Scientific Research Publishing

Applied Mathematics, 2020, 11, 1275-1291
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Credit Card Fraud Detection Using Weighted Support Vector Machine

Dongfang Zhang, Basu Bhandari, Dennis Black

Model Development Department, Comerica Bank, Dallas, USA
Email: DZhang@comerica.com

How to cite this paper: Zhang, D.F., Bhandari, B. and Black, D. (2020) Credit Card Fraud Detection Using Weighted Support Vector Machine. *Applied Mathematics*, **11**, 1275-1291.
https://doi.org/10.4236/am.2020.1112087

Received: November 20, 2020
Accepted: December 18, 2020
Published: December 21, 2020

Abstract

Credit card fraudulent data is highly imbalanced, and it has presented an overwhelmingly large portion of nonfraudulent transactions and a small portion of fraudulent transactions. The measures used to judge the veracity of the detection algorithms become critical to the deployment of a model that accurately scores fraudulent transactions taking into account case imbalance, and the cost of identifying a case as genuine when, in fact, the case is a fraudulent transaction. In this paper, a new criterion to judge classification algorithms, which considers the cost of misclassification, is proposed, and several undersampling techniques are compared by this new criterion. At the same time, a weighted support vector machine (SVM) algorithm considering the financial cost of misclassification is introduced, proving to be more practical for credit card fraud detection than traditional methodologies. This weighted SVM uses transaction balances as weights for fraudulent transactions, and a uniformed weight for nonfraudulent transactions. The results show this strategy greatly improve performance of credit card fraud detection.

Keywords

Support Vector Machine, Binary Classification, Imbalanced Data, Undersampling, Credit Card Fraud

1. Introduction

Credit card use is popular in the United States, and with continued and stable growth. The opportunity for fraudulent credit card transactions will also increase as documented in 2019 by the Federal Trade Commission's Consumer Sentinel Network Databook January 2020 ([1], p 11) which reports 53,763 cases of credit card fraud with an associated cost of $135 mm. The situation is not improving with continued reporting of customer information breaches, even prestigious Ca-

pitalOne reporting on Monday September 23, 2019, that data breaches occurring from 2005 through 2019 compromised the personal information of 100 mm+ customers, and individuals applying for credit cards, also according to Federal Trade Commission, credit card fraud has increase 72.4% in 2019 compared with 2018 [1]. Because of the large reported credit card fraud dollar amounts, the importance of credit card fraud detection has invoked increased interest in industry and academia.

Researchers have developed machine learning algorithms to predict credit card fraud, although this research has progressed well, it is still challenging in the following areas: how to preprocess imbalanced data, choosing a criterion to judge the performance of different algorithms, and finding an efficient and effective algorithm.

Real-world credit card data set transactions are highly imbalanced with many non-fraudulent records versus only a few fraudulent records, and for the data set used in this study, illicit transactions account for only 0.17% of all transactions.

Note however, misclassification of a fraudulent transaction is far more serious than misclassification of a non-fraudulent transaction, since, misclassification of a fraudulent transaction as a non-fraud transaction will result in a financial loss for the bank, but misclassification of a non-fraudulent transaction as fraud the bank only needs to send a verification message to the customer. Misclassification of different fraudulent transactions also costs the bank differently, and in the data set used in this paper, the minimum fraudulent transaction is $0.10, and the maximum fraud transaction is $25,691.00, so misclassification of the maximum transaction costs much more than the minimum for fraud transactions.

Most industry practitioners, and academics, use accuracy, precision-recall, or area under curve (AUC) to measure the performance of a classifier, which does not reflect the financial cost of individual misclassifications, however in this paper, researchers will include a new measure to overcome this flaw.

Because of the enormous size of credit card transaction data sets, training on a data set without any modification by using some type of sophisticated machine learning models, will use a large amount of computational resources adding to the computational time, and therefore is impractical and inefficient. The imbalanced characteristics of the data set will also cause problems for training, which skew to the prevailing class, and as long as classifying all cases as the prevailing class can achieve a very high, but useless accuracy rate, the need to accommodate the imbalance is critical.

Usually, undersampling is employed by taking all the positive cases and a sample (possibly of equal size, maybe some percentage greater) of negative cases, and preprocessing the data as needed before applying machine learning models to classify the data set. To help with the computational processing, an online SVM algorithm can also be used to reduce training time when SVM model is investigated. This online SVM (LASVM), automatically only chooses the most informative data, instead of using all the available data, in every iteration [2].

Researchers have found, and logic indicates, that if a data set is highly imbalanced, classification accuracy using standard SVM or logistic regression will be biased toward the prevailing class, however, this bias can be corrected by introducing a weight in the regularization process associated with the loss function, and this weight, helps to alleviate the model classification bias produced by data set imbalance. Assigning different weights to different classes in the regularization process will help ameliorate the imbalance bias, and it has been found that for SVM, if the weights for two classes are the inversion of their data sizes, this will achieve unbiased accuracy for both classes [3].

Although this modification could help improving prediction accuracy of minority class, in our case, fraudulent transaction, further improvement can be achieved by assigning different weights to different individual data points. By assigning large weights to the fraudulent transactions with higher loss, the weight assignment can guarantee that those points will have a higher chance to be correctly detected by the model, and achieving this outcome is the one of the main goals of this study.

The rest of the paper is arranged as follows: Section 2 will introduce related work in this field, including evaluation measurements for imbalanced data classification, preprocessing techniques, and model development for credit card fraud detection. Section 3 will establish the theoretical framework, including developing a weighted SVM considering the financial cost for individual instances, and a new evaluation measurement specifically designed for credit card fraud detection is introduced, and in Section 4, we will be conducting experimental computations, and the last section we will give the final conclusion of this study.

2. Literature Review

2.1. Classification Measurement

Broadly speaking, credit card fraud detection belongs to the category of imbalanced data classification, and research achievements within the area of imbalanced data classification can be automatically applied to credit card fraud detection. In any data classification problem, the most fundamentally important issue is choosing a valid measure to precisely and accurately classify data, and in the imbalance data framework case, its significance is doubled.

Besides the famous, and widely used, measures of accuracy, AUC score, and precision-recall, researchers also developed and investigated other measurements, such as G-mean, discriminant power, and likelihood ratio [4] [5] [6]. But there was no clear winner that the author's recommended from these measurements, although accuracy is apparently misleading, which every author agrees.

The idea of adjusting methodologies using a weighting scheme to account for imbalance and cost, as well as adjusting the evaluation metrics, is not well researched, although [7] created a metric, *wtdAcc*, and examines the following weighting schema: $wtdAcc = w \times Sensitivity + (1-w) \times Specificity$. the authors take, w = 0.7 to indicate higher weights for accuracy on the fraud cases, this framework

is extremely ad-hoc, and not linked to the formal modeling framework.

[8], use a schema that 1) weights the sample, and 2) updates a model weight taking cost into view.

> *converting sample-dependent costs into sample weights, are also known as cost-sensitive learning by example weighting. The weighted training samples are then applied to standard learning algorithms. This approach is at the data level without changing the underlying learning algorithms.*

For updates to the model weight taking cost into view [8] indicates:

> *the tree-building strategies are adapted to minimize the misclassification costs.*

Using these methodologies, [8] employs traditional unadjusted evaluation measures to determine a model's performance for each of the above two procedures potentially biasing the decision process.

The approach in this paper not only updates weights with respect to cost or sample, but presents a new evaluation metric to account for the "cost", in the form of balance, enhancing model selection, minimizing costs, and ameliorating, in part, data imbalance.

2.2. Resampling Techniques

Most standard machine learning techniques cannot handle data sets with highly skewed data distribution. The accuracy will highly bias toward to prevailing class. In the case of credit card fraud detection, only 0.2% of all transaction are illicit, which will predict accuracy for minority class poorly. To overcome the problems introduced by data imbalance, resampling techniques are often applied to the original data set to adjust imbalance and to create unbiased prediction.

Random undersampling or oversampling are simple, which can help solve the problem of data skewness, but often introduce non-informative or ill-informative sub-structures in data set. [9] investigated different resampling techniques and concluded K-Medoids technique based undersampling can achieve best overall result using AUC score as the evaluation measurement. With the popularity of deep learning, [10] used generative adversarial network to oversample the minority class, and achieved good result using recall as the evaluation measurement. [11] investigated different resampling techniques, and concluded oversampling technique, SMOTE + ENN, can achieve best performance using recall as the evaluation measurement and logistic regression as the model. The shortcomings of resampling include possible information loss and extra computing cost. Besides resampling, a method of active learning was introduced by [2], which will choose only a small portion of data from the data set at every training iteration.

2.3. Supervised Machine Learning Models

There are varieties of choice for the purpose of classification of credit card fraud. Logistic regression (LR), SVM, and random forest (RF) are the three most fre-

quently chosen classification techniques used in many different applications. [7] compared these three techniques, using a real-world data set of credit card transaction. They found logistic regression can achieve comparative performance with other two more sophisticated models, of which no parameter was tuned and optimized.

AdaBoost has also been a good choice because it is within the algorithm that assigning weights to different classes can be achieved, which will help to predict minority class more accurately. [12] used an AdaBoost as black-box model for credit card fraud considering financial loss of misclassification and achieved comparable results with start-of-art commercial system. [8] developed new weights updating strategies of AdaBoost, which assigned higher weights to the misclassified instances of minority class.

Deep learning techniques, such as convolutional neural network or recurrent neural network, have great success in computer vision and language processes, which need very sophisticated algorithms to distinguish different features, however, in the area of credit card fraud detection, these methods have not had such success to date. [13] used long short-term memory (LSTM) for credit card fraud detection, considering the characteristics of time series in the data set, and found the LSTM model did not improve the performance detection dramatically compared with RF, and finally, the authors suggested an ensemble model combining these two methods, LSTM and RF, could achieve better results than using only one of them as the classifier.

2.4. Unsupervised Machine Learning Models

Besides supervised classification algorithms, unsupervised learning algorithms can also be employed for the purpose of fraud detection, as [14] specifically mention in their data mining work, the use of K-mean Clustering could be used to implement a fraud detection algorithm, and [15], implemented a combination of PCA and Simple K-mean Clustering, in the WEKA machine learning environment [16], to obtain an optimized combination of dimension reduction and clustering achieving 100% precision on a generated credit card transaction data set. [17] [18] also compared supervised and unsupervised learning algorithms on fraud detection, and found using unsupervised learning algorithms is more difficult, and performance is worse, than using supervised learning algorithms.

3. Theory

3.1. Support Vector Machine

The principle ideas surrounding the support vector machine started with [19], where the authors express neural activity as an all-or-nothing (binary) event that can be mathematically modeled using propositional logic, and which, as ([20], p. 244) succinctly describe is a model of a neuron as a binary threshold device in discrete time.

Thus for binary classification, when two classes can be completely separated

the classification problem is characterized as considering a training data set $\{x_1, y_1\}, \{x_2, y_2\}, \cdots, \{x_n, y_n\}$, in which x_i is a vector of d dimensions, and y is a scale $\in \{+1, -1\}$. Therefore, y is a label of the data belonging to one class or the other class, and assuming linear separability, a straightforward algorithm finds a hyperplane which is linear combination of x_i that separates the two classes. If we know the linear separator, $y = w \cdot \Phi(x_i) + b$, in which Φ is called feature function specified by hand, w and b are parameters determined by the learning algorithm on training data. The criteria for deciding a data point belongs to a specific class is:

$$y_i \cdot \left(w \cdot \Phi(x_i) + b\right) \geq 1$$

Rosenblatt in 1962 described this algorithm with the perceptron, as mentioned in ([21], p. 192), and produced in [22], with a mechanism to discover a hyperplane which can separate two classes with maximum margins between two categories. The margin is defined as the distance from nearest points from both classes to the separating hyperplane, and these nearest points are called support vectors and are only a small fraction of all data. The perceptron methodology assumes that the two classes are completely separable. Equation (1) can be used to solve w, b assuming the hyperplane achieve maximum margins between the two categories.

$$\begin{gathered} \min_{w,b} \frac{1}{2}\|w\|^2 \\ s.t. y_i \cdot \left(w \cdot \Phi(x_i) + b\right) \geq 1, \ i = 1, 2, \cdots, l \end{gathered} \tag{1}$$

In the real world, the classes cannot always be separated clearly. Sometimes, there are some points on the wrong side of the hyperplane, that is a separable hyperplane may not exist ([23], p. 343), and to deal with classification error, a soft margin is introduced, which allows some data to be classified on the wrong side of the hyperplane.

$$\begin{gathered} \min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ s.t. y_i \cdot \left(w \cdot \Phi(x_i) + b\right) \geq 1 - \xi_i, \ i = 1, 2, \cdots, l \\ \xi_i \geq 0, \ i = 1, 2, \cdots, l \end{gathered} \tag{2}$$

As shown in Equation (2), a second term, x_i, is introduced, which is used to handle the cases of misclassification. The user-specified parameter C is weight for the cost of misclassification. Setting a large C gives high penalty for misclassification, and a small C gives low penalty for misclassification. As shown in **Figure 1**, only data points beyond the right side of margin space has no penalty. Equation (2) is called primal equation, which is also a constrained optimization problem.

Above primal problem, Equation (2), can also be transformed to a dual problem, Equation (3), according to Lagrange duality where α_i is the Lagrange multiplier.

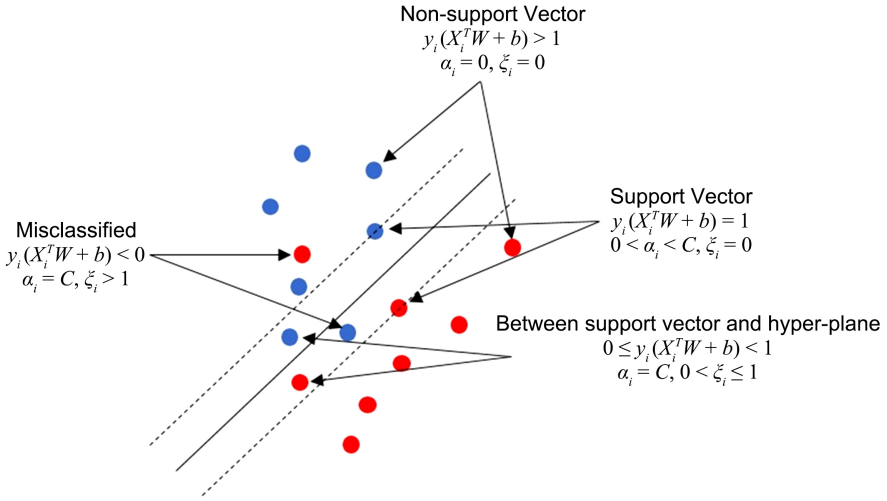

Figure 1. Illustration of soft margin of SVM.

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K\left\langle x_i, y_j\right\rangle - \sum_{i=1}^{l}\alpha_i$$

$$s.t.\sum_{i=1}^{l}\alpha_i y_i = 0$$

$$0 \le \alpha_i \le C,\ i = 1,2,\cdots,l \tag{3}$$

Equation (3) will naturally introduce the kernel function to SVM, which is the most powerful characteristics of SVM. Kernel function, $K\left\langle x_i, y_j\right\rangle$, can be applied to transform the linear hyperplane to nonlinear hypersurface. It also maps low dimension features to high dimension features, for some cases, infinite dimensions, without explicitly building high dimension features, which can circumvent the curse of dimensionality. The only drawback of SVM is the computation is at least quadratic to data size, which make SVM hard to train on large data set [24]. In this paper, different undersampling techniques will be used to trim the training data set.

3.2. Weighted Support Vector Machine

In the standard SVM methodology, a weight for the penalty of misclassification is the same for every datapoint, nevertheless, a weighted SVM can be further distinguished from the uniform weight paradigm, if the penalty for an individual datapoint is different for different transactions according to the potential for financial loss. Introducing another model parameter, S_i, representing the weighted financial loss of each misclassified datapoint, the weighted SVM can then augmented and improved, and be written as exhibited in Equation (4) below:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} S_i\xi_i$$

$$s.t. y_i \cdot \left(w \cdot \Phi\left(x_i\right) + b\right) \ge 1 - \xi_i,\ i = 1,2,\cdots,l$$

$$\xi_i \ge 0,\ i = 1,2,\cdots,l \tag{4}$$

In this paper, weights will be introduced to the loss function to reflect the financial importance between each transaction. The weights for legitimate transactions, S_{nf}, are assigned same. The weights for fraudulent transactions are assigned proportional to the amount of money transferred. To solve the weighted SVM optimization problem, lagrange multiplier can also be introduced, the resulted dual equations is similar to that of standard SVM, Equation (3). The only difference is that the constraint of $0 \le \alpha_i \le C$ is changed to $0 \le \alpha_i \le CS_i$.

4. Result

To align this paper with other papers in the literature, fraudulent transactions will be labeled positive cases, and non-fraudulent transactions will be labeled negative cases, and again following the literature, false negative cases (type II errors) are cases which are fraud but are classified as non-fraud cases, and false positive cases (type I errors) are cases which are non-fraud, but are classified as fraud. The accuracy is defined as Equation (5). The definition of precision and recall will follow the accepted conventions, see Equation (6) and Equation (7).

The contribution of this paper is the introduction of a new measure, which we call financial recovery, and is defined as the portion of total detected monetized fraudulent transactions in Equation (8) divided by all monetized fraudulent transactions. The new financial recovery measure developed in this paper is a much more practical measure than other measures for this application, since the objective of detecting fraudulent transactions is to minimize financial loss for a firm or financial institution.

$$\text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{total samples}} \tag{5}$$

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{6}$$

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{7}$$

$$\text{financial recovery} = \frac{\text{amount of transactions in tue positive}}{\text{amount of transactions of all positive}} \tag{8}$$

The data set used in this paper was downloaded from Kaggle.com, an online data science platform with publicly available data, and the records downloaded include credit card transactions from a European bank over a two-day period in September 2013. Each record in the data set includes 30 features, which are all derived principal components from a set of original features, except for the first feature which is time, and the last feature which is monetized card transaction amount, both these features are native to the original data set.

The total number of credit card transactions exhibited in the database is 284,807 with a scarcity of fraudulent transactions presenting, only 492, which is a mere 0.17% of all records, and as shown in **Figure 2**, most transaction amounts were lower than $100, but it could be more than $10,000.

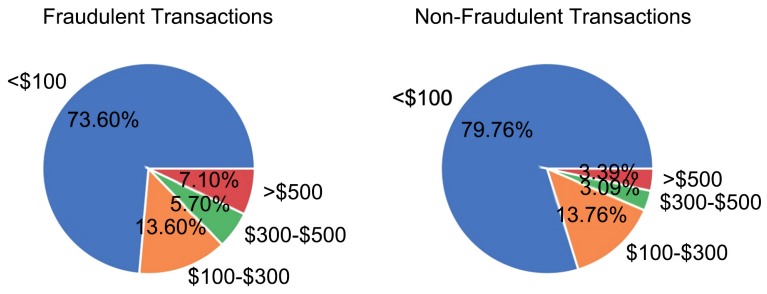

Figure 2. Distribution of transaction amounts.

This classification problem will be carried as following:

- Three commonly used algorithms, LR, SVM, and RF, will be applied to the data. The results will be used as the benchmark for the more advanced algorithms.
- The weighted SVM will be applied to the data. Three undersampling techniques are compared and sampling size will be optimized.
- The weights of nonfraudulent classes will be optimized while individual weights of fraudulent data points are assigned as the transaction balance.
- The impact of kernel functions, such as sigmoid, polynomial, and radial basis function (RBF), on the performance of SVM algorithm will be investigated.

Python is the language for programming with the sci-kit learn package used for all the aforementioned algorithms, and data preprocessing was carried out with preprocessing function, StandardScaler, to put all the data on the same scale, which is very important for SVM, and finally, to keep the results consistent and repeatable across algorithms, the random-state seed was set when calling functions from the Sci-kit learn package.

4.1. Benchmark Results

The benchmark classification process consisted of using the three most used algorithms, Linear Regression (LR), Support Vector Machines (SVM), and Random Forests (RF) under the benchmark rubric of no parameters tuned, and all the parameters using the default settings.

The results in Table 1 shows that the accuracy of all three algorithms is 99.9%, not unexpected as there are only 0.17% fraud transactions in the data, but this metric gives no differentiation capacity to assess which methodology best classifies fraud. For the precision metric, the methodology with higher precision, the less non-fraudulent transactions will be classified as fraudulent transactions which will give customers a better shopping experience with less verification messages or calls being be made, and the precision of SVM presents at 89.5% a little better than LR and RF.

The recall measure provides the percentage of fraudulent transactions found in the data, and the recall of SVM presents at 61.6% which is 2.9% less than LR and 0.7% more than RF giving mixed results for SVM giving no clear methodological winner.

Table 1. Results of benchmark models.

Model	Accuracy	Precision	Recall	Financial recovery
LR	0.999	0.856	0.645	0.428
SVM	0.999	0.895	0.616	0.475
RF	0.999	0.875	0.609	0.379

The very important, newly introduced performance metric is financial recovery, which represents the percentage of the amount of fraudulent transactions that have been correctly detected compared to sum of all transaction amounts. The financial recovery score for SVM is 47.5%, the highest compared with the other two algorithms presented, and is of primary importance since financial recovery is of paramount importance to the company.

Overall, the above results indicate the SVM model maintains the most consistency among the three algorithms presented, and therefore will be used as a benchmark model for further evaluation and investigation in the remainder of the paper.

4.2. Comparison of Undersampling Techniques

The computational burden of SVM may extend running time beyond practical limits as [25] indicate below:

> *Training an SVM requires solving a constrained quadratic programming problem, which usually takes* $O(m^3)$ *computations where m is the number of examples. Predicting a new example involves* $O(sv)$ *computations where* sv *is the number of support vectors and is usually proportional to m. Consequently, SVMs' training time and prediction time to a lesser extent on a very large data set can be quite long, thus making it impractical for some real-world applications.*

And as Kramer indicates above, runtime for the credit card data set considered here with 284,807 records could be extensive.

To accelerate the computational efficiency of SVM, random, nearmiss, and k-nearest neighbors (KNN) undersampling techniques were employed where the fraudulent transactions were kept untouched and the nonfraudulent transactions were undersampled. The random technology, as [11] on page 2 of his work indicates below, is a reasonable undersampling methodology:

> *A simple undersampling technique is uniformly random undersampling of the majority class. This can potentially lead to loss of information about the majority class. However, in cases where each example of the majority class is near other examples of the same class, this method might yield good results.*

[11] opines on the nearmiss technology on page 2 as shown below:

In NearMiss-1, those points from L (majority class) are retained whose mean distance to the k nearest points in S (minority class) is lowest, where k is a tunable hyperparameter.

[11] also discusses KNN, which is characterized in More's work as CNN, described on page 4 as follows:

In CNN undersampling, the goal is to choose a subset U of the training set T such that for every point in T its nearest neighbor in U is of the same class.

First, below these algorithms are discussed in relation to measures of accuracy, precision, recall, and the new measure introduced here, financial recovery, and second, the most important parameter for random undersampling, sample size is considered. Note, in relation to sample size, the sample size of random undersampling sets non-fraudulent samples at 10 times the number of fraudulent samples, and nearmiss, as well as KNN, undersampling techniques used default values.

The results in **Table 2** shows that random undersampling is the best technique for the data considered here, with these three algorithms achieving similar results for accuracy, precision, recall, and financial recovery, however, random undersampling can achieve 35.7% precision far superior than the other two undersampling techniques.

At the same time, using random undersampling, the total samples were reduced from 199,364 to 3894, and compared to standard SVM, the training time using random undersampling was reduced from more than two hours to less than 1 minute using a PC with intel core i7 CPU and 32 GB memory. Compared with the SVM benchmark model, random undersampling, not only dramatically improves the computation efficiency, but also increase financial recovery from 47.5% to 84.6%, although precision was reduced from 89.5% to 35.7% as shown in **Table 1** and **Table 2**.

Then the ratio of samples of nonfraudulent to fraudulent transactions for SVM with the random undersampling technique was varied to find the optimized ratio, and **Figure 3** shows the model performs well when the ratio is15. Note, when the ratio is higher than 15, financial recovery does not improve and training with SVM becomes slower, and when the ratio is lower than 15, precision deteriorates. Consequently, random sampling with a ratio 15 will be used in the weighted SVM algorithm.

Table 2. SVM with different undersampling techniques.

Undersampling	Accuracy	Precision	Recall	Financial recovery
Random	0.997	0.357	0.862	0.846
Nearmiss	0.975	0.054	0.899	0.847
KNN	0.973	0.053	0.935	0.890

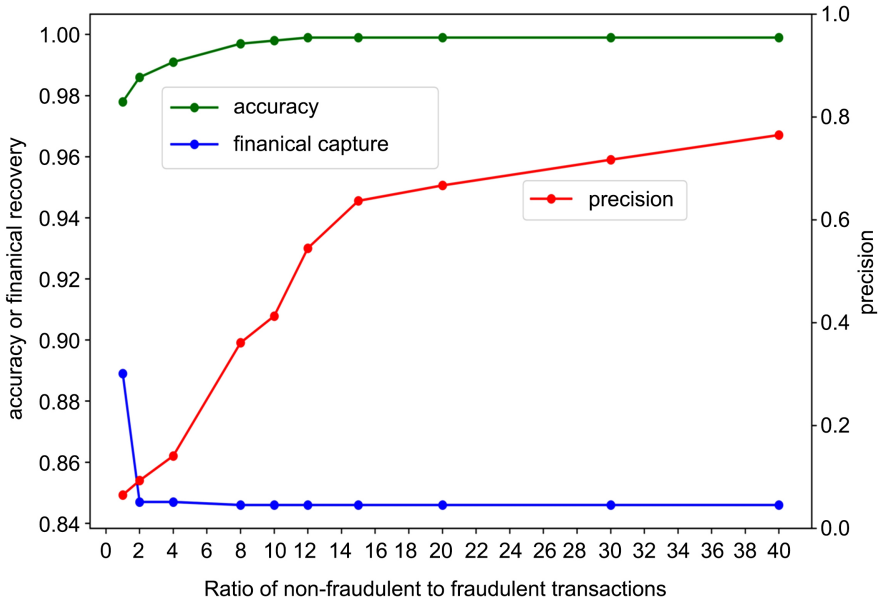

Figure 3. Distribution of amount of transactions.

4.3. Optimization of Weighted Linear SVM

Usually, weights are assigned exactly the same, *i.e.* identical, for each class member [3], however, in practice, different fraudulent transactions have different costs since a fraudulent transaction worth $10,000 is much more important than that worth $100. In this paper, the weights assigned to the fraudulent transactions in the training step, pertain to the dollar amount of transactions scaled by total dollar amount of all transactions, and unlike the fraudulent class, the weights assigned to all nonfraudulent transactions are same. The logic behind this decision is that the cost of misclassification of a nonfraudulent as fraudulent transaction is the same for each nonfraudulent record, that is, the cost of the misclassification is the cost of investigating the record transaction which is similar for each misclassified record. After assigning the transaction amount as the weight to each according fraudulent transaction, the optimized weight of all nonfraudulent transactions was investigated, and in **Table 3**, the weight of the nonfraudulent transaction is in the first column, labeled as S_{nf}. It can be found that with the increased weight in the S_{nf}, column accuracy increased. The best performance happened at a weight of 10 with the financial recovery at 99.6%, accuracy at 97.6%, precision at 5.8%, and note financial recovery decreases significantly when the weight is more than 10. Precision increases when the nonfraudulent weight increases, and because higher precision and higher financial recovery is better, the balance of these two performance measures merits setting the weight at 10 for nonfraudulent transactions.

4.4. Optimization of Weighted Nonlinear SVM

In all the above results, SVM employed a linear kernel, and to further improve the performance of the weighted SVM, three nonlinear kernel functions were

investigated: Radial Basis Function (RBF), polynomial, and sigmoid which will enhance performance in the face of diverse data structures [26].

Table 4 presents the kernel functions investigated in this study. The most important parameter for kernels above is the gamma, and the polynomial kernel has a degree parameter. Besides these kernel parameters, optimized weights for nonfraudulent transactions, S_{nf}, also needs to be found. A grid search approach is used to find the best parameters for these three kernel functions. The three kernels researched here, and reported on in this study, provide extensions of the linear kernel examined above, $K\left\langle x_i, x_{i'}\right\rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$, and the results are presented in Table 5.

Table 3. Weighted SVM with random undersampling technique.

S_{nf}	Accuracy	Precision	Recall	Financial recovery
0.1	0.723	0.006	0.964	0.998
1	0.893	0.013	0.949	0.997
4	0.954	0.032	0.928	0.996
10	0.976	0.058	0.913	0.996
20	0.987	0.098	0.884	0.953
30	0.992	0.144	0.862	0.846
40	0.994	0.199	0.855	0.846
80	0.998	0.417	0.841	0.846

Table 4. Weighted SVM with random undersampling technique.

Kernel name	Kernel functions [26] [27]
Random	$\exp\left(-\gamma\left\|x, x'\right\|^2\right)$
Nearmiss	$\left(\gamma\left\langle x, x'\right\rangle + r\right)^d$
KNN	$\tanh\left(\gamma\left\langle x, x'\right\rangle + r\right)$

Table 5. Weighted kernel SVM with undersampling technique.

Kernel	$S_{nonfraud}$	γ	Degree	r	Accuracy	Precision	Financial recovery
Linear	10	n/a	n/a	n/a	0.976	0.058	0.996
RBF	0.1	0.05	n/a	n/a	0.937	0.024	0.997
Polynomial	0.5	0.01	2	0.1	0.938	0.024	0.997
Sigmoid	0.8	0.01	n/a	1e−5	0.738	0.006	0.997

The best weight for the RBF kernel nonfraudulent class is 0.1, with best gamma reported at 0.05, and the best weight for the polynomial kernel nonfraudulent class reported at 0.5 with polynomial of degree of 2 for the kernel function. The optimized weight for nonfraudulent class using sigmoid function is 0.8, with best gamma at 0.01. It can also be found that using kernel functions did not significantly improve the performance of classification. This demonstrates that the most important factor to improve SVM algorithms is to use right weights for individual data point.

- Recall improves materially from the linear case which overpowers the drop in accuracy since financial recovery increases with the kernel functions.
- By using linear kernel, about 2% unfraudulent will be classified as fraudulent. It increases to 6% using RBF kernel or polynomial, and more than 20% using sigmoid kernel.
- The drop in precision, which results in more unfraudulent as fraudulent, cannot adjust minor increase of financial recovery.

The linear kernel performance is superior to more complex kernels in the face of optimal weighting of the nonfraudulent cases thus satisfying the desirable statistical property of parsimonious modeling.

4.5. Confusion Matrix

To evaluate the performance of our model, the confusion matrix results of a weighted SVM model and standard SVM are compared in **Figure 4**. In this confusion matrix with a cost function [28], we assume that TN and TP have no cost since both of them are classified correctly. The cost of FN is considered as the balance of transaction since the balance will be lost when it is classified incorrectly. The cost of FP is also considered as 0 since only a verification message or email is sent from bank to clients. From the **Figure 4**, we can see that the financial cost of using a standard SVM is $10,396. The financial cost will be reduced to $90 when using the weighted SVM with undersampling techniques. This is only two days of transaction data of a European bank, the annually saving amount will be a great benefit for the bank.

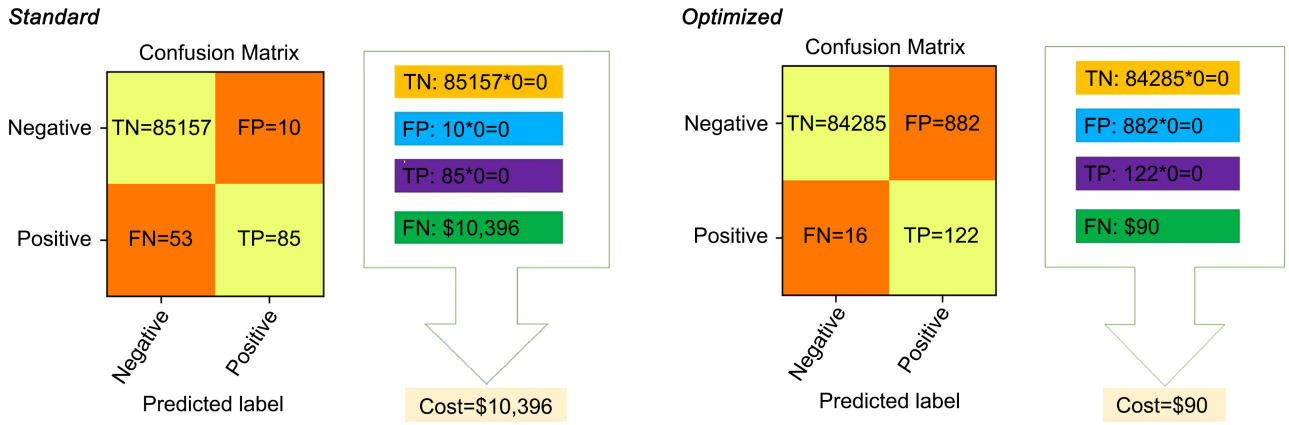

Figure 4. Financial cost comparison of standard SVM and optimized weighted SVM.

5. Conclusion

A new criterion, financial recovery, is created to judge the performance of classification algorithms based on financial lost. A weighted SVM model with random undersampling methodology, using the amount of transaction, as a weight for fraudulent data points is developed and applied to records of credit card transactions in a bank in Europe which occurred over a two-day period. The result shows that using the new criteria and novel weight scheme can greatly improve the performance of credit card fraud detection. Most importantly, this strategy will minimize the financial loss of bank in the aspect of credit card fraud.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Consumer Sentinel Network Data Book 2019 (2020). https://www.ftc.gov/system/files/documents/reports/consumer-sentinel-network-data-book-2019/consumer_sentinel_network_data_book_2019.pdf

[2] Ertekin, S., Huang, J., Bottou, L. and Giles, L. (2007) Learning on the Border: Active Learning in Imbalanced Data Classification. *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, Lisbon, 6-10 November 2007, 127-136. https://doi.org/10.1145/1321440.1321461

[3] Huang, Y.-M. and Du, S.-X. (2005) Weighted Support Vector Machine for Classification with Uneven Training Class Sizes. 2005 *International Conference on Machine Learning and Cybernetics*, Guangzhou, Vol. 7, 4365-4369. https://doi.org/10.1109/ICMLC.2005.1527706

[4] Bekkar, M., Djemaa, H.K. and Alitouche, T.A. (2009) Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, **3**, 27-38.

[5] Gu, Q., Zhu, L. and Cai, Z.H. (2009) Evaluation Measures of the Classification Performance of Imbalanced Data Sets. 2009 *International Symposium on Intelligence Computation and Applications*, Huangshi, 23-25 October 2009, 461-471. https://doi.org/10.1007/978-3-642-04962-0_53

[6] Hossin, M. and Sulaiman, M. (2015) A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, **5**, 1-11. https://doi.org/10.5121/ijdkp.2015.5201

[7] Bhattacharyya, S., Jha, S., Tharakunnel, K. and Westland, J.C. (2001) Data Mining for Credit Card Fraud: A Comparative Study. *Decision Support Systems*, **50**, 602-613. https://doi.org/10.1016/j.dss.2010.08.008

[8] Sun, Y.M., Kamel, M.S., Wong, A.K.C. and Wang, Y. (2007) Cost-Sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognition*, **40**, 3358-3378. https://doi.org/10.1016/j.patcog.2007.04.009

[9] Dubey, R., Zhou, J.Y., Wang, Y.L., Thompson, P.M., Ye, J.P. and Alzheimer's Disease Neuroimaging Initiative and Others (2014) Analysis of Sampling Techniques for Imbalanced Data: Ann = 648 ADNI Study. *ACM SIGMOD Record*, **87**, 220-241. https://doi.org/10.1016/j.neuroimage.2013.10.005

[10] Fiore, U., De Santis, A., Perla, F., Zanetti, P. and Palmieri, F. (2019) Using Generative Adversarial Net-Works for Improving Classification Effectiveness in Credit Card Fraud Detection. *Information Sciences*, **479**, 448-455. https://doi.org/10.1016/j.ins.2017.12.030

[11] More, A. (2016) Survey of Resampling Techniques for Improving Classification Performance in Unbalanced Datasets.

[12] Chan, P.K., Fan, W., Prodromidis, A.L. and Stolfo, S.J. (1999) Distributed Data Mining in Credit Card Fraud Detection. *IEEE Intelligent Systems and Their Applications*, **14**, 67-74. https://doi.org/10.1109/5254.809570

[13] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L.Y. and Caelen, O. (2018) Sequence Classification for Credit Card Fraud Detection. *Expert Systems with Applications*, **100**, 234-245. https://doi.org/10.1016/j.eswa.2018.01.037

[14] Berry, M.J.A. and Linoff, G.S. (2004) Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. John Wiley & Sons, Hoboken.

[15] Lepoivre, M.R., Avanzini, C.O., Bignon, G., Legendre, L. and Piwele, A.K. (2016) Credit Card Fraud Detection with Unsupervised Algorithms. *Journal of Advances in Information Technology*, **7**, 34-38. https://doi.org/10.12720/jait.7.1.34-38

[16] Witten, I.H. and Frank, E. (2002) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. *ACM SIGMOD Record*, **31**, 76-77. https://doi.org/10.1145/507338.507355

[17] Hilas, C.S. and Mastorocostas, P.A. (2008) An Application of Supervised and Unsupervised Learning Approaches to Telecommunications Fraud Detection. *Knowledge-Based Systems*, **21**, 721-726. https://doi.org/10.1016/j.knosys.2008.03.026

[18] Niu, X.T., Wang, L. and Yang, X.L. (2019) A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised.

[19] McCulloch, W.S. and Pitts, W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, **5**, 115-133. https://doi.org/10.1007/BF02478259

[20] Venables, W.N. and Ripley, B.D. (2013) Modern Applied Statistics with S-PLUS. Springer Science & Business Media, Berlin.

[21] Bishop, C.M. (2006) Pattern Recognition and Machine Learning. Springer, Berlin.

[22] Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992) A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, 27-29 July 1992, 144-152. https://doi.org/10.1145/130385.130401

[23] Gareth, J., Daniela, W., Trevor, H. and Robert, T. (2013) An Introduction to Statistical Learning: With Applications in R. Springer, Berlin. https://doi.org/10.1007/978-1-4614-7138-7

[24] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. https://doi.org/10.1007/BF00994018

[25] Kramer, K.A., Hall, L.O., Goldgof, D.B., Remsen, A. and Luo, T. (2009) Fast Support Vector Machines for Continuous Data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* (*Cybernetics*), Vol. 65, 989-1001. https://doi.org/10.1109/TSMCB.2008.2011645

[26] Achirul Nanda, M., Boro Seminar, K., Nandika, D. and Maddu, A. (2018) A Comparison Study of Kernel Functions in the Support Vector Machine and Its Application for Termite Detection. *Information*, **9**, 1-14.

https://doi.org/10.3390/info9010005

[27] Vapnik, V. (2013) The Nature of Statistical Learning Theory. Springer Science & Business Media, Berlin.

[28] Shmueli, G., Bruce, P.C., Yahav, I., Patel, N.R. and Lichtendahl Jr., K.C. (2017) Data Mining for Business Analytics: Concepts, Techniques, and Applications in R. John Wiley & Sons, Berlin.