2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Credit Card Fraud Detection with Subspace Learning-based One-Class Classification

Zaffar Zaffar*, Fahad Sohrab*, Juho Kanniainen* and Moncef Gabbouj*
*Faculty of Information Technology and Communication Sciences, Tampere University, Finland
{zaffar.zaffar, fahad.sohrab, juho.kanniainen, moncef.gabbouj}@tuni.fi

***Abstract*—In an increasingly digitalized commerce landscape, the proliferation of credit card fraud and the evolution of sophisticated fraudulent techniques have led to substantial financial losses. Automating credit card fraud detection is a viable way to accelerate detection, reducing response times and minimizing potential financial losses. However, addressing this challenge is complicated by the highly imbalanced nature of the datasets, where genuine transactions vastly outnumber fraudulent ones. Furthermore, the high number of dimensions within the feature set gives rise to the "curse of dimensionality". In this paper, we investigate subspace learning-based approaches centered on One-Class Classification (OCC) algorithms, which excel in handling imbalanced data distributions and possess the capability to anticipate and counter the transactions carried out by yet-to-be-invented fraud techniques. The study highlights the potential of subspace learning-based OCC algorithms by investigating the limitations of current fraud detection strategies and the specific challenges of credit card fraud detection. These algorithms integrate subspace learning into the data description; hence, the models transform the data into a lower-dimensional subspace optimized for OCC. Through rigorous experimentation and analysis, the study validated that the proposed approach helps tackle the curse of dimensionality and the imbalanced nature of credit card data for automatic fraud detection to mitigate financial losses caused by fraudulent activities.**

***Index Terms*—Credit card fraud detection, financial data processing, one-class classification, subspace learning.**

I. Introduction

The Federal Trade Commission has recently reported an alarming increase in credit card fraud reports and the revenue lost due to such frauds in the past few years [1]. One of the key factors of such an increase in credit card fraud is the digitalization of commerce because of the COVID-19 outbreak and the shutdown of the whole world [2], [3]. Credit card fraud has existed since the invention of payment cards, and different policies were formulated and brought into practice from time to time to reduce the losses incurred by such frauds. The address verification system, keeping a scoring record of positive and negative lists to identify and prevent high-risk transactions [4], and the use of Card Verification Value (CVV) by Visa and Card Verification Code (CVC) by MasterCard [5] are a few of the examples of the preventive policies. As for the detective approach, many Machine Learning (ML) models, such as Support Vector Machines (SVM), logistic regression, random forest [6], artificial neural networks, k-nearest neighbors (kNN) [7] and Self-Organizing Maps (SOM) [8], have been implemented for this cause. The uptrend in fraud cases and lost revenue, despite these policies, clearly shows that the previous set of measures, both preventive and detective, is not enough.

In order to have a better solution that can effectively and efficiently mitigate the losses due to these frauds, we have to understand the shortcomings of the previous approaches as well as the challenges in the credit card fraud detection problem in general. Credit card fraud detection is a binary classification problem having two classes: normal (or positive class) and fraudulent (or negative class). A very basic property and one of the main issues in these problems is that the data is highly imbalanced [9], owing to the fact that billions of card transactions take place every month worldwide, and a significantly smaller amount of transactions are fraudulent. To deal with the data imbalance issue, the ML models that are in practice have used sampling techniques; that is, a sample from the majority class, based on some sampling criterion, is taken [10] or instances for the minority class are synthetically generated based on some criterion [11] so that the number of instances in both classes is made equal. In some cases, an approach based on both of the sampling techniques is used to have a balanced dataset [12].

Another property of fraud detection problems is that the fraudulent activities and techniques evolve with time [13]. Any method used by the fraudsters is identified by the anti-fraud team of the respective organization, and efforts are made to stop further losses through the same fraudulent technique. Consequently, personnel with the aim of gaining unlawful advantage of people or systems (or both) try to come up with new ideas and techniques. The ML algorithms that have been implemented for this purpose can only model the fraudulent techniques that are already in practice; that is, they cannot model, and hence, detect, the fraud that will be carried out by methods that are not existent and are yet to be invented. Therefore, we need a model that can also detect, predict, and stop fraud by the methods that will be invented in the future.

One-Class Classification (OCC) algorithms, on the other hand, take data from only a single (positive or the normal) class for training, which is usually available in abundance, and form a boundary around the positive class (or between the two classes). These algorithms classify everything that lies outside the inferred boundary as a negative class object. These algorithms have been implemented in many different domains and have proved to be a good solution with good performance for the respective problem. The examples of such domains include but are not limited to bot detection on

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

Twitter [14], spoofing detection [15], [16], video surveillance [17], machine fault detection for predictive maintenance [18], hyper-spectral image analysis and classification [19], and Myocardial Infarction (MI) detection [20].

To address the above-mentioned challenges in fraud detection problems and to resolve the curse of dimensionality by embedding the feature extraction into the training phase of the algorithm and letting the model extract a discriminative set of features, we propose to use a set of OCC algorithms that are ideal for the highly imbalanced dataset and can effectively model and detect the fraudulent transactions carried out by to-be-invented techniques. For this purpose, we experimented with several OCC models to find a more efficient way to reduce the losses by such frauds.

II. Methodology

In the OCC setting, data from the target (positive) class is used to develop an optimal boundary between the target data and outliers. Depending on the model, the structure of the decision boundary varies. For instance, the One-Class Support Vector Machine (OCSVM) has a hyperplane [21], the Support Vector Data Description (SVDD) has a hyper-sphere [22], and the Ellipsoidal Subspace Support Vector Data Description (ESSVDD) has an ellipsoidal boundary [23] differentiating the two class (fraudulent and normal transaction) data from each other. A general overview of the credit card fraud detection system with an OCC algorithm is depicted in Figure 1. The Subspace Support Vector Data Description (SSVDD), is the SVDD-based model where the data is projected, using a projection matrix $\mathbf{Q}$, from the original *D* dimensions to the lower *d*-dimensional subspace iteratively during the training [24]. $\mathbf{Q}$, incorporated with the matrix $\mathbf{S_Q}$, representing the geometric information of the data in the subspace, is employed to find the optimized set of features in the Graph-embedded Subspace Support Vector Data Description (GESSVDD) [25]. For data vectors represented by $\mathbf{x}_i \in \mathbb{R}^D$, where $i = 1, 2, ..., N$, the mathematical formulation of the GESSVDD problem is as follows:

$$\begin{aligned} &min \quad R^2 + C\sum_{i=1}^{N}\xi_i \\ &s.t.: \left\|\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i - \mathbf{u}\right\|_2^2 \leq R^2 + \xi_i, \\ &\xi_i \geq 0 \quad i = 1, 2, ..., N, \end{aligned} \tag{1}$$

where N is the number of data points, R is the radius, and $\mathbf{u} = \mathbf{S_Q}^{-\frac{1}{2}}\mathbf{a}$ is the center of the hyper-sphere in the subspace ($\mathbf{a}$ is the center in the original feature space). The variable ξ_i represents the slack variables, and C denotes the trade-off between maximizing the margin (enclosing more data points in the boundary) and minimizing the radius. To solve the optimization problem in (1), it is reformulated into a Lagrangian function using the Lagrange multipliers α_i and γ_i.

$$\begin{aligned} L = R^2 &+ C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\,(\,R^2 + \xi_i \\ &- (\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i)^T\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i + 2\mathbf{u}^T\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i - \mathbf{u}^T\mathbf{u}\,) \\ &- \sum_{i=1}^{N}\gamma_i\xi_i. \end{aligned} \tag{2}$$

The solution of (2) provides us with the α_i values for each instance in the dataset. These α_i values, representing the position of a data point in the projected subspace, are important for determining $\mathbf{u}$ and R of the hyper-sphere. If an α value is zero, the data point lies inside the boundary of the hyper-sphere. If an α value falls between 0 and the regularization parameter *C*, such data point, denoted by $\mathbf{s}$, lies on the boundary of the hyper-sphere and is known as a support vector. On the other hand, if an α value exceeds *C*, the data point lies outside the boundary of the hyper-sphere. The radius of the optimal hyper-sphere can be calculated using

$$R = \sqrt{(\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{s})^T\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{s} - 2(\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{s})^T\mathbf{u} + \mathbf{u}^T\mathbf{u}}. \tag{3}$$

To classify any test data vector $\mathbf{x}^*$ into its respective class, it must first be transformed to the lower *d*-dimensional subspace using the same $\mathbf{Q}$ and $\mathbf{S_Q}$ and the distance of $\mathbf{x}^*$ from $\mathbf{u}$ in transformed feature subspace is calculated and checked if it is greater or smaller than the R given in (3). It is classified as a non-fraudulent transaction if it satisfies the following decision rule:

$$\left\|\mathbf{S_Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}^* - \mathbf{u}\right\|_2^2 \leq R^2. \tag{4}$$

The matrix $\mathbf{S_Q}$, having geometric information in the data in the transformed feature subspace, is mathematically represented as:

$$\mathbf{S_Q} = \mathbf{Q}\mathbf{X}\mathbf{L}_x\mathbf{X}^T\mathbf{Q}^T = \mathbf{Q}\mathbf{S}_x\mathbf{Q}^T, \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the data matrix and $\mathbf{L}$ is the matrix representation of the graph. Based on the choice of the $\mathbf{L}$ in (5), there can be many variants of the model. In this study, we have implemented three GESSVDD variants by considering different options for $\mathbf{L}$. These are:

- The first variant, denoted as GESSVDD-I, replaces $\mathbf{L}_x$ with the identity matrix, $\mathbf{I}$.
- The second variant, referred to as GESSVDD-PCA, utilizes the Principal Component Analysis (PCA) graph where $\mathbf{S}_x$ is replaced with $\frac{1}{N}\mathbf{S}_t$. The Scatter matrix, $\mathbf{S}_t$ is derived as

  $$\mathbf{S}_t = \mathbf{X}\mathbf{L}_t\mathbf{X}^T = \mathbf{X}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{X}^T, \tag{6}$$

  where $\mathbf{1}$ is a vector of ones.
- In third variant, denoted by GESSVDD-kNN, the $\mathbf{L}_x$ is replaced with the kNN graph $\mathbf{L}_{kNN}$, where $\mathbf{L}_{kNN} = \mathbf{D}_{kNN} - \mathbf{A}_{kNN}$. In this variant, we use the diagonal

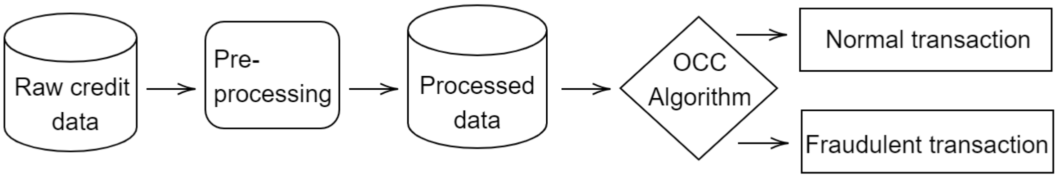

Fig. 1. The flowchart depicting the credit card detection system.

and adjacency matrices, denoted by $\mathbf{D}_{kNN}$ and $\mathbf{A}_{kNN}$, respectively. The elements of the $\mathbf{A}_{kNN}$ matrix are set to 1 if data points $\mathbf{x}_i$ or $\mathbf{x}_j$ are in each other's neighborhood, and 0 otherwise, mathematically expressed as

$$[\mathbf{A}_{ij}] = \left\{ \begin{array}{ll} 1, & if \quad \mathbf{x}_i \in \mathbf{N}_j \quad or \quad \mathbf{x}_j \in \mathbf{N}_i \\ 0, & \text{otherwise} \end{array} \right\}, \quad (7)$$

where $\mathbf{N}_i$ represents the neighborhood of the i^{th} data point.

Furthermore, all these variants have been solved using three different techniques: gradient-based, spectral, and spectral regression technique. In a gradient-based solution, given by $\mathbf{Q} \leftarrow \mathbf{Q} - \eta\Delta L$, the gradient of (2) is used to update the $\mathbf{Q}$. The variable η is a hyper-parameter defining the step of the gradient. In contrast, the other two use eigenvalue and eigenvectors to find the optimized set of features [25]. The variants based on the solution technique are referred to by 'G' for gradient-based, 'E' for spectral, and 'S' for spectral regression. Since there exists no rule of thumb to either maximize or minimize each solution, we experimented with both strategies (denoted by max and min, respectively) for SSVDD and GESSVDD models. In the gradient-based method, the ascending and descending steps in the update rule are used for maximizing and minimizing, respectively. In contrast, for the other two methods, the highest and lowest set of positive eigenvalues and corresponding eigenvectors are chosen for maximization and minimization, respectively. Moreover, different variants of the SSVDD model are implemented based on the regularization term Ψ [24]. A hyper-parameter β, which gives weight to the regularization term Ψ in SSVDD, is tuned during cross-validation. Also, the non-linear version of all these models and variants is implemented using the non-linear projection trick (NPT) [26]. The kernel function utilized in the NPT is the Radial Basis Function (RBF), given by

$$\mathbf{K}_{ij} = exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right), \quad (8)$$

where σ is a hyper-parameter that defines the width of the kernel.

III. Experiments and Discussion

A. Datasets

In this paper, four datasets, all sourced from Kaggle[1] open source dataset repository, are employed for evaluating the OCC models for detecting fraudulent credit card transactions. The first dataset, denoted by Dataset-1, originates from the Worldline and Machine Learning Group at Université Libre de Bruxelles (ULB). It includes credit card transactions made by European cardholders over two days in September 2013. It consists of 29 features and 284,807 transactions with only 492 fraudulent ones (which makes up 0.172% of the dataset). The Dataset-2 contains digital payment transactions with 7 features and 1×10^6 instances, of which 87,403 are fraudulent. The imbalance ratio for this dataset is 0.087. The Dataset-3 is synthetically generated using the Paysim simulator based on a sample of real mobile transactions for one month. It comprises 5 features and 1,048,575 transactions, among which only 1142 are fraudulent, resulting in an imbalance ratio of 0.001. Lastly, a dataset from a bank, available at Kaggle, is utilized, which is denoted by Dataset-4. It includes 112 features and 20,467 transactions, with 5437 being fraudulent, representing 26.6% of the dataset.

B. Experimental Setup

All datasets used in this study are split into 70-30 train-test sets. To handle the high number of instances in the training data, random resampling is performed while maintaining the skewed nature of the data. The resampled training Dataset-1 consists of 344 fraudulent and 2800 normal transactions, leading to a fraudulent-to-normal ratio of 0.12, whereas Dataset-2 to -4 consists of 500 fraudulent and 2500 normal transactions, giving a fraudulent-to-normal ratio of 0.2. Mean and standard deviation are calculated from target class data of the respective original (before resampling) dataset, which is used to normalize the reduced training dataset.

Model training involves tuning hyperparameters using 5-fold cross-validation over the training set. Performance metrics calculated and observed for this study are precision, F1-measure, and geometric mean of sensitivity and specificity (denoted by G-mean), but because of its balanced assessment of positive and negative instances, G-mean is used as an assessment metric during the cross-validation and for model evaluation. The iterative methods' number of iterations and the number of neighbors for the kNN graph are both set to 5. The hyperparameters tuned during cross-validation are given below:

- $C \rightarrow$ [0.1 0.2 0.3 0.4 0.5]
- $d \rightarrow$ [1 2 3 4 5 10 20]
- $\beta \rightarrow$ [0.01, 0.1, 1, 10, 100]
- $\eta \rightarrow$ [0.1, 1, 10, 100, 1000]
- $\sigma \rightarrow$ [0.1, 1, 10, 100, 1000]

[1] https://www.kaggle.com/datasets

TABLE I

RESULTS FOR THE LINEAR VERSIONS OF ALL MODELS FOR ALL DATASETS. PRE STANDS FOR PRECISION, F1 DENOTES F1-MEASURE, G-M REPRESENTS G-MEAN, AND AVG OF G-MEANS IS THE AVERAGE OF G-MEANS ACROSS THE DATASETS. THE HIGHEST PERFORMER IN TERMS OF G-MEAN FOR EACH DATASET IS MARKED IN BOLD. THE MODEL NAMES FOLLOW THE FOLLOWING RULE: FOR GRAPH-BASED: [MODEL]-[GRAPH]-[SOLUTION METHOD]-[MIN/MAX] AND FOR SSVDD: [MODEL]-[REGULARIZATION TERM]-[MIN/MAX].

Model	**Dataset-1**			**Dataset-2**			**Dataset-3**			**Dataset-4**			**Avg of G-means**
	Pre	F1	G-m	Pre	F1	G-m	Pre	F1	G-m	Pre	F1	G-m	
GESSVDD-kNN-G-min	1.000	0.922	**0.906**	0.961	0.523	0.551	0.999	0.998	0.603	0.866	0.664	0.644	**0.676**
GESSVDD-kNN-G-max	1.000	0.994	0.849	0.925	0.766	0.541	0.999	0.998	0.603	0.850	0.803	**0.691**	0.671
GESSVDD-kNN-E-min	0.999	0.996	0.640	0.819	0.373	0.326	1.000	0.838	0.697	0.798	0.743	0.598	0.565
GESSVDD-kNN-E-max	0.999	0.997	0.686	0.953	0.896	**0.692**	0.999	0.998	0.603	0.744	0.687	0.501	0.620
GESSVDD-kNN-S-min	0.998	0.999	0.296	0.906	0.675	0.472	1.000	0.707	**0.728**	0.690	0.623	0.410	0.477
GESSVDD-kNN-S-max	0.998	0.999	0.296	0.949	0.089	0.214	1.000	0.591	0.638	0.752	0.769	0.472	0.405
GESSVDD-PCA-G-min	1.000	0.326	0.432	0.919	0.955	0.282	0.999	0.998	0.595	0.734	0.844	0.074	0.346
GESSVDD-PCA-G-max	1.000	0.619	0.644	0.915	0.953	0.192	0.999	0.998	0.605	0.734	0.844	0.070	0.378
GESSVDD-PCA-E-min	0.999	0.996	0.777	0.915	0.953	0.177	0.999	0.998	0.624	0.734	0.845	0.055	0.408
GESSVDD-PCA-E-max	0.999	0.998	0.720	0.915	0.954	0.186	0.999	0.998	0.595	0.735	0.845	0.082	0.396
GESSVDD-PCA-S-min	0.998	0.999	0.164	0.915	0.954	0.178	0.999	0.998	0.595	0.745	0.852	0.241	0.294
GESSVDD-PCA-S-max	0.998	0.999	0.082	0.915	0.954	0.193	0.999	0.998	0.595	0.744	0.848	0.243	0.279
GESSVDD-I-G-min	0.998	0.998	0.116	0.901	0.861	0.209	0.999	0.998	0.595	0.718	0.716	0.401	0.330
GESSVDD-I-G-max	0.998	0.999	0.000	0.914	0.953	0.123	0.999	0.998	0.595	0.765	0.748	0.527	0.311
GESSVDD-I-E-min	0.999	0.998	0.725	0.914	0.953	0.170	0.999	0.998	0.624	0.748	0.787	0.434	0.488
GESSVDD-I-E-max	0.999	0.999	0.493	0.915	0.954	0.186	0.999	0.998	0.595	0.711	0.668	0.429	0.426
GESSVDD-I-S-min	0.998	0.999	0.164	0.913	0.951	0.086	0.999	0.998	0.595	0.734	0.844	0.085	0.233
GESSVDD-I-S-max	0.998	0.999	0.082	0.916	0.954	0.205	0.999	0.998	0.610	0.735	0.843	0.113	0.252
SSVDD-Ψ_0-min	0.999	0.354	0.438	0.916	0.954	0.204	0.999	0.998	0.407	0.738	0.847	0.150	0.300
SSVDD-Ψ_0-max	0.999	0.291	0.391	0.914	0.954	0.162	0.999	0.998	0.404	0.736	0.846	0.123	0.270
SSVDD-Ψ_1-min	0.998	0.999	0.000	0.916	0.954	0.204	0.999	0.998	0.407	0.738	0.847	0.150	0.190
SSVDD-Ψ_1-max	0.998	0.999	0.000	0.914	0.954	0.162	0.999	0.998	0.404	0.736	0.846	0.123	0.172
SSVDD-Ψ_2-min	0.989	0.092	0.183	0.916	0.954	0.204	0.999	0.998	0.407	0.738	0.847	0.150	0.236
SSVDD-Ψ_2-max	0.989	0.092	0.183	0.914	0.954	0.162	0.999	0.998	0.404	0.736	0.846	0.123	0.218
SSVDD-Ψ_3-min	0.989	0.092	0.182	0.916	0.954	0.204	0.999	0.998	0.407	0.738	0.847	0.150	0.236
SSVDD-Ψ_3-max	0.989	0.092	0.182	0.914	0.954	0.162	0.999	0.998	0.404	0.736	0.846	0.123	0.218
OCSVM	0.999	0.958	0.446	0.941	0.599	0.559	1.000	0.098	0.227	0.582	0.356	0.355	0.397
SVDD	0.993	0.092	0.198	0.915	0.954	0.185	0.999	0.998	0.404	0.742	0.848	0.216	0.251
ESVDD	0.000	0.000	0.000	0.915	0.954	0.187	0.999	0.998	0.595	0.734	0.847	0.035	0.204

C. Results and Discussion

The results for the linear and non-linear versions of the models for all datasets are given in Tables I and II, respectively. From the analysis of these results, it is evident that, for Dataset-1, the approach GESSVDD stands out. Particularly, its linear version and the utilization of the minimization-update rule exhibit notably better performance. The kNN graph and the gradient-based solution technique also outperform their counterparts for this specific dataset. For the other datasets, a non-linear model Graph-embedded One-Class Support Vector Machine GEOCSVM [27] displays better performance compared to other models. However, some models in each dataset exhibit a significantly high or low precision value. These models are either biased towards the positive class (in case of high values), or the boundary formed by these models is very small, and consequently, the normal transactions are forced out of the boundary and classified as fraudulent (in case of low values).

The analysis of the variants of SSVDD with regard to the regularization term, Ψ shows that Ψ_0 produces more favorable results for Dataset-1. For the remaining datasets, all variants yield similar performance. Additionally, an overall assessment based on the average G-mean highlights the supremacy of Ψ_0, which indicates that, for the given datasets, incorporating the regularization term does not provide significant additional insights, and solving the conventional Lagrange equation suffices for optimization.

The analysis of graph-based vs. non-graph-based models shows that the integration of geometric information from the data yields enhanced performance. Consequently, models that leverage graph embeddings outperform those without such added information. Particularly, the kNN graph consistently outperforms other graph options considered for this study. Both eigenvalue decomposition and gradient-based solutions exhibit consistent performance across all datasets.

An investigation based on the average of G-means across datasets is performed to find the best-performing model and other strategies for all four datasets. It is found that GESSVDD with kNN graph, gradient-based solution, and minimization strategy in linear case works well on average for all datasets. It is also found that, on average, the linear version of the models outperforms the counter (non-linear) version. In contrast, the minimization or maximization update rule does not have a significant effect on the performance of the model. Moreover, it is also established that, in general, the kNN graph works better than the other graphs considered in the study.

IV. CONCLUSION

Detecting credit card fraud remains a challenge despite the advances in technology. The imbalanced data and evolving

TABLE II

RESULTS FOR THE NON-LINEAR VERSIONS OF ALL MODELS FOR ALL DATASETS. PRE STANDS FOR PRECISION, F1 DENOTES F1-MEASURE, G-M REPRESENTS G-MEAN, AND AVG OF G-MEANS IS THE AVERAGE OF G-MEANS ACROSS THE DATASETS. THE HIGHEST PERFORMER IN TERMS OF G-MEAN FOR EACH DATASET IS MARKED IN BOLD. THE MODEL NAMES FOLLOW THE FOLLOWING RULE: FOR GRAPH-BASED: [MODEL]-[GRAPH]-[SOLUTION METHOD]-[MIN/MAX] AND FOR SSVDD: [MODEL]-[REGULARIZATION TERM]-[MIN/MAX].

Model	**Dataset-1**			**Dataset-2**			**Dataset-3**			**Dataset-4**			**Avg of**
	Pre	F1	G-m	Pre	F1	G-m	Pre	F1	G-m	Pre	F1	G-m	**G-means**
GESSVDD-kNN-G-min	0.998	0.999	0.329	0.918	0.955	0.264	0.999	0.705	0.570	0.478	0.278	0.283	0.362
GESSVDD-kNN-G-max	0.970	0.011	0.070	0.910	0.199	0.314	0.999	0.767	0.559	0.592	0.435	0.345	0.322
GESSVDD-kNN-E-min	0.294	0.000	0.007	0.846	0.304	0.346	0.999	0.919	0.576	0.555	0.381	0.321	0.313
GESSVDD-kNN-E-max	0.000	0.000	0.000	0.846	0.304	0.346	0.999	0.919	0.576	0.555	0.381	0.321	0.311
GESSVDD-kNN-S-min	0.999	0.992	0.522	0.849	0.441	0.365	0.999	0.913	0.580	0.800	0.005	0.052	0.380
GESSVDD-kNN-S-max	0.999	0.975	**0.550**	0.849	0.441	0.365	0.999	0.913	0.580	0.571	0.009	0.066	0.390
GESSVDD-PCA-G-min	1.000	0.000	0.003	0.913	0.954	0.110	0.999	0.945	0.592	0.726	0.770	0.347	0.263
GESSVDD-PCA-G-max	0.999	0.975	0.446	0.542	0.001	0.019	0.999	0.945	0.592	0.575	0.407	0.335	0.348
GESSVDD-PCA-E-min	0.988	0.092	0.181	0.978	0.552	0.591	0.999	0.970	0.547	0.632	0.026	0.114	0.358
GESSVDD-PCA-E-max	0.998	0.999	0.000	0.795	0.338	0.301	0.999	0.972	0.533	0.634	0.026	0.113	0.237
GESSVDD-PCA-S-min	0.999	0.975	0.417	0.966	0.640	0.627	0.996	0.129	0.227	0.332	0.031	0.121	0.348
GESSVDD-PCA-S-max	0.998	0.976	0.341	0.966	0.640	0.627	0.999	0.958	0.579	0.332	0.031	0.121	0.417
GESSVDD-I-G-min	0.250	0.000	0.003	0.474	0.008	0.060	0.999	0.953	0.555	0.803	0.072	0.192	0.203
GESSVDD-I-G-max	0.999	0.975	0.494	0.538	0.004	0.043	0.999	0.980	0.601	0.547	0.356	0.323	0.365
GESSVDD-I-E-min	0.222	0.000	0.007	0.849	0.346	0.360	0.999	0.969	0.557	0.553	0.009	0.068	0.248
GESSVDD-I-E-max	0.998	0.999	0.000	0.849	0.346	0.360	0.999	0.975	0.497	0.447	0.036	0.133	0.247
GESSVDD-I-S-min	0.998	0.046	0.151	0.879	0.149	0.268	0.999	0.969	0.557	0.713	0.599	0.469	0.361
GESSVDD-I-S-max	0.998	0.999	0.000	0.879	0.149	0.268	0.999	0.969	0.557	0.358	0.035	0.129	0.238
SSVDD-Ψ_0-min	0.999	0.091	0.215	0.951	0.277	0.384	1.000	0.016	0.089	0.723	0.733	0.398	0.272
SSVDD-Ψ_0-max	0.996	0.092	0.208	0.890	0.315	0.380	0.996	0.171	0.244	0.723	0.733	0.398	0.308
SSVDD-Ψ_1-min	0.992	0.092	0.194	0.951	0.277	0.384	1.000	0.016	0.089	0.723	0.733	0.398	0.266
SSVDD-Ψ_1-max	0.999	0.059	0.173	0.890	0.315	0.380	0.996	0.171	0.244	0.723	0.733	0.398	0.299
SSVDD-Ψ_2-min	0.992	0.092	0.193	0.951	0.277	0.384	1.000	0.016	0.089	0.723	0.733	0.398	0.266
SSVDD-Ψ_2-max	0.999	0.999	0.386	0.890	0.315	0.380	0.996	0.171	0.244	0.723	0.733	0.398	0.352
SSVDD-Ψ_3-min	0.993	0.092	0.197	0.951	0.277	0.384	1.000	0.016	0.089	0.723	0.733	0.398	0.267
SSVDD-Ψ_3-max	0.996	0.092	0.208	0.890	0.315	0.380	0.996	0.171	0.244	0.723	0.733	0.398	0.308
OCSVM	1.000	0.034	0.131	0.955	0.577	0.574	1.000	0.021	0.102	0.830	0.822	0.662	0.367
SVDD	0.993	0.092	0.198	0.214	0.014	0.073	0.934	0.003	0.039	0.284	0.083	0.179	0.122
ESVDD	0.999	0.089	0.214	0.348	0.005	0.048	0.999	0.945	0.592	0.185	0.026	0.108	0.241
GEOCSVM	0.999	0.066	0.183	1.000	0.936	**0.937**	1.000	0.926	**0.791**	0.859	0.828	**0.714**	**0.656**
GESVDD	1.000	0.089	0.215	0.997	0.923	0.913	0.999	0.882	0.593	0.849	0.817	0.694	0.604

fraud techniques contribute to this difficulty. Additionally, the curse of dimensionality is a challenge that poses problems for feature extraction. To address these issues without altering data proportions synthetically, we employed OCC algorithms, particularly subspace learning-based models. These models efficiently learn patterns in the data and predict fraudulent transactions, reducing losses caused by fraud.

In this research, we used four imbalanced datasets from Kaggle, resampled while retaining the data's imbalanced nature. We trained 60 model variants, including OCSVM, GEOCSVM, SVDD, GESVDD [27], ESVDD, SSVDD, and GESSVDD. Results show that, on average, the linear GESSVDD with kNN graph, gradient-based solution, and minimization-update rule outperforms other models for all datasets. The G-mean metric is used for model evaluation based on its balanced assessment of both positive and negative instances.

Due to the high complexity of the models, high computational power is required to train the models, calling for improved complexity and efficiency. Additionally, the lack of real-world datasets due to data privacy rules hinders the interpretability and extraction of meaningful features by handcrafted methods. Future work involves investigating other kernel types and graphs for existing methods for improved results. In the future, we plan to adapt Multi-modal Subspace Support Vector Data Description (MSSVDD) [28] for credit card fraud detection.

ACKNOWLEDGEMENT

This work was supported by the NSF-Business Finland project AMALIA. Foundation for Economic Education (Grant number: 220363) funded the work of Fahad Sohrab at Haltian.

REFERENCES

[1] F. T. Commission, "Identity theft reports," 2023. Available online: https://public.tableau.com/app/profile/federal.trade.commission/viz/Identity TheftReports/TheftTypesOverTime (accessed on 10 August 2023).

[2] M. Habibpour, H. Gharoun, M. Mehdipour, A. Tajally, H. Asgharnezhad, A. Shamsi, A. Khosravi, and S. Nahavandi, "Uncertainty-aware credit card fraud detection using deep learning," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106248, 2023.

[3] G. Zhang, Z. Li, J. Huang, J. Wu, C. Zhou, J. Yang, and J. Gao, "Efraudcom: An e-commerce fraud detection system via competitive graph neural networks," *ACM Trans. Inf. Syst.*, vol. 40, mar 2022.

[4] 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.

[5] K. J. Barker, J. D'amato, and P. Sheridon, "Credit card fraud: awareness and prevention," *Journal of financial crime*, vol. 15, no. 4, pp. 398–410, 2008.

[6] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011. On quantitative methods for detection of financial fraud.

[7] R. Asha and S. K. KR, “Credit card fraud detection using artificial neural network,” *Global Transitions Proceedings*, vol. 2, no. 1, pp. 35–41, 2021.

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

[9] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, “Learned lessons in credit card fraud detection from a practitioner perspective,” *Expert systems with applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[10] 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*, pp. 1–8, IEEE, 2019.

[11] D. S. Sisodia, N. K. Reddy, and S. Bhandari, “Performance evaluation of class balancing techniques for credit card fraud detection,” in *2017 IEEE International Conference on power, control, signals and instrumentation engineering*, pp. 2747–2752, IEEE, 2017.

[12] J. Ahammad, N. Hossain, and M. S. Alam, “Credit card fraud detection using data pre-processing on imbalanced data-both oversampling and undersampling,” in *Proceedings of the International Conference on Computing Advancements*, pp. 1–4, 2020.

[13] G. K. Kulatilleke, “Challenges and complexities in machine learning based credit card fraud detection,” *arXiv preprint arXiv:2208.10943*, 2022.

[14] J. Rodríguez-Ruiz, J. I. Mata-Sánchez, R. Monroy, O. Loyola-González, and A. López-Cuevas, “A one-class classification approach for bot detection on twitter,” *Computers & Security*, vol. 91, p. 101715, 2020.

[15] F. Alegre, A. Amehraye, and N. Evans, “A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns,” in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems*, pp. 1–8, 2013.

[16] J. Ying, Y. Feng, Q. A. Chen, and Z. M. Mao, “Gps spoofing attack detection on intersection movement assist using one-class classification,” in *ISOC Symposium on Vehicle Security and Privacy*, 2023.

[17] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani, “Multi-class classification on riemannian manifolds for video surveillance,” in *Computer Vision – ECCV 2010* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), (Berlin, Heidelberg), pp. 378–391, Springer Berlin Heidelberg, 2010.

[18] H. J. Shin, D.-H. Eom, and S.-S. Kim, “One-class support vector machines—an application in machine fault detection and classification,” *Computers & Industrial Engineering*, vol. 48, no. 2, pp. 395–408, 2005.

[19] S. Kilickaya, M. Ahishali, F. Sohrab, T. Ince, and M. Gabbouj, “Hyperspectral image analysis with subspace learning-based one-class classification,” in *2023 Photonics & Electromagnetics Research Symposium*, pp. 953–959, 2023.

[20] A. Degerli, F. Sohrab, S. Kiranyaz, and M. Gabbouj, “Early myocardial infarction detection with one-class classification over multi-view echocardiography,” in *2022 Computing in Cardiology*, vol. 498, pp. 1–4, 2022.

[21] A. Senf, X.-w. Chen, and A. Zhang, “Comparison of one-class svm and two-class svm for fold recognition,” in *Neural Information Processing: 13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006. Proceedings, Part II 13*, pp. 140–149, Springer, 2006.

[22] D. M. Tax and R. P. Duin, “Support vector data description,” *Machine Learning*, vol. 54, pp. 45–66, 1 2004.

[23] F. Sohrab, J. Raitoharju, A. Iosifidis, and M. Gabbouj, “Ellipsoidal subspace support vector data description,” *IEEE Access*, vol. 8, pp. 122013–122025, 2020.

[24] F. Sohrab, J. Raitoharju, M. Gabbouj, and A. Iosifidis, “Subspace support vector data description,” in *2018 24th International Conference on Pattern Recognition*, pp. 722–727, IEEE, 2018.

[25] F. Sohrab, A. Iosifidis, M. Gabbouj, and J. Raitoharju, “Graph-embedded subspace support vector data description,” *Pattern Recognition*, vol. 133, p. 108999, 2023.

[26] N. Kwak, “Nonlinear projection trick in kernel methods: An alternative to the kernel trick,” *IEEE transactions on neural networks and learning systems*, vol. 24, no. 12, pp. 2113–2119, 2013.

[27] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, “Graph embedded one-class classifiers for media data classification,” *Pattern Recognition*, vol. 60, pp. 585–595, 2016.

[28] F. Sohrab, J. Raitoharju, A. Iosifidis, and M. Gabbouj, “Multimodal subspace support vector data description,” *Pattern Recognition*, vol. 110, p. 107648, 2021.