. Since fraud instances represent only a small portion of the data, traditional supervised methods may yield unsatisfactory results. Techniques such as oversampling the positive class or undersampling the negative class have demonstrated improved performance compared to scenarios without data pre-processing.

Synthetic Minority Oversampling Technique (SMOTE) or Adaptive Synthetic (ADASYN) can be employed to upsample the minority class, which is an effective feature engineering method that is often associated with enhanced results [27], [28], [11].

Undersampling can also help a lot to balance the dataset in order to make learning more relevant. In [29] the authors

balanced the data distribution using Gaussian mixture undersampling and applied it to credit card fraud detection.

III. FRAUD DETECTION SYSTEM

As explained in Section IV-A, most Fraud Detection Systems (FDS) do not detect fraud, but anomalies. While detecting anomalies remains relevant, it deviates slightly from our initial problem. In this paper we propose a template based approach to deal with this issue. The algorithm computes a distance between templates and current transactions to determine whether to accept or reject them. We consider it as an end-to-end classifier but our final goal is to incorporate the template deviation score as a feature in an ensemble classification technique alongside other anomaly features, similar to the methodology demonstrated by the authors in [30].

A. Classification and Templates

To deal with this problem, we chose to compare each transaction to two groups of data (belonging to the "Fraud" or "Genuine" class), which we will call "templates". By doing so, we greatly decrease the instances where a transaction appears unusual but does not correspond to a known instance of fraud.

The templates are either real or simulated transactions that are divided in two classes, "Fraud" and "Genuine". For the templates of normal transactions, they are randomly selected from recent transactions (by default, all transactions are considered normal because statistically, over 99% of them are normal). The fraud templates are either generated through simulations (described in Section IV-B) or taken from our set of client reported transactions. The sets of templates are updated every month to prevent drift (see Section IV-D).

Upon obtaining these templates, cluster membership of new data is determined by computing the mean distance to all data points within the cluster (the formula is detailed in Section III-B). We chose to try several distances such as Euclidean and Mahalanobis distances.

B. Cluster Membership

Let $\vec{x} = (x_1, x_2, x_3, \ldots, x_p)^T$ be the transaction to classify, p being the number of features, M the template matrix(n templates,p features), and M_i be the ith template of the matrix with $M_i = (M_{i1}, M_{i2}, M_{i3}, \ldots, M_{ip})^T$.

Euclidean: The normalized Euclidean distance can be used to compute the distance between the new data and the templates. The distance is defined as follows:

$$d(\vec{x}, M) = \frac{\sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} \frac{(\vec{x}_j - M_{ij})^2}{\sigma_j^2}}}{n}.$$

where σ_i is the standard deviation of $\vec{x}$.

Despite the known limitations of the Euclidean distance in high-dimensional spaces due to the curse of dimensionality [31], we chose to test it in order to evaluate the relevance of compression algorithms.

Mahalanobis: To calculate membership in a cluster, we can use the Mahalanobis distance [32] for its ability to take into account the similarity and variances between data series. The main idea is to calculate the Mahalanobis distance between the mean values of templates and the current transaction.

The distance between the vector $\vec{x}$ and a set of templates M with covariance matrix Σ and mean value $\vec{\mu} = (\mu_1, \mu_2, \mu_3, \ldots, \mu_p)^T$ is defined as follows:

$$d(\vec{x}, M) = \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}.$$

Two mean vectors are calculated respectively for fraud and genuine transaction templates:

$$\begin{aligned} \vec{\mu}(M_{fraud}) &= (\mu_{f1}, \mu_{f2}, \mu_{f3}, \ldots, \mu_{fp})^T, \\ \vec{\mu}(M_{genuine}) &= (\mu_{g1}, \mu_{g2}, \mu_{g3}, \ldots, \mu_{gp})^T. \end{aligned}$$

Let $f(\vec{x})$ be the function that assigns a class to a transaction $\vec{x}$ such that:

$$f(\vec{x}) = \begin{cases} 1\ (Fraud) & \text{if } d(\vec{x}, M_{fraud}) > d(\vec{x}, M_{genuine}), \\ 0\ (Genuine) & \text{otherwise.} \end{cases}$$

C. Issues with templates

The major issue with using traditional distances lies in the multimodality of the data. Our dataset includes both continuous and binary categorical data. The benefit of utilizing a distance metric such as Mahalanobis is its capability to factor in the correlation between data points. As a result, there will be a strong correlation between identical categorical fields, rendering such comparisons considerably more relevant. However this advantage can also bring a problem: The use of the Mahalanobis distance requires each template matrix to be invertible (in order to calculate the inverse of the covariance matrix Σ^{-1}). Our data contains many categorical variables (+40) that are deemed valuable (see Section IV-C) and therefore require one-hot encoding to retain their information. Consequently, when the number of templates is relatively limited, there is a strong likelihood that there is a column where all variables are identical and the covariance matrix Σ is consequently non-invertible. We have considered and compared different solutions to overcome this problem:

- **Increasing the number of templates:** The possibility of increasing the number of templates was not explored, as the required quantity would be too large.
- **Removing problematic columns:** To make the matrix invertible and use the Mahalanobis distance, categorical columns can be eliminated.
- **Encoding information differently:** The problem being just a matter of form, there are algorithms (SOM, PCA,

Auto-Encoders, etc.) that can compress data while preserving as much information as possible.

We chose to evaluate the following methods, as each has its own strengths and weaknesses.

1) Removing boolean features: The removal of binary columns (dimension goes from +50 to less than 10) can make the correlation matrix invertible because we remove the multi modality. It also enables the possibility of using different distance metrics that are not well-suited for high-dimensional or mixed-types data because 90% of our features are one-hot encoded. The challenge lies in the fact that these columns contain a wealth of information (purchase category, payment method, etc.), which seems crucial in comparing user profiles. Despite a study indicating the significance of categorical variables in feature selection for classification (as illustrated in Figure 1), we have decided to proceed with testing this approach to confirm our hypothesis that boolean features are mandatory.

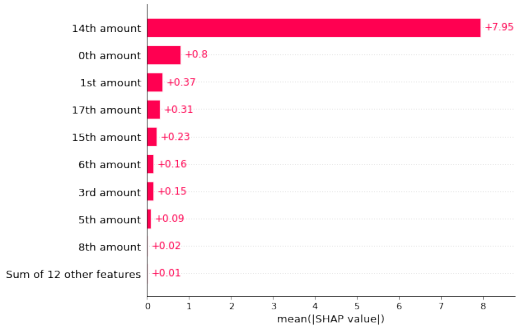

Figure 1. SHapley Additive exPlanations (Shap) value of features using XGB classifier. Shap values can be seen as how much the feature contributed to the decision of the algorithm. Note that Shap values of categorical features are summed. Feature names are hidden to guarantee the integrity of our partner.

2) Encoding information: Principal Component Analysis (PCA): PCA is a common technique to visualize or preprocess any high dimensional data. PCA aims at transforming the data into a new coordinate system by grouping highly correlated columns. We used PCA to compress the data and go through the template issue described in Section III-C.

3) Encoding information: Variational Auto-Encoder (VAE): Auto-Encoders have widely demonstrated their performance in finding a subspace that preserves as much information as possible [**?**]. Learned latent space represents a condensed variant of the data. Compression alters the form of the data, thereby enabling it to be fed into the Mahalanobis binary Classifier.

IV. EXPERIMENTAL SETTINGS

Our partner provided us the opportunity to study real transactions from credit cards that cover from September 2020 to July 2022. The dataset is about +1.5 million business card transactions from +100 countries. The features are not transformed making interesting the use of data engineering techniques such as feature engineering. We have an extensive set of features (+50) such as the amount of the transaction, the country of the transaction, the type of payment (card present/card not present/internet, etc.). In contrast to prior research studies [33] that rely on user responses to a questionnaire, we made a deliberate decision to exclude any private information as a feature to minimize the potential impact of individual biases. Therefore our algorithms rely only on anonymized transaction data and peer group information. Given that these transactions originate from business card transactions, the data is highly sensitive to demographic factors and market fluctuations. This presents both a drawback (as training a single model is not applicable) and an advantage (as it is indicative of the industry reality). We decided not to compare our algorithms with the fraud dataset from ULB [26] as it only covers transactions over a span of 2 days, which is not reflective of the dynamic nature of the data in a shifting environment.

A. Real world evaluation constraints

In France, upon detecting fraud, it is necessary to submit a report of suspicion to TRACFIN. However, TRACFIN does not provide feedback on whether the transaction is indeed fraudulent or not. In most of the publications, when authors discuss “fraud detection”, they are referring to the identification of anomalies that could possibly indicate the presence of fraud. As we do not have a 100% trustful labels, the evaluation of FDS remains really hard without a human in the loop. As explained in [34] “The financial institution can not find out if a money laundering suspect was guilty of the crime”. Labels that are 100% trustful are issued by the compliance team when a client said he has been victim of a fraudster. It is worth noting that it is only a small part of the labels because proven frauds happens only when a client notifies the card issuer.

B. Fraud simulations

When there is uncertainty in labeling, simulation can be an effective solution. The objective is to use actual transactions as a foundation and incorporate only the fraudulent ones, resulting in a synthetic dataset where all features are managed. However, simulating data is a complex task, and it is crucial that the data are derived from known scenarios. Otherwise it makes no sens to use the trained algorithm on live production data.

In [35] the authors employed scenario matching to detect fraud in an Enterprise Resource Planning (ERP). According to the French “Observatoire de la Sécurité des Moyens de Paiement”, there are several relatively common scenarios. These scenarios are divided into two categories depending on whether the card is physically present at the time of purchase or not.

Card present frauds: The frauds in which the card is physically present represent about 20% (including 9.2% for Automated Teller Machines (ATM) withdrawals and 10% for Point Of Sell (POS) payments). This can be caused by a loss or theft of the card but also by counterfeit cards. We extracted two scenarios from those data:

- Scenario 1-P: The user has lost or had his/her card stolen, then a third party withdraws cash from an ATM.
- Scenario 2-P: The user has lost or had his/her card stolen, then a third party made a POS transaction.

These scenarios aim to reproduce real events. We notice that in these scenarios, the user can no longer make payments with a physical card since he no longer has one. He can however continue to make payments on the Internet.

Card not-present frauds: "Card not-present frauds" are much more frequent (approx. 75%). They are often the result of a misuse of card numbers. The following fraud scenarios were identified:

- Scenario 1-NP: The user has been hacked. The hacker uses the card for internet payments with reasonable amounts to avoid detection. 1/3 of his/her new expenses will be related to the hacker. This forces the algorithms to take into account the spending profile of a user to detect a deviant transaction.
- Scenario 2-NP: The user has been hacked. The hacker makes large purchases which arouses the suspicions of the owner who immediately blocks his card.

We are also leveraging additional simulation scenarios that capture other known fraud cases, but their confidentiality is essential to guarantee the integrity of our partner.

C. Feature engineering

Many publications focus on minimized and anonymized ULB dataset, and therefore, barely touch on feature engineering. We chose to discuss the feature engineering we performed, without going into too much detail for security reasons.

Profiling: According to [14], it is not possible to detect fraudulent behavior if only one transaction is available. The user profile is a valuable information for detecting sudden changes in user's behavior. The Recency-Frequency-Monetary (RFM) analysis [36] enables comparison of the current transaction with a similar group of users, helping to avoid bias towards any individual in the algorithm. RFM analysis gives the customer a score based on the amount, frequency, and recency of his transactions. As we process data in real-time, it is necessary to keep a variable correlated to the customer's spending history. According to [37], it can be beneficial to add information about the customer's last transaction to the features of the current transaction, such as the time since the last transaction, the amount of the transaction, the country where it was made, etc. Therefore, we decided to extract the RFM features for 7 and 30-day windows. With the training strategy described in Section IV-D, RFM features of the current batch are calculated from the past data at each batch in order to simulate real working conditions.

No personal features: In a traditional bank, when a transaction is flagged as fraudulent, the compliance team will verify the profile and investigate the individual. It may be tempting to add these user features to the transaction features for the classification algorithm. In Personalized Approaches (PA) [38], fraud detection algorithms take into account features related to individuals, which often improve detection results [6]. However, we want the algorithm to only operate on multiple transaction data with the multiple-user approach (MUA). To prevent the algorithm from promoting individual discrimination, we will refrain from using any personal data. It is important to note that MUA yields poor results when there are few or no transaction for a given individual.

D. Training strategy

Temporal data is often subject to drift caused by external factors such as seasonality or changes in user behavior. For this reason, we have decided to simulate model training on a monthly basis by generating training batches as illustrated in Figure 2. Model M_t is trained on the two months (B_{t-2} and B_{t-1}) preceding the current month B_t. Fraud templates can be obtained in real working conditions as we can have delayed supervised samples from the feedback of compliance team. To simplify the experiment and focus solely on studying the compression algorithm, we have decided not to consider ensemble learning [12], [39]. This strategy ensures some adaptability to data drift while remaining relatively simple.

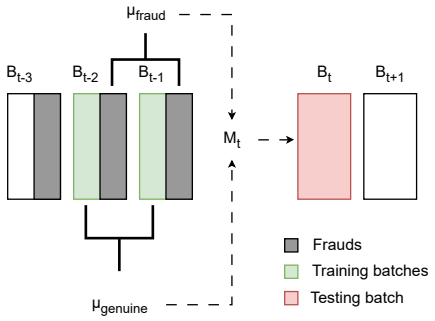

Figure 2. Learning Strategy

The two months prior to the current month are used for training or template generation. Considering that the models are trained from scratch for each iteration, we have a total of 19 distinct models and training processes (excluding incomplete months).

The experiments were conducted on 4 datasets from 4 different simulations. For the first two datasets, frauds were

generated on a monthly basis in order to obtain a relatively stable distribution between the different batches. In these datasets, frauds represent 1.15% and 0.55% respectively. In dataset 2, fraud scenarios are activated and deactivated randomly between batches, making it more realistic. In the 3rd and 4th datasets, frauds are added by randomly selecting a card IDs. This card is then flagged fraudulent and transactions will be labeled as fraud based on the activated scenarios. While this approach makes the distribution more realistic (distribution is not flat across each batches), it also increases the complexity of the task.

E. Metrics

Given the high degree of imbalance in our dataset, it is essential to employ suitable metrics. Accuracy alone is insufficient since if 99% of the data belong to Class A and if 100% of the data are classified as Class A, the resulting accuracy would be 99%, thereby overlooking the 1% of potentially critical data that requires classification. We therefore decided to use the F1 score, which is an average of precision and recall. As the False Positive Rate (FPR) is important, it seemed obvious that this metric should be included in our benchmark. However, as the classifier improves, the FPR approaches 0 due to the overwhelming size of the negative class in comparison to the positive class. That is the reason why we use False Discovery Rate (FDR) instead of FPR.

$$FDR = \frac{False\ Positive}{False\ Positive + True\ Positive}.$$

Miss rate or False Negative Rate (FNR) is also important as we want to maximize the number of detected frauds among frauds.

$$FNR = \frac{False\ Negative}{False\ Negative + True\ Positive}.$$

Additionally, to account for the fact that our classifier is designed to excel in positive instances, we incorporated the Area Under the Precision-Recall Curve (AUPRC) metric. We used macro-average for recall and F1 score in order to keep in mind the unbalanced nature of the data. Results are averaged across 19 batches of training to take in consideration the training strategy of Section IV-D and make results easier to compare. All the metrics are also averaged within the 4 datasets.

F. Baseline

We chose to designate logistic regression, random forest and gradient boosting as three baseline methods. These algorithms are easy to implement and offer valuable insights into how traditional approaches perform on diverse datasets. The algorithms are trained using fraud templates and ALL genuine transactions from the last two months. They are then tested on the current month, similar to other methods employed in this study.

Table I
EXPERIMENTAL RESULTS

	Recall	F1	FDR	FNR	AUPRC
Logistic Regression	0.782	0.815	n/a	0.435	0.574
Random forest	0.941	0.94	0.067	0.107	0.836
LightGBM	**0.992**	0.983	0.032	**0.011**	**0.959**
SOM	0.898	0.894	0.182	0.196	0.667
Euclidean + DROP	0.528	0.39	0.974	0.577	0.022
Euclidean + PCA-25	0.528	0.39	0.974	0.577	0.022
Euclidean + VAE-25	0.867	0.587	0.808	0.095	0.175
Mahalanobis + DROP	0.645	0.485	n/a	0.514	0.04
Mahalanobis + PCA-25	0.883	0.927	**0.001**	0.234	0.772
Mahalanobis + VAE-25	0.981	**0.986**	0.009	0.038	0.956

As the aim of this paper is to compare several compression methods and how they impact algorithm performance, we trained every compression algorithm on the same data (genuine transactions of the last two months). Templates and test transactions are transformed to the new coordinates system using the trained model.

1) Self-Organizing Map: We also deemed it relevant to compare our results with those obtained through unsupervised training using a SOM. In [19] the authors used a 70x70 neuron map with 5 clients, each having 105 transactions, each with 7 features.

By experimentation, we found that increasing the size of the map beyond that described by [40] did not produce significantly better results, while requiring a substantial increase in computation time. Therefore, we set the grid size to $5 * \sqrt{k}$ (where k is the number of features). We used a step count of $50 * grid\_size^2$; beyond this, improvements were minimal. The map is trained on the templates of the last two months. As SOM have the ability to directly predict the class, we did not employ them as a preprocessing step for binary classifiers but as standalone classifier.

2) PCA: We considered several configurations of dimensionality reduction by principal component analysis. The experiments show optimal results for a reduction to between 20 and 25 components. PCA is trained using genuine transactions only. Fraud and Genuine templates are then reduced using trained PCA.

3) VAE: A VAE was employed instead of an AE due to its capability to generate a latent space distribution, which can be sampled from. The VAE has 30 neurons on the first layer and 15/25 neurons on the hidden layer depending on the selected configuration. Experiments showed the best results with 15 epochs, a batch size of 64, a Rectified Linear Unit (ReLU) activation and a hidden layer of 25 neurons. We trained the VAE using the latest two months of genuine transactions only, and used the latent representation of the templates to compute the distances.

G. Results

Traditional supervised methods yielded average results, whereas gradient boosting performance were notably impressive. LightGBM excelled in classifying data from batch simulations, but exhibited average performance when faced with emerging patterns, as observed in dataset 3 and 4.

Self-Organizing Maps demonstrated promising results; however, the training and inference processes are time-consuming from an industrial standpoint. A thorough investigation would be worthwhile as SOMs are highly sensitive to hyperparameters.

In Table I, the metrics from the Euclidean/DROP and Euclidean/PCA approaches exhibit striking similarity, attributed lack of relevance of such distance on binary values. The VAE, on the other hand, has convincingly demonstrated its efficiency in preserving vital information and has reaffirmed our initial intuition regarding the mandatory incorporation of categorical variables into the learning process.

The Mahalanobis classifier demonstrated superior performance when applied to the latent space generated by the VAE. In contrast to LightGBM, the Mahalanobis classifier exhibited relatively consistent results across all datasets, indicating promising real life outcomes, particularly for dataset 3 and 4.

During the training process of the VAE, the distribution of fraud reconstruction errors tends to diverge from the distribution of genuine reconstruction errors, indicating that both distributions can be differentiated by this method. However, the distributions are still too entangled to employ the reconstruction error as a threshold-based classifier.

V. CONCLUDING REMARKS

Based on our experiments, we observed a substantial improvement of third part classifier by exploring alternative data representations. This enables the utilization of algorithms that are not applicable to categorical data and facilitates the learning of representations for "normal" or "fraudulent" transactions. These representations can also assist in generating data for oversampling methods. Our experiments demonstrated the most favorable outcomes when compressing input features into a 25-dimensional latent representation and feeding them into a binary Mahalanobis classifier.

As stated in [36], "Any particular study is always a snapshot in time and space". Therefore, it should not be forgotten that these results were partially generated by scenario-driven simulations. This study would benefit greatly from more annotated data provided by specialists to make the results more robust, but such data are rare and very challenging to obtain. It should be noted that due to the private nature of the dataset, the replication of the results presented in this study is not feasible. The public datasets available online often inadequately reflect reality due to their limited duration, security encoding, and other factors. This is one of the main obstacles to the development of high-performing bank fraud detection systems.

We will continue to work on this dataset, using the results obtained through our study as a tool score for more comprehensive algorithms. An in-depth investigation into which algorithms perform effectively on specific datasets and the underlying reasons behind their success appears to be intriguing. The inferred data, which may deviate significantly from all templates but closely resemble specific instances of fraud, highlights the need for a better solution than averaging the scores associated with every template. The inferred data may be far from all templates but very close to only certain frauds. Thus fuzzy classification appears promising in this domain, as highlighted in previous studies such as [41]. Oversampling seems also very promising as many publications describe it as a necessary processing step and it is often associated with improved results.

REFERENCES

[1] "Nilson Report," Tech. Rep., 2020. [Online]. Available: https://nilsonreport.com/

[2] D. J. Weston, D. J. Hand, N. M. Adams, C. Whitrow, and P. Juszczak, "Plastic card fraud detection using peer group analysis," Advances in Data Analysis and Classification, vol. 2, no. 1, pp. 45–62, Apr. 2008.

[3] T. Fawcett and F. Provost, "Adaptive Fraud Detection," Data mining and knowledge discovery, pp. 7–15, 1997.

[4] K. Burbeck and S. Nadjm-Tehrani, "Data Mining Techniques in Fraud Detection," Information security technical report, 2007.

[5] J. R. Quinlan, "Induction of Decision Trees," Machine learning, vol. 1, pp. 81–106, 1986.

[6] R.-C. Chen, M.-L. Chiu, Y.-L. Huang, and L.-T. Chen, "Detecting Credit Card Fraud by Using Questionnaire-Responded Transaction Model Based on Support Vector Machines," in Intelligent Data Engineering and Automated Learning. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3177, pp. 800–806.

[7] R. Saia and S. Carta, "Distributed data mining in credit card fraud detection," IEEE Intelligent Systems and their Applications, 1999.

[8] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," IEEE Access, vol. 8, pp. 25 579–25 587, 2020.

[9] J. R. Dorronsoro, F. Ginel, C. Sgnchez, and C. S. Cruz, "Neural fraud detection in credit card operations," IEEE transactions on neural networks, vol. 8, no. 4, pp. 827–834, 1997.

[10] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," Expert Systems with Applications, vol. 100, pp. 234–245, Jun. 2018.

[11] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," IEEE Access, vol. 10, pp. 16 400–16 407, 2022.

[12] A. D. Pozzolo, "Adaptive Machine Learning for Credit Card Fraud Detection," PhD Thesis, Université Libre de Bruxelles, 2015.

[13] T. Fawcett and F. Povost, "Automated Design of User Profiling Systems for Fraud Detection," Knowledge Discovery and Data Mining, 1998.

[14] R. J. Bolton and D. J. Hand, "Unsupervised profiling methods for fraud detection." Credit scoring and credit control, 2001.

[15] C. S. Hilas and J. N. Sahalos, "User Profiling for Fraud Detection in Telecommunication Networks," in 5th International Conference on Technology and Automation, 2005, journal Abbreviation: 5th International Conference on Technology and Automation.

[16] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," Expert Systems With Application 51, 2016.

[17] T. E. Senator, "Ongoing management and application of discovered knowledge in a large regulatory organization: a case study of the use and impact of NASD Regulation's Advanced Detection System (RADS)," in Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000, pp. 44–53.

[18] M. Carminati, R. Caron, F. Maggi, I. Epifani, and S. Zanero, "BankSealer: A decision support system for online banking fraud analysis and investigation," Computers & Security, vol. 53, pp. 175–186, Sep. 2015.

[19] J. T. Quah and M. Sriganesh, "Real Time Credit Card Fraud Detection using Computational Intelligence," Expert systems with applications, vol. 35, no. 4, pp. 1721–1732, 2008.

[20] V. Zaslavsky and A. Strizhak, "Credit card fraud detection using self-organizing maps," Information and Security, vol. 18, p. 48, 2006.

[21] D. Rumelhart, G. Hinton, and R. Williams, "Learning Internal Representations by Error Propagation," in Readings in Cognitive Science. Elsevier, 1988, pp. 399–421.

[22] M. Zamini and G. Montazer, "Credit Card Fraud Detection using autoencoder based clustering," in 2018 9th International Symposium on Telecommunications (IST). Tehran, Iran: IEEE, Dec. 2018, pp. 486–491.

[23] J. Zou, J. Zhang, and P. Jiang, "Credit Card Fraud Detection Using Autoencoder Neural Network," 2019, arXiv:1908.11553 [cs, stat].

[24] S. Misra, S. Thakur, M. Ghosh, and S. K. Saha, "An Autoencoder Based Model for Detecting Fraudulent Credit Card Transaction," Procedia Computer Science, vol. 167, pp. 254–262, 2020.

[25] H. Hotelling, "Analysis of a complex of statistical variables into principal components." Journal of Educational Psychology, vol. 24, no. 6, pp. 417–441, Sep. 1933.

[26] ULB, "Credit Card Fraud Detection," 2021. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[27] H. Zhou, L. Wei, G. Chen, P. Lin, and Y. Lin, "Credit Card Fraud Identification Based on Principal Component Analysis and Improved Adaboost Algorithm," in 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS). Chongqing, China: IEEE, Dec. 2019, pp. 507–510.

[28] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient Resampling for Fraud Detection During Anonymised Credit Card Transactions with Unbalanced Datasets," in 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC). Leicester, UK: IEEE, Dec. 2020, pp. 426–433.

[29] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, "GMM-based Undersampling and Its Application for Credit Card Fraud Detection," in 2019 International Joint Conference on Neural Networks (IJCNN). Budapest, Hungary: IEEE, Jul. 2019, pp. 1–8.

[30] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," Information Sciences, vol. 557, pp. 317–331, May 2021.

[31] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the Surprising Behavior of Distance Metrics in High Dimensional Space," in Database Theory — ICDT 2001, G. Goos, J. Hartmanis, J. Van Leeuwen, J. Van Den Bussche, and V. Vianu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, vol. 1973, pp. 420–434.

[32] P. C. Mahalanobis, "On the Generalized Distance in Statistics," Proceedings of the National Institute of Sciences (Calcutta), 1935.

[33] Rong-Chang Chen, Shu-Ting Luo, Xun Liang, and V. Lee, "Personalized Approach Based on SVM and ANN for Detecting Credit Card Fraud," in 2005 International Conference on Neural Networks and Brain, vol. 2. Beijing, China: IEEE, 2005, pp. 810–815.

[34] D. J. Hand, "Fraud detection in telecommunications and banking: Discussion of Becker, Volinsky, and Wilks (2010) and Sudjianto et al.(2010)," Technometrics, vol. 52, pp. 34–38, 2010.

[35] A. K. Islam, M. Corney, G. Mohay, A. Clark, S. Bracher, T. Raub, and U. Flegel, "Fraud detection in ERP systems using scenario matching," in IFIP International Information Security Conference. Springer, 2010, pp. 112–123.

[36] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," Data Mining and Knowledge Discovery, vol. 18, no. 1, pp. 30–55, Feb. 2009.

[37] D. K. Tasoulis, N. M. Adams, D. J. Weston, and D. J. Hand, "Mining Information from Plastic Card Transaction Streams," Proceedings in computational statistics, 2008.

[38] R.-C. Chen, T.-S. Chen, and C.-C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," International Journal of Pattern Recognition and Artificial Intelligence, vol. 20, no. 02, pp. 227–239, 2006.

[39] M. Zareapoor and P. Shamsolmoali, "Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier," Procedia Computer Science, vol. 48, pp. 679–685, 2015.

[40] J. Vesanto, SOM toolbox for Matlab 5. Espoo: Helsinki University of Technology, 2000, oCLC: 58272790.

[41] T. K. Behera and S. Panigrahi, "Credit Card Fraud Detection: A Hybrid Approach Using Fuzzy Clustering & Neural Network," in 2015 Second International Conference on Advances in Computing and Communication Engineering. Dehradun, India: IEEE, May 2015, pp. 494–499.