KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

FRAUD DETECTION ON THE TIME SERIES DATA WITH MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

MASTER THESIS

Eman ABDULKARIM

JULY 2022
TRABZON

KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

FRAUD DETECTION ON THE TIME SERIES DATA WITH MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

Eman ABDULKARIM

This thesis is accepted to give the degree of
"MASTER OF SCIENCE"
By
The Graduate School of Natural and Applied Sciences at
Karadeniz Technical University

The Date of Submission : 18/06/2022
The Date of Examination : 25/07/2022

Supervisor : Prof. Dr. Mustafa ULUTAŞ

Trabzon 2022

ACKNOWLEDGMENTS

This thesis is written as the completion of the Master of Department of Computer Engineering, Institute of Sciences, at Karadeniz Technical University. First, I would like to express my gratitude to my respected thesis supervisor Prof. Dr. Mustafa ULUTAŞ, for his cooperation in collaborating with me remotely during the pandemic and for his valuable discussions and feedback. I would also like to thank Prof.Dr. Erdal KILIÇ, Asst. Prof. İbrahim SAVRAN for their support and for being a part of my thesis committee. I also want to thank my family for their sacrifices and support. Finally, thanks to all my friends who supported me during studying and shared lovely memories.

Eman ABDULKARIM
Trabzon, 2022

THESIS STATMENT

This study, titled "Fraud Detection on Time Series Data with Machine Learning and Deep Learning Techniques," which I presented as a master's Thesis, was fully supported by my supervisor Prof. Dr. Mustafa ULUTAŞ. I have completed under his responsibility of him. I have done the experiments in the relevant laboratories, I have fully indicated the information I have obtained from other sources in the text and the bibliography, I have acted by scientific research and ethical rules during the working process, and I have made all kinds of decisions in case the opposite arises. I declare that I accept the legal result. 25/07/2022

Eman ABDULKARIM

TABLE OF CONTENTS

Sayfa No

Yüksek Lisans Tezi

ÖZET

MAKİNE ÖĞRENİMİ VE DERİN ÖĞRENME TEKNİKLERİ İLE ZAMAN SERİSİ VERİLERİ ÜZERİNDE SAHTEKARLIK TESPİTİ

Eman ABDULKARIM

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Lisansüstü Programları
Danışman: Prof. Dr. Mustafa ULUTAŞ
2022, 46 Sayfa

Teknoloji kullanımının artması ve e-ticarete geçiş nedeniyle kredi kartları daha popüler hale geliyor. Dolandırıcılık tespit yöntemlerinin geliştirilmesi, kredi kartı dolandırıcılık oranını azalttı ve kullanıcı tabanı büyüdükçe dolandırıcılık işlemlerinde kaybedilen toplam para miktarı artıyor. Ayrıca, her gün yeni finansal dolandırıcılık yöntemleri tanıtılmakta, bu da onu örüntü tanıma yoluyla araştırma topluluğunun dikkatini gerektiren dinamik ve gelişen bir sorun haline getirmektedir. Bu çalışma, bir dolandırıcılık tespit algoritması tasarlamanın çeşitli zorluklarını ele alan bir kredi kartı dolandırıcılık tespit yaklaşımı sunmaktadır. Hileli işlemlerin sayısı normal işlemlerden çok daha az olduğundan, sınıf dengesizliği önemli bir sorundur. Makul bir örneklem büyüklüğü elde etmek için azınlık sınıfının sentetik aşırı örneklemesini içeren bir yaklaşım seçilmiştir. Buradaki ana katkı, optimum özellikleri elde etmek için filtre tabanlı ve sarmalayıcı tabanlı yöntemlerin çıktılarını birleştiren hibrit bir özellik seçimi yaklaşımıdır. Önerilen bu teknik, diğer üç öznitelik seçme tekniğiyle karşılaştırıldı; dört makine öğrenimi algoritmasında diğer üç teknikleri geride bıraktı ve ardından performansı artırmak için farklı dengeleme teknikleri uyguladı. Azınlık sınıfları artırılarak SMOTE örnekleme yöntemi kullanılarak veri seti üzerinde bir derin öğrenme modeli (LSTM) de eğitildi ve çok yüksek bir sonuç elde edildi. Önerilen model rakiplerinden daha iyi performans gösteriyor.

Anahtar Kelimeler: Dolandırıcılık Tespiti, Makine Öğrenimi, Derin Öğrenim

Master Thesis

SUMMARY

FRAUD DETECTION ON THE TIME SERIES DATA WITH MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

Eman ABDULKARIM

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Programs
Supervisor: Prof. Dr. Mustafa ULUTAŞ
2022, 46 Pages

Credit cards are becoming more popular due to the increased use of technology and a shift toward e-commerce. The development of fraud detection methods has reduced the ratio of credit card frauds, and the total amount of money lost to fraudulent transactions increases as the user base grows. Furthermore, newer financial fraud methods are introduced daily, making it a dynamic and evolving problem that requires the research community's attention through pattern recognition. This study presents a credit card fraud detection approach that addresses several challenges in designing a fraud detection algorithm. Because fraudulent transactions are far fewer in number than regular transactions, the class imbalance is a significant issue. An approach incorporating the synthetic oversampling of the minority class is chosen to obtain a reasonable sample size. The main contribution here is a hybrid feature selection approach, which combines the outputs of the filter-based and wrapper-based methods to get optimal features. This proposed technique was compared to three other feature selection techniques; outperformed all those techniques on the four machine learning algorithms, and then applied different balancing techniques to increase the performance. A deep learning model (LSTM) was also trained on the dataset, using SMOTE sampling method with minority classes increased, achieving a remarkably high result. The proposed model outperforms its competitors.

Key Words: Fraud Detection, Machine-Learning, Deep-Learning

LIST OF FIGURES

Sayfa No

LIST OF TABLE

Sayfa No

1. INTRODUCTION

The advancement of digital technologies such as smartphones and high-speed internet has paved the way for rapid e-commerce proliferation [1]. Fraudulent transactions are associated with these networks, and enterprises and government agencies ensure secure financial transactions [2]. Collaboration among regulators, credit card companies, and banking institutions has improved security by preventing fraud [3]. Regulators have ensured consumer protection by holding credit card providers and banking companies liable for fraud and swindles [3]. The development of fraud detection and prevention technologies is becoming more sophisticated, making it more difficult to steal money [4].

In the e-commerce consumer industry, a fraudulent transaction is referred to as the unauthorized use of a credit card or similar payment method to make a payment without the knowledge or consent of the credit card's owner [4]. Fraudulent credit card transactions can occur in two ways: authorized (where the credit card owner is tricked into making a payment in a fake account) or unauthorized (where the transactions are made without the card holder's authorization). Because they are deployed so that both authorized users and fraudsters use the same e-commerce networks, the risk of fraud is high. Fraud prevention is the first mechanism to secure advanced technologies from fraud to making payments secure. Fraud detection is a viable and most reliable method for fraud prevention [4]. The systems that detect fraud in e-commerce transactions are based on data-driven approaches.

Machine learning modeling is commonly used as a credit card fraud detection method. The trained machine learning model detects fraudulent transactions and generates an alert, resulting in transaction denial or interruption from a system administrator. As it is a sensitive matter, it necessitates reliable and accurate detection to detect as many frauds as possible while avoiding false alarms to legitimate transactions. Due to a range of factors, modeling a machine learning classifier to classify a financial transaction as fraudulent or normal is complex [5]. Preprocessing requires representative data to generate a feature set and train a classification model.

In general, the number of normal financial transactions far outnumbers the number of fraudulent transactions [4, 5]. This problem causes the financial transaction data to be highly skewed, resulting in a class imbalance that impedes the training of a reliable classification model. Because this is a common issue in many machine learning problems, many methods

are proposed in the literature to address it. Oversampling [6, 7] and under-sampling [7] are the two most common approaches to this problem. The most common under-sampling technique is random sample removal from the majority class, but other strategies include Tomek links [8], clustering [9], and ensemble learning [10]. This method is appropriate when the number of samples is significant, and dropping samples from the majority class is not an issue.

On the other hand, oversampling comes in handy when the number of samples in the minority class is insufficient. SMOTE [11], or Synthetic Minority Oversampling, is the most used approach, but ADASYN [12], random oversampling, and augmentation [13] are also used.

The feature set used for model training is the second significant limitation of training these models. Financial transaction data, credit cardholder information, transaction history, and other information may help develop a reliable and accurate prediction model.

Additionally, this sensitive information may not be available to design a credit card fraud detection system, posing a limitation [14]. Some institutions have released a mathematical representation of actual data that can be used for model building while maintaining data confidentiality and data provision to fuel research. Principal components derived from the principal component analysis [15] of actual data are used as features in machine learning model training. However, there is limited flexibility to perform extensive analysis of these features and calculate new or compound features from this feature set. Feature selection is thus a suitable method for selecting the most appropriate set of features to train the model to ensure high accuracy and better generalization.

There are several approaches to feature selection, each with its own benefits and drawbacks. It is not always possible to choose the most appropriate system for a specific problem, so it may be necessary to analyze and select several candidate methods before deciding on the best one. The same is true for classification algorithms, which use different processes to construct decision boundaries, but it is difficult to predict which algorithm will perform better ahead of time.

To overcome these challenges and present a solution to the credit card fraud detection problem using machine learning, we proposed several data processing and modeling strategies to address the class imbalance issue.

Similarly, feature set preprocessing is used to choose the best set of features to maximize the accuracy and generalizability of the trained model. A hybrid strategy is used

to transform features before selecting them. This strategy is designed with the specific needs of the modeling process in mind. Two wrapper-based and one filter-based method are used to perform the feature selection. The obtained feature subsets are combined using set theory's union operation, and the final feature set is obtained for model building.

The modeling process is carried on by training four popular classical machine learning methods to perform supervised learning. The modeling process is carried on by training four popular classical machine learning models to perform supervised learning. These models are trained and then evaluated. The four classification models are compared to our final solution, a deep neural network model. This model includes a long-short-term memory network and classification layers. The obtained accuracy and performance parameters are superior to four classification models based on classical machine learning that have been widely reported in the literature. The deep neural network training and validation results indicate that the model is well trained and generalizable enough to be used for any data with a similar feature set to detect credit card fraud with accuracy and reliability.

1.1. Research Motivation

Due to several factors, detecting credit card fraud is a worthwhile research topic. The apparent reason for requiring a credit card fraud detection system is to prevent potential monetary loss due to fraudulent activity. Card issuers suffer significant financial losses due to fraudulent transactions, avoiding an accurate and dependable fraud detection system. The ratio of fraudulent financial transactions has decreased due to improved fraud detection systems. Still, the total monetary loss has increased due to increased users. Furthermore, newer fraudulent methods are introduced, which are countered by various approaches, such as the introduction of chip cards. However, there is still a need for improved fraud detection systems that are more efficient and reliable. These factors make credit card fraud detection a dynamic and evolving pattern recognition problem requiring the research community's attention to develop better methods.

2. RELATED WORK

The Credit Card Fraud Detection approach falls among those problems that have received extensive attention from the research community due to budgetary impact. This problem has some inherent limitations, some of which are unique and require a carefully crafted framework to mitigate these limitations and solve the problem. One of these limits is the imbalance of data for class distribution. A limited number of financial transactions fall in the category of fraud, and the rest of the transactions are legitimate. Therefore, class imbalance correction is one of these systems' fundamental limitations addressed in all the solutions. Another limitation is the availability of the transaction information, which is not available in sufficient detail due to the involvement of confidential information of an individual, his financial or personal profile, or other such information. Therefore, public datasets contain limited information or information transformed into some other representation such as principal components of PCA or not publicly sharing the dataset. Although this limitation is genuine and cannot be overcome, approaches are designed to model the problem with limited information successfully. Another challenge with this problem is to perform credit card fraud detection in real-time or near-real-time, necessitating efficient implementations.

Moreover, different machine learning algorithms are explored to solve the problem of credit card fraud detection; among them are random forest, k-nearest neighbor, support vector machines, and artificial neural networks. Ensemble learning algorithms such as different boosting classifiers are particularly successful among conventional supervised classification methods. Deep Learning, especially sequence modeling approaches, has recently gained attention, and researchers are increasingly using these approaches for credit card fraud detection anomaly prediction.

Among the most common approaches to credit card fraud detection is supervised learning using conventional machine learning algorithms such as support vector machines, decision trees, naive Bayes, k-nearest neighbors, logistic regression, artificial neural networks, and bagging boosting classifiers. Pradhan et al. [16] have compared the use of all these classifiers and proposed random forest and artificial neural networks for Credit Card Fraud De tection. Similarly, Rathore et al. [17] compared the decision tree, logistic regression, k-nearest neighbor, and random forest-based credit card fraud detection. Gürsoy

et al. [18] also provided the AUC of conventional classification algorithms. Extensive experiments are performed using decision trees, naive Bayes, k-nearest neighbor, logistic regression, support vector machines, artificial neural networks, multilayer perceptron, random forest, and ensemble classification. Hussain et al. [19] presented a similar solution by incorporating decision trees, support vector machines, and random forest on comparative results; random forest classifiers are optimized to improve classification accuracy. Precision, accuracy, and sensitivity are reported for comparison and benchmarking.

To address the limitation of class imbalance, synthetic oversampling of minorities. Conventional machine learning algorithms are trained using this balanced class data to classify financial transactions. Tyagi et al. [20] have followed a similar approach. The class imbalance problem is addressed by performing synthetic oversampling using SMOTE and training support vector machines, decision trees, logistic regression, and random forest to make predictions of credit card fraud.

Uttam et al. [21] used SMOTE and SMOTE-Tomek for class imbalance correction. These two approaches are compared with their random oversampling technique, claiming superior results. Jenipher et al. [22] demonstrated conventional supervised learning algorithms on a class-balanced dataset. They have incorporated SMOTE and PyCaret for class imbalance correction of credit card data and trained five supervised classification algorithms. Random forest trained on class-balanced datasets has superior performance due to its ensemble nature and good generalization.

Krishnan et al. [23] explored various class imbalance correction methods as the nature of the problem necessitates the class balancing of the data. It is not essential to perform detection of as many frauds as possible, but it is equally important not to penalize legitimate transactions. Therefore, the focus of supervised learning is coupled with dataset balancing concerning class distribution, resulting in a solution to the problem. Kumar et al. [24] proposed using feedback and conventional fraud classification. They have used the support vector machine on a Class-balanced dataset for model building and used a feedback mechanism to improve the learning of decision boundaries. Their approach has demonstrated superior performance in credit card fraud detection than the state-of-the-art.

The principal component analysis is a versatile feature transformation and dimensionality reduction method which serves two purposes in Credit Card Fraud Detection research. The first purpose is that it anonymizes the information of a financial transaction. The second purpose is to reduce dimensionality and eliminate redundancy in the features,

thus improving the predictive performance. Singh et al. [25] attempted to eradicate the erroneous misrepresentation by incorporating PCA and supervised learning. Four supervised classification methods, support vector machines, logistic regression, k-nearest neighbors, and random forest are used on the PCA transformed dataset to detect anomalies.

Similarly, Arafath et al. [26] used PCA to conceal sensitive financial and user information and reduce dimensionality. They have used a tweaked fraud detection model to detect fraudulent financial transactions using a RandomizedSearchCV hyper-parameter optimization method. The proposed framework determines the fraudulent transaction based on the prior transaction pattern.

Islam et al. have addressed the problem of credit card fraud detection by incorporating different data-driven approaches to predictive modeling. Five popular supervised learning algorithms are used for modeling, namely k-nearest neighbor, logistic regression, support vector machines, ensemble classifier, random forest, and gradient boosting. Feature preprocessing is performed using PCA for feature transformation and Robust Scalar for feature scaling in categorical or numerical features. Forward feature selection is used to retain only the most relevant (useful) features. The problem of class imbalance correction is addressed by performing random under-sampling. Different classification models are compared based on precision, recall, f1, and classification accuracy on a benchmark dataset.

As discussed at the start of this chapter, the class imbalance is one of the significant problems of modeling a Credit Card Fraud Detection framework. Some authors have used random or under-sampling, and others have used synthetic approaches to balance the number of samples in both classes. Similarly, feature preprocessing is a primary requirement, and most studies have not paid much attention to this issue. Model building is performed using conventional machine learning methods, and little work is undertaken to use deep neural networks. To address these issues, we have proposed a hybrid feature selection method, an efficient class balancing approach, and modeling the problem using deep neural networks with memory layers.

3. PROPOSED MODEL AND RESULTS OVERVIEW

Credit card fraud detection is a significant problem due to its financial impacts. The problem is characterized by various challenges, among which the unavailability of public datasets due to confidentiality issues and class imbalance are major ones. Similarly, expectations of high prediction precision and recall are another major challenge, and the task must be performed in real-time or near real-time. The project aims to improve credit card fraud detection by incorporating data processing strategies followed by conventional machine learning and deep learning-based modeling approaches.

The research objective is to implement the strategy to process the data for supervised learning. The credit card fraud data contains the financial transactions performed by many customers during a specified time. This data includes a vast imbalance as the positive instances are less frequent than negative ones. This imbalance occurs as the fraudulent transactions are rare and increasing their number through synthetic oversampling may not yield satisfactory results. Therefore, most such problems are modeled through majority class under-sampling techniques, but the other approach can also be considered. To solve such a problem, a systematic approach to data processing and modeling is required to cover all the aspects of data modeling.

3.1. Project Requirements

Predictive modeling is the process in data mining and machine learning that performs the tasks of forecasting, detection, and identification, among others. Business research is heavily influenced by modeling through machine learning algorithms, which is valid for finance and other areas. Analysis of financial transactions is performed to identify trends and perform modeling in which decision boundaries are learned from the training data and predictions are made. However, modeling is a straightforward process, but it necessitates preprocessing the data to convert it into a suitable form for regression or classification algorithms. The data preprocessing and other design decisions are interlinked and require a systematic methodology for modeling.

Using a systematic methodology is desirable, and some established approaches for machine learning modeling are desirable. Each of these methodologies has a particular process with advantages and disadvantages, and applying these approaches depends on the domain and specific industry practices. KDD, CRISP-DM, and SEMMA are the most widely used and popular machine learning project development. Approaches this work is based on CRISP-DM, a six-stage project development process.

This six-stage process is a cyclic process in which each subsequent stage depends on one or several previous steps, and the outcome of each stage influences the following stages. Figure 1 depicts the process in which the outer circle indicates the cyclic nature of machine learning project development as the final model deployment influences another cycle of project development based on the obtained outcome. The arrows indicate the typical flow between stages, but the process can iterate or switch between stages to accomplish the intended purpose.

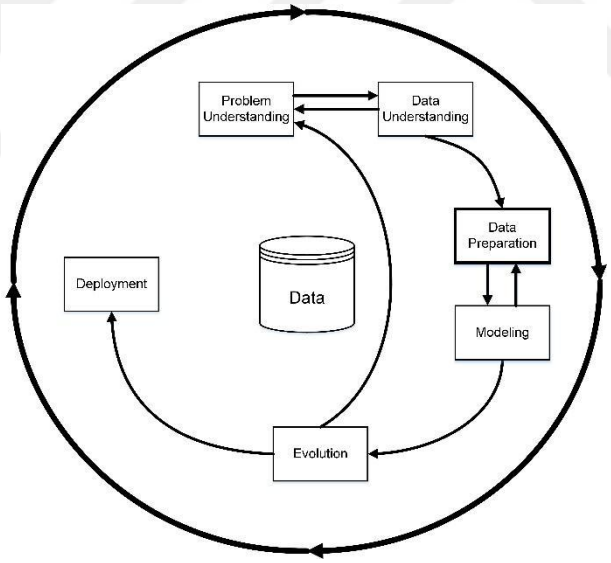


Figure 1. CRISP-DM process for potential customer prediction

3.2. Problem Understanding

As the fundamental step of predictive modeling, problem understanding is the basis of all the design decisions. It is important to understand the objectives of a research project and the expected outcomes, along with the deployment environment and operational requirements. Credit card fraud detection is a critical problem as it involves an economic

loss in case of inaccurate prediction. Moreover, predictions are required in real-time to minimize the delay introduced in e-commerce transaction requests. A machine learning approach is followed in this work, which is a data-centric approach based purely on consumers' online transaction data. It is required to preprocess the data and select suitable features used for model building and validation. The following subsection describes the details of the available features in the data and the preprocessing required for the modeling process.

3.3. Data Understanding

Data understanding is needed to prepare the data preprocessing and model building strategy. Data typically have different attributes; each attribute has some specific type of information. Moreover, in the case of numeric data, the range of the data can vary, and it may be essential to identify the mean and spread of the data. Moreover, the ordinal and categorical data may also be required to be understood to design the steps which may convert it to a suitable form for use in predictive modeling. In this stage, data records are checked through visualization and descriptive statistics, which provide helpful insights and understanding of the dataset.

The dataset contains financial transactions made by European credit cardholders in September 2013. There are 284,807 transactions recorded in two days, and 492 cases are registered as fraudulent transactions. The records are imbalanced for class distribution as 99.83% of records fall in one class (negative) and 0.172% in positive class; this makes it a binary classification problem with class imbalance. The data contains thirty independent variables, and an additional variable is the target variable with either "0" for normal transactions or "1" for fraudulent transactions. There are no missing values in the data.

The data is not provided in raw form due to confidentiality issues and therefore contains numerical input variables for twenty-eight attributes obtained through PCA transformation of the original data. Moreover, background information about credit card fraud is also not provided for some reason. Table 1. provides ten randomly selected records from the dataset, which provides the values and names of different attributes. Attributes from V1, V2, and up to V28 are the principal components obtained from the original data and explain the 99% variance. Two attributes, namely "Time" and "Amount," are provided in their original form without any processing. The attribute "Time" contains the number of

seconds elapsed between the first transaction to each transaction in the dataset. The attribute named "Amount" represents the number of funds transferred in the transaction and can be used for cost-sensitive learning. The target variable is labeled with 0 or 1, corresponding to positive or negative class.

Table 1. Ten randomly selected records of the data

Time	**V1**	**V2**	**V3**	**V4**	**V5**	**...**	**V26**	**V27**	**V28**	**Amount**	**Class**
0	-1.36	-0.07	2.54	1.38	-0.34	...	-0.19	0.13	-0.02	149.62	0
0	1.19	0.27	0.17	0.45	0.06	...	0.13	-0.01	0.01	2.69	0
1	-1.36	-1.34	1.77	0.38	-0.50	...	-0.14	-0.06	-0.06	378.66	0
1	-0.97	-0.19	1.79	-0.86	-0.01	...	-0.22	0.06	0.06	123.5	0
2	-1.16	0.88	1.55	0.40	-0.41	...	0.50	0.22	0.22	69.99	0
172786	-11.88	10.07	-9.83	-2.07	-5.36	...	0.25	0.94	0.82	0.77	0
172787	-0.73	-0.06	2.04	-0.74	0.87	...	-0.40	0.07	-0.05	24.79	0
172788	1.92	-0.30	-3.25	-0.56	2.63	...	-0.09	0.00	-0.03	67.88	0
172788	-0.24	0.53	0.70	0.69	-0.38	...	0.55	0.11	0.10	10	0
172792	-0.53	-0.19	0.70	-0.51	-0.01	...	-0.82	0.00	0.01	217	0

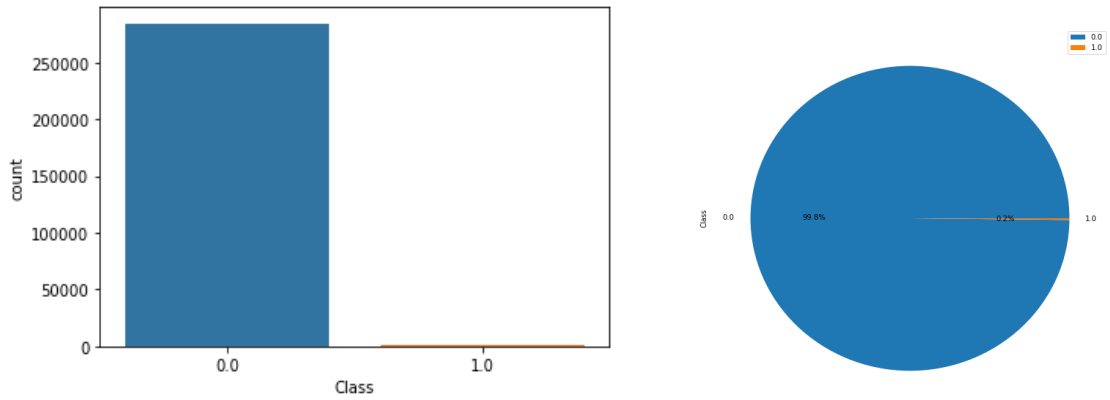
Table 2. Descriptive statistics of the dataset

Statistics	Time	V1	V2	V3	V4	V5	...	V26	V27	V28	Amount
count	284807	284807	284807	284807	284807	284807	...	284807	284807	284807	284807
mean	94813.86	3.92E-15	5.62E-16	-8.77E-15	2.77E-15	-1.55E-15	...	1.70E-15	-3.67E-16	-1.23E-16	88.34962
std	47488.15	1.96E+00	1.65E+00	1.52E+00	1.42E+00	1.38E+00	...	4.82E-01	4.04E-01	3.30E-01	250.1201
min	0	-5.64E+01	-7.27E+01	-4.83E+01	-5.68E+00	-1.14E+02	...	-2.60E+00	-2.26E+01	-1.54E+01	0
25%	54201.5	-9.20E-01	-5.99E-01	-8.90E-01	-8.49E-01	-6.92E-01	...	-3.27E-01	-7.08E-02	-5.30E-02	5.6
50%	84692	1.81E-02	6.55E-02	1.80E-01	-1.98E-02	-5.43E-02	...	-5.21E-02	1.34E-03	1.12E-02	22
75%	139320.5	1.32E+00	8.04E-01	1.03E+00	7.43E-01	6.12E-01	...	2.41E-01	9.10E-02	7.83E-02	77.165
max	172792	2.45E+00	2.21E+01	9.38E+00	1.69E+01	3.48E+01	...	3.52E+00	3.16E+01	3.38E+01	25691.16

Descriptive statistics is crucial to obtain statistical information about each attribute in a tabulated form. Table 2. provides descriptive statistics of the data in which the first row provides the count of values in each attribute which are 284,807 for each of them as there are no missing values in the dataset. The second and third rows provide the mean and standard deviation of each of these values, which are important to understand the spread of data in each attribute. The minimum, maximum, 25^{th}, $50^{th,}$ and 75^{th} quantities are provided in the rest of the rows, which are again helpful in understanding the range and spread of the data. This information may be beneficial in gaining a basic understanding of the data, and several types of visualizations can be obtained for further clarification of any of these statistics. Class distribution of the data is performed to check the number of samples falling in each class. This information is helpful in checking if the dataset is balanced with respect to class distribution, which is a common assumption of most supervised learning algorithms.

Moreover, it also allows us to check the number of classes and if it is possible to merge some classes or perform hierarchical classification. In the case of a dataset with non-uniform class distribution, some undersampling or oversampling techniques can be incorporated to balance the samples in each class. Figure 2. provides the class distribution of samples in the form of a bar chart of the number of samples (Figure 2. (a)) and the percentage distribution of samples in each class (Figure 2. (b)).

The distribution of numeric features helps gain some insights about the data attributes. Histogram analysis is one such technique that falls under univariate analysis as it provides descriptions or visualization of individual attributes only. The histogram is a commonly used method that plots the count of instances on y-axes and the values on x-axes. Therefore, the histogram plots are a useful way to plot the distribution of attribute values. Figure 3. provides the histogram plots of thirty variables, among which the first one is the time variable, and the last one is the transaction variable. Input features are calculated are obtained through principal component analysis. It is apparent from the figures that many of these variables are centered at zero and follow a Gaussian distribution. The only difference between the distributions of these variables among each other is their spread.



(a) Count of samples in each class (b) Percentage of samples in each class

Figure 2. Class distribution of samples

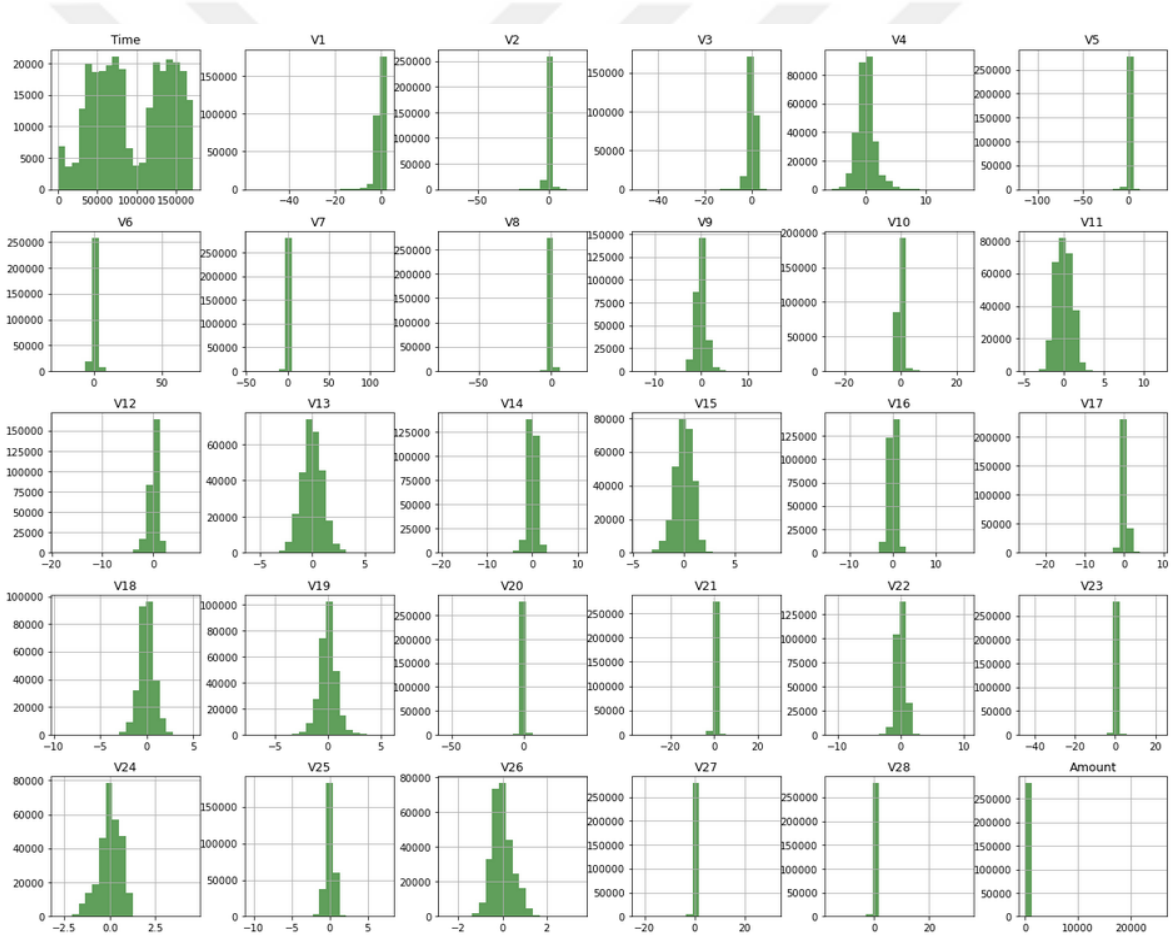


Figure 3. Histogram of thirty independent variables

Correlation analysis is a bivariate analysis, unlike earlier ones. In its broadest sense, correlation provides a degree of the linear relation between two variables. Pearson correlation coefficient analysis is one such correlation that measures the relationship

between two continuous variables. It is the most used method to measure the relationship between two variables, and it calculates this relationship through covariance. It is a linear correlation, and its values vary from -1 to 1. A value of 1 indicates an unrealistically perfect positive relationship in which the value of one variable increases proportionally to the other. A value of -1 is an example of a negative relationship in which the values are inversely related. Whereas a value of zero indicates no relationship between the two variables. Figure 4. provides the Pearson correlation coefficient analysis of all independent attributes and the dependent variable. The correlation of these variables is significantly less with each other except for variables indicating transaction amount and class label. Due to many attributes, the correlation values are not readable from but the color bar can be used to gauge the range of correlation values for each feature. This low correlation determined by the nature of attributes obtained through principal component analysis, and the variables are transformed such that the correlation is minimal to each other. The amount variable indicates the transaction amount and is not transformed and therefore may have a positive or negative correlation with few attributes. Similarly, the target variable "class label" has a higher correlation with few attributes, indicating these attributes' predictability.

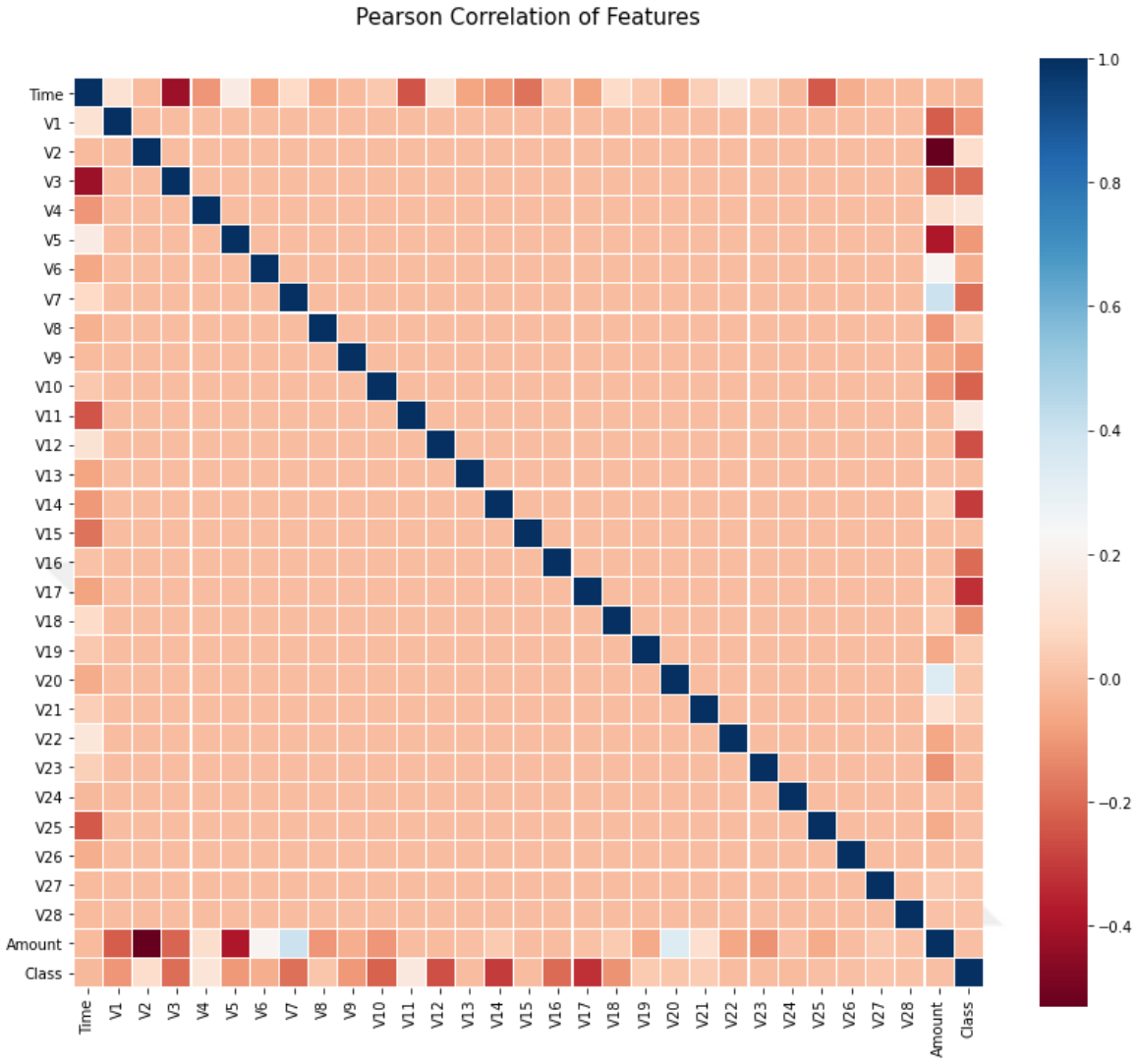


Figure 4. Pearson correlation of attributes with each other

3.4. Data Preparation

The data preparation process is highly dependent on the type of supervised learning algorithm for which it is prepared and may require a few iterations before obtaining a ready-to-use dataset. The data understanding section has indicated that the attributes are preprocessed and are provided in the form of principal component analysis outcome. These attributes have a low correlation and are perfect for model building. Moreover, two of the attributes are provided in an unprocessed form and therefore may require some preprocessing. Furthermore, the distribution of data for class type is highly imbalanced, and thus the use of a suitable class imbalance correction strategy is necessary.

Removal of duplicate values is the first step to data preparation. These values add little to no value to model training and may cause degradation in the performance of the trained model. The duplicate value matching indicated 1081 duplicate values in the data, and therefore these records are dropped from the data and the time attribute are also dropped because it contains truly little useful information. Table 3. summarizes the number of samples before and after duplicate removal and the number of duplicate samples.

Table 3. Duplicate detection and removal

Type of Action	**Samples**
Number of samples in the original dataset	284807
Number of duplicate samples	1081
Number of samples after duplicate removal	283726

3.4.1. Data Transformation

Data transformation is performed through different strategies, and each one of them has its advantage. One such data transformation objective is rescaling the attributes done in several ways. This rescaling of the attribute values results in changes in their range or mean in such a way to make them better for modeling and representation. The histogram analysis and descriptive statistics have indicated that most attributes have zero mean; the range varies considerably. This difference in the range of attributes can cause concern for some predictive modeling algorithms such as distance-based classifiers. This concern is that the difference in range of these attributes may cause a skew in distance calculation. Due to its usefulness, logistic regression, support vector machines, and artificial neural networks use some form of feature scaling due to its effectiveness in terms of performance. Gradient descent-based methods have demonstrated the usefulness of faster convergence speed [27].

Similarly, support vector calculation time is significantly reduced when the attributes are scaled through feature standardization methods [28]. The Amount column has not been transformed by PCA and scaled. Because it is not distributed normally and there are noticeable outliers in the feature, the Robust Scaler is used for handling the outliers scale Amount. The below-mentioned formula is used to perform feature scaling:

$$\frac{xi - Q1(x)}{Q3(x) - Q1(x)} \quad (1)$$

Here, x(i) is the scaled value whereas the Q1 and Q3 represent 25% and 75% quartiles.

3.5. Feature Selection

Feature selection is the process of selecting a subset of attributes that are most relevant to the prediction of target variables. There are several ways to select such a feature subset, and there are three broad categories to feature selection. Filter-based feature selection methods are most superficial and straightforward and are most used. These feature selection methods work based on some statistical analysis of the features and are independent of the prediction algorithm. The advantage of such feature selection methods is that the choice of a predictive modeling algorithm is not required beforehand, and the selected feature set can be used to compare the performance of several candidates' predictive modeling algorithms. Correlation-based measures are the most used methods among other statistical methods to perform feature selection through filter-based methods. The selection of the features is achieved by removing the least informative or redundant features, which provides faster modeling and better generalization.

On the other hand, Wrapper-based methods are based on some prediction algorithm. The process of wrapper-based feature subset selection is based on feature subset selection. These strategies work by adding or removing features from the feature subset iteratively and performing model evaluation until the most representative feature subset is selected. This feature subset selection method provides the highest predictive performance but has two inherent limitations. The first limitation is the computational complexity which increases exponentially with candidate attributes. The second limitation is the associated risk of over-fitting, which is especially prevalent in the case of smaller numbers of instances in the dataset.

The third approach to feature selection is a trade-off between the above two approaches, which tries to combine the advantages and overcome the limitations. Wrapper-based methods perform feature selection during the learning process. They, therefore, prove to be advantageous to both filter-based and wrapper-based methods as they combine the best of both approaches.

Due to the added advantage of this approach, we have opted for a similar strategy that combines two filter-based and wrapper-based methods to obtain an optimal feature set through the hybrid method. Both filter-based and wrapper-based methods perform feature selection, and the hybrid feature set is obtained by taking the union of both feature sets. The proposed hybrid feature selection approach performs feature selection through filter-based and wrapper-based methods and then combines the features through union operation to obtain the hybrid feature set. Wrapper-based methods alone are superior to filter-based methods in their predictive performance. Still, as we must experiment with several candidate models before the selection of a final model, it may not yield an optimal feature set. Therefore, a hybrid feature set is obtained, a better candidate feature set, and helps identify the most suitable prediction algorithm.

3.5.1. Feature Selection Using L1-norm

Sklearn of python provides Select from Model, a wrapper-based feature selection algorithm. L1-norm-based features are selected using Support Vector Machines (SVM) as a classifier model. SVM with the linear kernel is trained with a sparsity ratio of 0.01 and L1-penalty assigned to the features that violated and qualified the threshold retained through the selection process. The feature selection through this process has resulted in a feature set provided in Table 4.

3.5.2. Feature Selection Using Decision Trees

This approach is also a wrapper-based feature selection approach that uses decision trees to perform feature selection. Decision trees are very versatile classifiers and work very well even when the training dataset size is minimal, and therefore there are very few chances of overfitting. Feature importance measure in this approach is based on an impurity measure which removes the less useful features. The feature set obtained through this approach is provided in Table 5.

3.5.3. Filter-based Feature Selection

SelectKBest, a function provided by Python through Sklearn, is used to perform filter-based feature selection. In this method, features are selected based on their contribution to the prediction of the class labels. The SelectKBest is provided with an attribute "f_classif," an ANOVA (Analysis of Variance) based feature selection method that works with categorical target variables. Categorical target variables are less sensitive to chi-square-based statistics, and therefore the selection of predictors is performed using this attribute. Table 6. provides the feature set selected using the filter-based feature selection method.

3.5.4. Hybrid Feature Set

The features selected from the two of the above methods are wrapper based, and one of them is filter-based. A hybrid feature set is constructed through union operation which combines the features selected from any of the three methods and provides a feature set of twenty-four features out of thirty features. Table 4., Table 5., and Table 6 provides the feature set selected using each individual feature selection approach. The first approach has provided a feature set with twenty features followed by the second approach providing a feature set of five features and the third approach has provided a feature set with fifteen features. The union operation intended to combine these individual feature sets has resulted in twenty-four features which are provided in the form of a final feature set array. The process of hybrid feature selection is demonstrated in Figure 5.

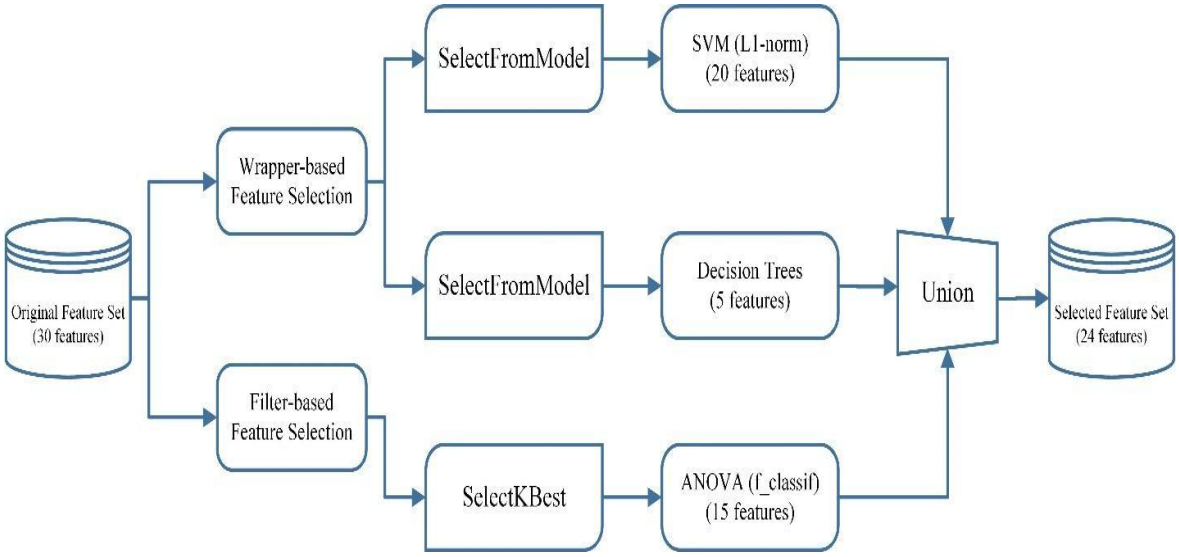


Figure 5. Hybrid feature selection approach

The vector of final features is provided below:

```
array([[-1.35980713e+00, -7.27811733e-02,  2.53634674e+00, ...,
         1.28539358e-01,  1.49620000e+02,  1.33558377e-01],
       [ 1.19185711e+00,  2.66150712e-01,  1.66480113e-01, ...,
         1.67170404e-01,  2.69000000e+00, -8.98309914e-03],
       [-1.35835406e+00, -1.34016307e+00,  1.77320934e+00, ...,
        -3.27641834e-01,  3.78660000e+02, -5.53527940e-02],
       ...,
       [ 1.91956501e+00, -3.01253846e-01, -3.24963981e+00, ...,
         2.65745453e-01,  6.78800000e+01,  4.45477214e-03],
       [-2.40440050e-01,  5.30482513e-01,  7.02510230e-01, ...,
        -5.69158864e-01,  1.00000000e+01,  1.08820735e-01],
       [-5.33412522e-01, -1.89733337e-01,  7.03337367e-01, ...,
        -4.73648704e-01,  2.17000000e+02, -2.41530880e-03]])
```

Table 4. Features selected through SVM (L1-based) wrapper method

	V1	V2	V4	V7	V8	V9	V10	V11	V12	V13	V14	V16	V17	V18	V19	V20	V21	V23	V25	Amount
0	-1.36	-0.07	1.38	0.24	0.10	0.36	0.09	-0.55	-0.62	-0.99	-0.31	-0.47	0.21	0.03	0.40	0.25	-0.02	-0.11	0.13	149.62
1	1.19	0.27	0.45	-0.08	0.09	-0.26	-0.17	1.61	1.07	0.49	-0.14	0.46	-0.11	-0.18	-0.15	-0.07	-0.23	0.10	0.17	2.69
2	-1.36	-1.34	0.38	0.79	0.25	-1.51	0.21	0.62	0.07	0.72	-0.17	-2.89	1.11	-0.12	-2.26	0.52	0.25	0.91	-0.33	378.66
3	-0.97	-0.19	-0.86	0.24	0.38	-1.39	-0.05	-0.23	0.18	0.51	-0.29	-1.06	-0.68	1.97	-1.23	-0.21	-0.11	-0.19	0.65	123.50

Table 5. Feature selection through decision trees-based wrapper method

	V10	V12	V14	V17	V27
0	0.090794	-0.6178	-0.31117	0.207971	0.133558
1	-0.16697	1.065235	-0.14377	-0.11481	-0.00898
2	0.207643	0.066084	-0.16595	1.109969	-0.05535
3	-0.05495	0.178228	-0.28792	-0.68409	0.062723
4	0.753074	0.538196	-1.11967	-0.23703	0.219422

Table 6. Feature selection through filter-based (f_classif) method

	V1	V2	V3	V4	V5	V6	V7	V9	V10	V11	V12	V14	V16	V17	V18
0	-1.360	-0.073	2.536	1.378	-0.338	0.462	0.240	0.364	0.091	-0.552	-0.618	-0.311	-0.470	0.208	0.026
1	1.192	0.266	0.166	0.448	0.060	-0.082	-0.079	-0.255	-0.167	1.613	1.065	-0.144	0.464	-0.115	-0.183
2	-1.358	-1.340	1.773	0.380	-0.503	1.800	0.791	-1.515	0.208	0.625	0.066	-0.166	-2.890	1.110	-0.121
3	-0.966	-0.185	1.793	-0.863	-0.010	1.247	0.238	-1.387	-0.055	-0.226	0.178	-0.288	-1.060	-0.684	1.966
4	-1.158	0.878	1.549	0.403	-0.407	0.096	0.593	0.818	0.753	-0.823	0.538	-1.120	-0.451	-0.237	-0.038

3.6. Data Balancing

Class imbalance represents that the number of samples in each class is different. This issue causes concerns in model building and introduces a bias in the model. The number of samples in each class is made equal by using a data balancing technique to overcome these issues. Two broad approaches are used: the first work by undersampling in the class with majority samples and the second work oversampling the class with minority samples.

Amputation is the simplest way of data balancing in which samples are removed from the majority class through random selection, but it is rarely used. This approach is only used when there are many training examples and removal of the desired number of samples does not affect the model's training. Minority class oversampling, on the other hand, introduces samples in the class with a smaller number of samples and therefore is considered a preferred approach in many cases.

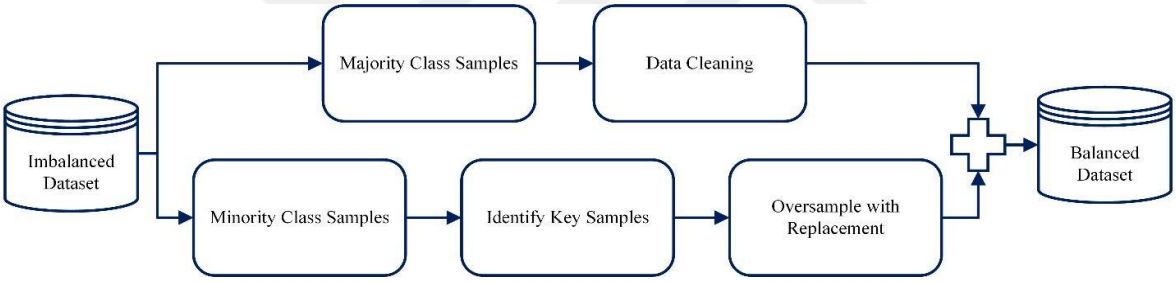


Figure 6. Class balancing through Synthetic minority class oversampling

3.6.1. Synthetic Minority Class Oversampling Technique (SMOTE)

This works by increasing the number of samples in the minority class. The oversampling process works by selecting minority classes and, after identifying critical samples performing synthetic samples generation with replacement. The samples are generated using the k-nearest neighbor algorithm, which selects the K number of neighbors and creates a new sample between the selected samples. SMOTE can be represented mathematically as:

$$X' = X + rand(0,1) \times |X - X_k| \quad (2)$$

Where X′ denotes the newly generated synthetic data, X denotes the original data, X_k denotes the data's kth attribute, and rand denotes a random number between 0 and 1.

3.6.2. Borderline-SMOTE SVM

The primary distinction between SVM-SMOTE and other SMOTE is that instead of employing K-nearest neighbors to detect misclassification like in Borderline-SMOTE, SVM-SMOTE would use the SVM algorithm. After training SVMs on the original training set, the boundary area is approximated by the support vectors in the SVM-SMOTE. Synthetic data will be generated at random along the lines that connect each minority class support vector to a few of its closest neighbors. The data is synthesized away from the class overlap zone. It is primarily concerned with how the data is split.

3.6.3. Borderline-SMOTE

A version of the SMOTE is the Borderline-SMOTE. It has something to do with the border, as the name suggests. As a result, unlike SMOTE, which generate synthetic data at random across the two classes, Borderline-SMOTE only generates synthetic data along the decision boundary between the two classes**.** Borderline-SMOTE is also divided into two types: Borderline-SMOTE1 and Borderline-SMOTE2. The distinctions are simple: Borderline-SMOTE1 additionally oversampled the majority class where the majority data are generating decision boundary misclassification, whereas Borderline-SMOTE2 simply oversampled the minority classes. When we know that misclassification occurs frequently around the boundary decision, Borderline-SMOTE is the best option.

3.6.4. Adaptive Synthetic Sampling (ADASYN)

ADASYN is a synthetic data generation algorithm that uses a weighted distribution to generate synthetic data for different minority class examples based on their learning difficulty, with more synthetic data generated for minority class examples that are more difficult to learn compared to minority class examples that are easier to learn. This can be expressed mathematically as:

$$G = (m_l - m_s) \times \beta \quad (3)$$

Where G denotes the total number of synthetic data instances required for the minority class, ml denotes the minority class, m_s denotes the majority class, and is used to compute the optimal balancing level between 0 and 1.

3.6.5. NearMiss-3

Near Miss refers to a class of undersampling algorithms that select samples depending on the distance between instances from the majority and minority classes. This approach has three variations: NearMiss-1, NearMiss-2, and NearMiss-3. NearMiss-3 picks a given number of majority class examples for each closest example in the minority class. The NearMiss-3 is the most preferred of the three types since it only keeps majority class samples that are on the decision boundary.

3.6.6. Tomek Links for Undersampling

Tomek links are pairs of opposite-class instances that are their own closest neighbors. To put it another way, they are close-knit pairs of opposing instances. Tomek's algorithm searches for such pairs and eliminates the bulk of them. The goal is to draw a clearer boundary between the minority and majority classes, allowing the minority region(s) to stand out more.

3.6.7. Edited Nearest Neighbor (ENN)

The ENN method works by determining each observation's K-nearest neighbor and then deciding whether the majority class from the observation's k-nearest neighbor is the same as the observation's class. If the majority class of the observation's K-nearest neighbor and the observation's class vary, the observation and its K-nearest neighbor are eliminated from the dataset. The following is an explanation of the ENN algorithm.

a. Given an N-observation dataset, calculate K, the number of nearest neighbors. If K cannot be calculated, it is assumed to be three.
b. From the dataset's other observations, find the observation's K-nearest neighbor, and then return the majority class from the K-nearest neighbor.
c. If the observation's class and the majority class of its K-nearest neighbor are not the same, the observation and its K-nearest neighbor are eliminated from the dataset.
d. Repeat steps b and c until each class has the desired student ratio.

Because ENN removes the observation and its K-nearest neighbor when the observation's class and the majority class from the observation's K-nearest neighbor differ, it is more powerful than Tomek Links. As a result, ENN is designed to sanitize data more thoroughly than Tomek Links.

3.6.8. One-Sided Selection (OSS)

(OSS) combines Tomek Links with the Condensed Nearest Neighbor (CNN) Rule to create an undersampling strategy. Tomek Links are uncertain locations on the class boundary that the majority class identifies and removes. After that, the CNN method is used to filter out redundant samples from the majority class that is far from the decision border. The algorithm can be implemented as:

a. Let S be the training set
b. Initially, C contains positive examples from S, and only one negative example which is randomly selected from
c. S is classified with 1-NN using the examples in C and compared with assigning the concept labels with the original. All the misclassified examples are moved to C that is consistent with the S while being smaller
d. Remove all negative examples from C which are involved with the Tomek links, which thereby removes those negative examples that are on the borderline or noisy. All the positive ones remain, the resulting set is the T

3.6.9. Neighborhood Cleaning Rule for Undersampling

When sampling the data sets, the Neighborhood Cleaning Rule (NCL) separates the majority and minority samples. NCL employs ENN to eliminate most cases. It discovers three closest neighbors for each instance in the training set. The chosen instance is deleted if it belongs to the majority class and the classification given by its three closest neighbors is the opposite of the class of the chosen instance. If the chosen instance belongs to the minority class and its three closest neighbors misclassify it, the majority neighbors are eliminated.

3.6.10. Random Undersampling (RUS)

This method seeks to select and exclude samples from the majority class at random, reducing the number of examples from the majority class in the updated data. Substantial amounts of data may be lost due to random under-sampling. This is troublesome because data loss makes learning the decision border between minority and majority occurrences more difficult, resulting in poor classification performance. Undersampling results in a data set with fewer examples in the majority class; this method can be repeated until all classes have the same number of examples. This strategy is useful when the minority class has enough instances despite the large imbalance. On the other hand, because we have no way of recognizing or conserving the cases in the majority class that are information rich, it is always critical to consider the prospect of valuable information being lost as we exclude them from our data collection at random.

3.7. Modeling

Model building is the process of selecting a suitable predictive modeling algorithm, providing it with the processed data and training parameters, performing a model evaluation, and selecting the final model based on the model's skill. There is no straightforward way to select a suitable classifier, but statistics obtained in data understanding may help identify a few candidate algorithms. The final model can be selected only by training and evaluating the algorithm on the dataset.

Four conventional classification algorithms are initially selected for model building and based on model evaluation; an ensemble of two best-performing models is constructed. Moreover, a deep neural network-based model is also designed, trained, and evaluated for mode validation. The classification algorithms used for model building are described briefly in the following paragraphs.

3.7.1. Random Forest

It is an ensemble algorithm that uses decision trees as its building block. Random forest [29] is robust and works without requiring much hyper-parameter tuning due to its inherent randomness. It is a popular algorithm due to its applicability in many applications [30], and it can perform very well on various regression and classification problems. As a random forest is based on decision trees, it uses bootstrapped sampling to train these decision trees. The output is combined through an aggregation method, making it an ensemble classifier.

3.7.2. Logistic Regression

Is a fundamental and robust classifier based on a statistical framework and uses a logistic function as its foundation. In multiple descriptive variables, logistic regression generates a rare ratio. It is like multiple linear regression except for a binomial response variable. The basic idea is to combine all variables by eliminating confusion [32]. Logistic regression is useful for categorical data and performs binary classification. In contrast to linear regression, it is an unbounded binary algorithm with only two outputs as zero or one. It allows simultaneous interpretations of varied factors while suppressing confounding factors; it is beneficial for conducting epidemiological research or spam classification.

3.7.3. Naive Bayes

Naive Bayes is the category of classification algorithms based on Bayes decision theory. It is a probabilistic classification algorithm and works by predicting the likelihood of occurrence of a sample in each class. The term "Naive" refers to the algorithm assumes that the attributes are independent of each other [33]. Although this assumption is not correct for

most cases, it helps accelerate the probability computation process [34]. We have used Gaussian naive Bayes in this implementation as it is the most common distribution in natural phenomena. The histogram analysis has shown that all the attributes are normally distributed. Moreover, naive Bayes is considered the most basic and widely used algorithm derived from Bayes decision theory, and significant improvement in their performance is achieved using kernel density estimation [35].

3.7.4. XGBoost Algorithm

Extreme Gradient Boosting is a decision tree for regression and classification problems that is based on a Machine Learning method. This algorithm generates decision trees with each successive tree attempting to reduce the previous tree's faults. Each tree learns from its predecessor and updates the leftover mistakes. As a result, the following tree in the sequence continues to learn from the prior tress's mistakes. In the XGBoost algorithm, the base learners are referred to as 'weak learners,' because their predictive power is just slightly better than random guessing and their bias is strong. Each of the weak learners gives vital information for estimation, allowing the boosting strategy to combine the weak learners to create a strong learner. Both bias and variation are reduced by the last good learner. Furthermore, in contrast to Random Forest, where trees are fully cultivated, boosting enables for the utilization of trees with less breaks.

3.7.5. Deep Neural Networks

Deep neural networks are a class of artificial neural networks in which input passes through several hidden layers until the output is obtained at the output. The models are trained through gradient descent-based optimizers which perform iterative adjustment of layer weights until convergence or the maximum epochs. In this work, a sequence modeling-based architecture that incorporates long short-term memory (LSTM) based architecture is used. The input is reshaped into 3D format. The resulting is treated as a dense layer having real values instead of binary. What KRRAS wants from us is to reshape the data such that the prediction at time step *t* is computed based on the specified number of previous and the current time step features. LSTM is insensitive to gap length and makes them more beneficial

than other sequence modeling methods such as RNNs, hidden Markov models, etc., in many applications. The LSTM layers perform modeling and sequence learning. The dense layer performs the reduction of outputs (demonstrated in Figure 7.) and provides a flatten layer. It takes the input from the LSTM layer, performs calculations using activation functions, and provides an output. Dropout layers randomly drop some neurons in each iteration, which helps reduce the possibility of over-fitting. The concept of dropout is demonstrated in Figure 8. A model trained with dropout is observed to perform better during model validation. An activation function is a node in a neural network that is attached at the terminal point of a neuron. The outcome of the activation function is provided as the input to the next layer. The concept of activation function is demonstrated in Figure 9.

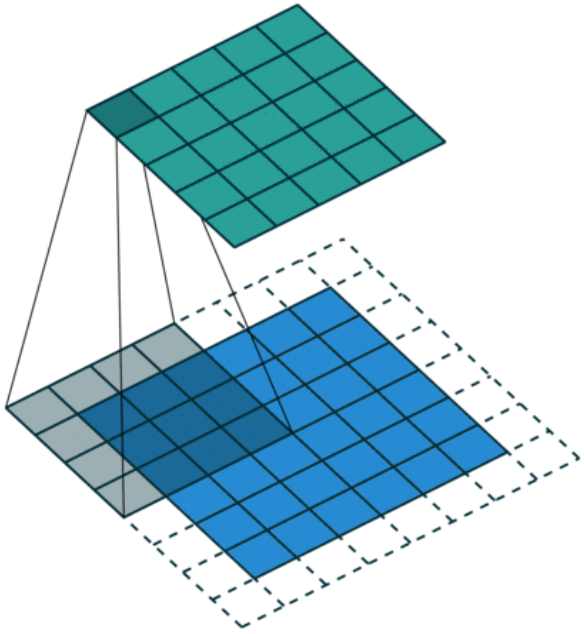


Figure 7. Demonstration of the reduction of neurons

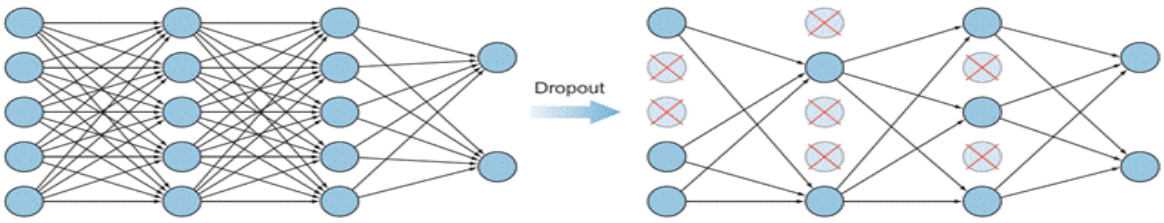


Figure 8. Demonstration of dropout layer

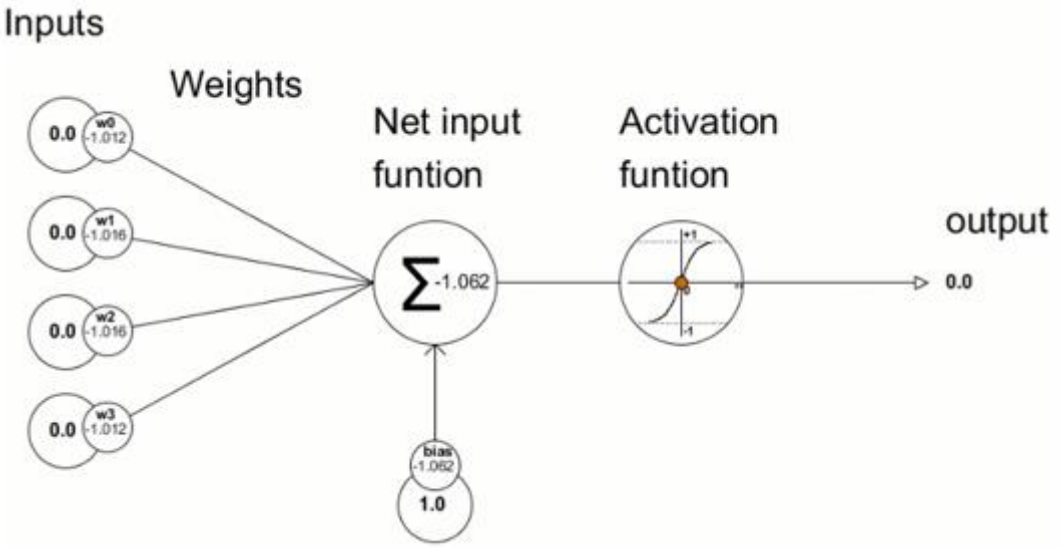


Figure 9. Demonstration of an activation function

3.8. Model Evaluation

The process of quantifying the skill of a prediction algorithm is known as model evaluation. There are several strategies for performing the model evaluation, with cross-validation and holdout validation being two of the most popular. On the other hand, this approach has used holdout validation because different machine learning models are used as candidate models for evaluation. Deep learning models are costly to train, so cross-validation may be time-consuming and should be avoided. Holdout validation is performed by randomly dividing the data into a 80 percent to 20 percent train-test split, with model training using 80% of the data and model evaluation using 20% of the data. Evaluation metrics are important for the quantification of a model's performance. The selection of a particular evaluation metric depends on the prediction algorithm, predictive modeling problem, and dataset type. In the case of credit card fraud detection, classification accuracy is less useful, and therefore, precision and recall are used as a combined metric for quantifying the model's performance. F1-score, a geometric mean between the two entities, can be regarded as a suitable metric to compare candidate models. Accuracy is not considered here due to the unbalanced dataset. Since we are dealing with credit card fraud, we do not want the predictive model to overlook the fraudulent transactions; hence we want to achieve high recall.

However, this does not mean we should overlook the precision because we do not want to mistakenly label a transaction as fraudulent when it is not. The f1 score is a particularly good metric to explore for this problem because it is the harmonic mean of recall and precision. In most classification problems, the ROC curve is the preferred method of assessing the model's performance at various thresholds. The precision-recall curve, on the other hand, will be more useful if the class distribution is unbalanced. Precision and recall are unaffected by relative imbalance since they do not evaluate relative negatives. So here, the precision recall AUC curve scores and the f1 score would be the key evaluation metrics.

3.9. Experimental Results

In this section, results of the various algorithms using the different feature selection techniques, on the unbalanced datasets, including our technique is compared, and then the best model is then used with the different sampling techniques to improve performance. These three feature selection types are the select best, the SVM, the decision tree, the no feature selection and the hybrid (our method), and their results compared.

The first section looks at the random forest algorithm, compares our hybrid technique to the other feature selection methods, and then the best method is run with the different sampling techniques for an improvement in results. Naïve Bayes, logistic regression and XGBoost are all experimented with the same conditions.

3.9.1. Experimental Results of Random Forest

This subsection shows the results of the Random Forest algorithm on all the relevant evaluation metrics using the different feature selection techniques including our proposed method. Table 7. below summarizes the results

Table 7. Comparison of hybrid techniques to other feature selection techniques for the Random Forest Algorithm

PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC	Random Forest with feature selection types
0.9712	0.8793	0.9203	0.8521	0.9902	Without feature selection.
0.9458	0.8792	0.9098	0.8136	0.9238	Decision Tree
0.9661	0.8965	0.9285	0.8556	0.9583	SVM
0.9538	0.8965	0.9232	0.8485	0.964	Select Best
0.9656	0.8908	0.9249	0.8561	0.97	Hybrid

From the table above, our proposed method, which is the hybrid method does better than all other feature selection methods, with improvement of 0.05 %, on the SVM feature selector, on the precision recall AUC score (the key evaluation metric), which suggests that our method is a very robust and high performing method that does exceedingly well to catch the fraudulent cases. This hybrid method can be further applied to other sampling methods to improve the results to make it more robust and catch a balance between precision and recall metrics.

Table 8. The results of the hybrid method with different sampling methods

Random Forest with hybrid feature selection		PRECISION	RECALL	F1-SCORE	PR-RE-AUC	ROC AUC
No Sampling	-	0.9656	0.8908	0.9249	0.8561	0.97
Smote	3 times	0.9724	0.8965	0.9312	0.8693	0.8965
	10 times	0.9328	0.9079	0.92	0.8704	0.9804
	100 times	0.9293	0.9194	0.9243	0.8548	0.985
	Half	0.9336	0.9137	0.9234	0.8508	0.9905
	All	0.9234	0.9137	0.9185	0.843	0.9846
SVMSMOTE	3 times	0.9544	0.9022	0.9267	0.8545	0.9692
Borderline-SMOTE	3 times	0.9604	0.9022	**0.9293**	**0.8753**	0.9754
ADASYN	3 times	0.9604	0.9022	0.9293	0.8687	0.9855
Undersampling	NearMiss-3	0.608	0.9286	0.6716	0.8276	0.9343
	Tomek Links	0.9593	0.8908	0.9222	0.8443	0.9642
	ENN	0.9106	0.8964	0.9034	0.7981	0.9754
	OSS	0.9661	0.8965	0.9285	0.8417	0.9697
	Neighborhood Cleaning Rule	0.9106	0.8964	0.9034	0.8112	0.9698
	RUS	0.5217	0.9553	0.5335	0.7313	0.9883
Combined Data Sampling Methods	Smote*3 with Tomek links	0.9593	0.8908	0.9222	0.8685	0.9806
	Smote*3 with OSS	0.9492	0.908	0.9276	0.8578	0.963
	SMOTE*3 with ENN	0.8976	0.9021	0.8998	0.8211	0.9744

The various sampling methods had their minority classes increased three times, ten times up to more than one hundred times. Initially, after increasing the minority classes 3 times, the performance of the model increases, as now more of the classes are the minority classes, but further increasing the minority class (beyond a 3 times), the performance starts degrading, since now the minority class starts taking over the whole target column, which means there is overlap, and it now becomes the majority class, leading to an unbalanced data, which was the problem we are trying to solve in the first place. From the table, it is evident that the borderline-smote, with the minority classes, increased three times achieves the best score in the precision-recall AUC score, and the smote with minority classes increased three times and achieved the best score in the f1 score.

3.9.2. Experimental Results of XGBoost Model

Different feature selection techniques, including our proposed method, are compared to the without feature selection methods row of the XGBoost algorithm. Table 9 below summarizes the results

Table 9. Comparison of hybrid techniques to other feature selection techniques of the XGBoost Algorithm

PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC	XGBoost
0.9646	0.8793	0.9176	0.8567	0.9823	Without sampler and feature selection.
0.9382	0.8677	0.8999	0.8116	0.9666	Decision Tree
0.9656	0.8908	0.9249	0.8588	0.9919	SVM
0.9587	0.885	0.9186	0.8502	0.9788	Select Best
0.9656	0.8908	0.9249	0.8613	0.9925	Hybrid

Using the XGBoost algorithm, there is evidence that our hybrid feature selection method, doing better than all the other feature selection methods, achieves the same F1 score as the SVM. Still, it has the highest precision-recall AUC score (0.29%) than the SVM, which shows that our method is robust and high-performing and does exceedingly well in catching the fraudulent cases with XGBoost. The hybrid feature selection method results would also be further applied to some sampling methods to balance the data.

Table 10. Hybrid feature selection with different sampling methods of the XGBoost algorithm

XGBoost with Hybrid feature selection		PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC
No Sampling	-	0.9656	0.8908	0.9249	0.8613	0.9925
SMOTE	3 times	0.9609	0.908	0.9328	0.8692	0.908
	10 times	0.9336	0.9137	0.9234	0.8693	0.9759
	100 times	0.9054	0.9194	0.9123	0.8566	0.9873
	Half	0.8762	0.9193	0.8966	0.8548	0.9811
	Equal	0.8736	0.9251	0.8977	0.8501	0.9812
SVMSMOTE	3 times	0.9396	0.9195	0.9293	0.8718	0.9875
Borderline smt	3 times	0.9556	0.9137	0.9336	0.8696	0.9865
ADASYN	3 times	0.9732	0.908	0.9382	0.8552	0.9797
Undersampling	NearMiss-3	0.5099	0.8986	0.5028	0.7318	0.9483
	Tomek Links	0.9604	0.9022	0.9293	0.8674	0.98
	ENN	0.9068	0.9022	0.9045	0.8289	0.9885
	OSS	0.9665	0.9023	0.932	0.8646	0.9891
	Neighborhood Cleaning Rule	0.9033	0.9079	0.9056	0.8085	0.9851
	RUS	0.5151	0.9536	0.5174	0.6841	0.989
Combined Data Sampling Methods	Smote*3 with Tomek links	0.9492	0.908	0.9276	0.8561	0.9855
	Smote*3 with OSS	0.955	0.908	0.9302	0.8581	0.9884
	SMT*3with ENN	0.9033	0.9079	0.9056	0.8078	0.9873

Using the same sampling techniques above, the ADASYN with minority classes increased three times, achieving the best F1 scores. SMOTE SVM with minority classes increased three times, reaching the best score in the precision-recall AUC, suggesting that combining our hybrid method with the SMOTE SVM improves the results in terms of precision-recall AUC since it is the most critical metric in this case.

3.9.3. Experimental Results of Logistic Regression

Table 11 below demonstrates the results of the logistic regression algorithm on all the relevant evaluation metrics using the different feature selection techniques, including our proposed method.

Table 11. Comparison of hybrid techniques to other feature selection techniques of the Logistic regression Algorithm

PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC	Logistic Regression with feature selection types
0.9087	0.7585	0.8167	0.6808	0.9745	Without sampler and feature selection.
0.8873	0.7183	0.7791	0.6536	0.9463	Decision Tree
0.9119	0.77	0.8162	0.675	0.9696	SVM
0.9087	0.7585	0.8167	0.6763	0.9726	Select Best
0.9163	0.7585	0.8189	0.6853	0.9714	Hybrid

From the table above, it is very apparent that the hybrid approach achieves better results on the PR-RE AUC scores (0.66% increase) and the f1 scores (0.27% increase), than the other feature selection techniques. The hybrid technique would further be used to evaluate the different sampling techniques to improve the results. Table 12 below shows the results of the hybrid technique on the different sampling techniques using the logistic regression algorithm.

Table 12. Hybrid feature selection with different sampling methods of the Logistic regression

Logistic Regression with Hybrid feature selection		PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC
No Sampling	-	0.9163	0.7585	0.8189	0.6853	0.9714
SMOTE	3 times	0.8949	0.8677	0.8808	0.6976	0.9794
	10 times	0.8762	0.9021	0.8887	0.6881	0.9797
	100 times	0.6297	0.9234	0.6979	0.6599	0.9797
	Half	0.5471	0.9417	0.5821	0.6695	0.9792
	Equal	0.5252	0.9576	0.5409	0.6651	0.9789
SVMSMOTE	3 times	0.8894	0.8849	0.8871	0.7033	0.985
Borderline smt	3 times	0.8862	0.8906	0.8884	0.6946	0.9862
ADASYN	3 times	0.8734	0.8734	0.8734	0.6773	0.98
Undersampling	NearMiss-3	0.5114	0.828	0.5108	0.1291	0.8889
	Tomek Links	0.9207	0.7758	0.8331	0.6864	0.9712
	ENN	0.9271	0.8045	0.8555	0.6876	0.9718
	OSS	0.9207	0.7758	0.8331	0.6861	0.9706
	Neighborhood Cleaning Rule	0.9228	0.816	0.8617	0.6952	0.976
	RUS	0.5164	0.9555	0.5208	0.5487	0.9775
Combined Data Sampling Methods	Smote*3 with Tomek links	0.8949	0.8677	0.8808	0.698	0.9801
	Smote*3 with OSS	0.8949	0.8677	0.8808	0.6976	0.9799
	SMOTE*3 with ENN	0.8949	0.8677	0.8808	0.696	0.9818

It is evident that smote with ten times the minority classes achieved the best score in the F1, but still, PR-RE AUC is not improving well. The SVMSMOTE with three times increased the minority classes achieved the best precision-recall AUC score, which improved the algorithm's performance by 2.6%.

3.9.4. Experimental Results of Gaussian Naive Bayes

This subsection gives an insight into the Gaussian naïve Bayes algorithm on all the relevant evaluation metrics using the three different feature selection techniques; Table 13 below summarizes the results.

Table 13. Comparison of hybrid techniques to other feature selection techniques of the Gaussian Naive Bayes Algorithm

PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC	GaussianNB with feature selection methods
0.5266	0.897	0.5446	0.4042	0.9719	Without sampler and feature selection.
0.5438	0.9129	0.5761	0.4357	0.9703	Decision Tree
0.5311	0.9155	0.5531	0.4021	0.9766	SVM
0.5497	0.9194	0.5862	0.4518	0.9774	Select Best
0.5499	0.9251	0.5866	0.4609	0.9787	Hybrid

Using the Naïve Bayes algorithm, our hybrid approach again comes out on top, which achieves better results on the precision-recall AUC scores, achieving a 14% increase, and on the f1 scores, achieving a 7.7% increase, than the other feature selection techniques, The hybrid method here combines the features selection methods with higher results. It can also be noticed that this algorithm has low performance across all sampling techniques on the f1 score and the precision-recall score because Naïve Bayes assumes that one feature is not dependent on another or the outcome of one part does not affect the other, which is not the case. The hybrid technique would further be used to evaluate the different sampling techniques to improve the results. Table 14 below shows the results of the Naïve Bayes algorithm, further enhanced by the hybrid feature selection and the other sampling techniques.

Table 14. Hybrid feature selection with different sampling methods of Gaussian Naive Bayes

GaussianNB with hybrid feature selection		PRECISION	RECALL	F1-SCORE	PR-RE AUC	ROC AUC
No Sampling	-	0.5499	0.9251	0.5866	0.4609	0.9787
SMOTE	*3 times*	0.5483	0.9249	0.584	0.4658	0.9786
	10 times	0.5453	0.9244	0.5789	0.4703	0.9789
	100 times	0.5401	0.9235	0.5697	0.4721	0.9789
	half	0.5373	0.9229	0.5647	0.4706	0.9788
	all	0.5348	0.9222	0.56	0.4699	0.9788
SVMSMOTE	*3 times*	0.5468	0.9246	0.5813	0.465	0.9792
Borderline smt	*3 times*	0.5423	0.9239	0.5735	0.4715	0.9804
ADASYN	*3 times*	0.5341	0.922	0.5589	0.4713	0.9778
Undersampling	NearMiss-3	0.6032	0.7285	0.642	0.2455	0.9646
	Tomek Links	0.5499	0.9251	0.5866	0.4609	0.9787
	ENN	0.5496	0.925	0.5861	0.4607	0.9787
	OSS	0.5478	0.9248	0.5831	0.4568	0.9773
	Neighborhood Cleaning Rule	0.5496	0.925	0.5861	0.4607	0.9787
	RUS	0.5293	0.926	0.5496	0.4797	0.9798
Combined Data Sampling Methods	Smote*3 with Tomek links	0.5483	0.9249	0.5838	0.4658	0.9786
	Smote*3 with OSS	0.5465	0.9246	0.5809	0.4617	0.9773
	SMOTE*3 with ENN	0.5474	0.9247	0.5823	0.4708	0.9786

With the hybrid method on the different sampling techniques of the Gaussian Naïve Bayes algorithm, the NearMiss-3 achieved the highest scores on the f1 score and the random under sampling, which down samples of the majority classes obtain the best score in the precision-recall AUC. As explained above, the Naïve Bayes makes certain assumptions, but the best results here are different from the other algorithms.

3.9.5. Experimental Results of Deep Neural Networks (LSTM)

The LSTM is trained using stochastic gradient descent-based optimizers, and the training progress of the LSTM trained using the Adam optimizer is provided in this section. Table 15 below shows the performance of the LSTM model with the different sampling techniques

Table 15. Comparison of LSTM results with different sampling strategies of SMOTE

Deep Learning		PRECISION	RECALL	F1-SCORE	MCC	ROC AUC
No Sampling	-	0.8157	0.7126	0.7607	0.7621	0.8561
SMOTE	3 times	0.9314	0.7938	0.8571	0.8592	0.8967
	10 times	0.9537	0.8792	0.9149	0.91439	0.9392
	15 times	0.9446	0.9523	0.9484	0.9472	0.9755

The LSTM was run on SMOTE with different minority samples and in a situation where no sampling technique was applied. The SMOTE with the minority classes increased fifteen times, achieving the best result. There is an additional metric here, known as the Matthews correlation coefficient (MCC), which is especially useful for unbalanced cases and produces a more informative and truthful score in evaluating binary classifications than accuracy and F1 score,

It is calculated as:

$$MCC = (TP*TN - FP*FN) / \sqrt{(TP+FP)\,(TP+FN)\,(TN+FP)\,(TN+FN)} \quad (5)$$

If the value for MCC ranges "-1" indicates impropriety between predicted classes and actual classes if its "0" refers to utterly random guessing, and "1" indicates conformity between predicted classes and actual classes.

Deep learning models (LSTM in this case) are data hungry, meaning they achieve the best results on more data. However, increasing the minority class further will cause memorization of the data, which is why we stopped at fifteen sampling strategy numbers. Increasing the minority classes of the smote fifteen times achieved the best results here. Figure 10. below shows the loss and accuracy graphs running the LSTM sampled by SMOTE with minority classes increased fifteen times (highest performing model).

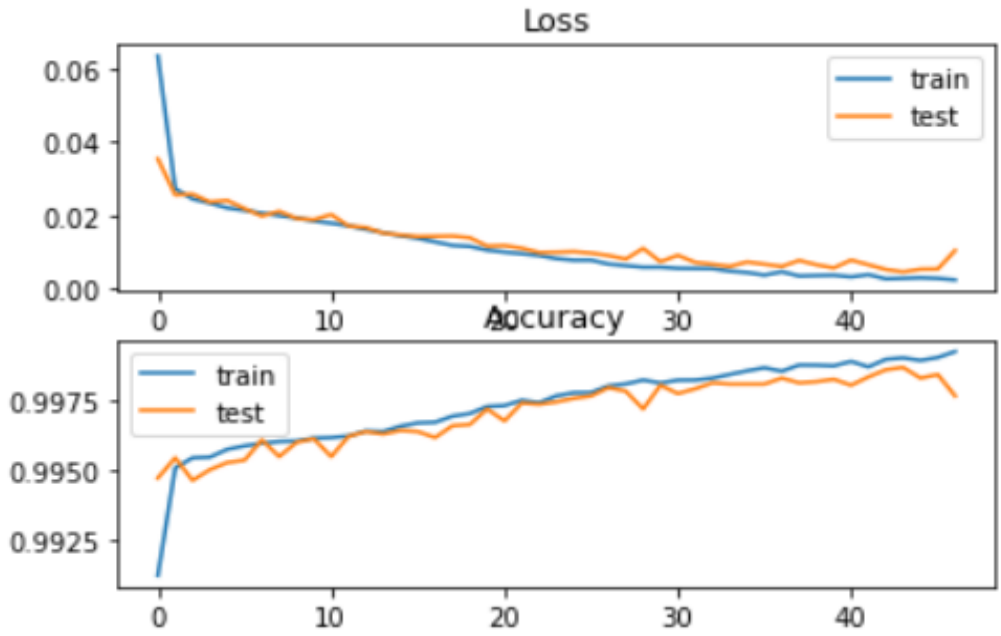


Figure 10. Loss and Accuracy graphs of the LSTM model

As can be seen, in the initial stages of training, the losses are higher for both the train and test, as the model is now learning the features. Still, as more iterations are being run, the loss keeps going lower as the model is currently learning the features. The train and test loss are close together, suggesting that overfitting has not occurred. Also, looking at the accuracy graph, the accuracy of both train and test is initially relatively lower. Still, as more iterations are being run, the model keeps learning the features, and the accuracy increases. The train and test accuracies are close together, suggesting no overfitting occurred. There are some slight distortions in the graph, a bit of a zigzag; this is because of the batch size and learning rate we used, as batch size and learning rate affect the rate of updates from one epoch to the next, hyperparameter tuning can make it smoother.

3.9.6. Comparison of All the Machine Learning Models

A comparison of the four machine learning classification models is made in this section. Table 16. provides the precision-recall AUC score and F1-score of the classifier models without resampling techniques.

Table 16. Comparison of the performance of the machine learning models using the hybrid feature selection

#	Classification Model	Precision-Recall AUC score	F1-Score
1	Random Forest	85.61%	92.49%
2	XGBoost	86.13%	92.49%
3	Logistic Regression	68.53%	81.89%
4	Gaussian Naive Bayes	46.09%	58.66%

It is seemingly straightforward that for the precision-recall-AUC, the XGBoost scored the highest with an 86.13% score without any resampling techniques, and for the F1 score, both the random forest and XGBoost scored the same; however, the precision-recall AUC score is our key evaluation metric here; the comparison results of different resampling techniques revealed that the random forest with hybrid feature selection approach and Borderline SMOTE resampling performed better. Hence, we achieved 0.9293 F1 scores and 0.8753 PR-RE AUC scores. The LSTM model was not included in this comparison because deep learning models do not need any feature selection, as they learn the features themselves, so it does not utilize our hybrid approach, hence its inclusion in the comparison. The comparison results are in table 17. Compare our results with several other studies.

Table 17. Comparison with previous studies on credit card fraud detection dataset

Model	PRECISION	RECALL	F1-SCORE	AUC
RF with hybrid feature selection and resampling.	**0.9604**	**0.9022**	**0.9293**	**0.9754**
RF+oversampling [36]				0.1
Distributed Random Forest with smote [37]	0.8643	0.8176	0.8403	
Logistic regression with smote [38]	0.9999	0.9704		0.9704
RF with smote [39]	0.93	0.82	0.88	
RF with RUS[40]	0.94	0.94	0.94	
Decision tree with Near Miss [41]			0.97	0.96
RF with SMOTE + Tomek links [42]	0.84	0.84	0.84	
Proposed method with LSTM + smote.	**0.9446**	**0.9523**	**0.9484**	
LSTM [43]			0.8485	
ANN [44]	0.81	0.76		
ANN [45]	98.41	98.98		
LSTM [46]	0.9885	0.9191		

We observe from table 17. that a few studies obtained higher results than our proposed method. E.g., [36] with the random Forest has the highest AUC score because they applied dataset resampling (Under and Oversampling) before splitting the data into train and test. This leads to the information leaking from the training set into the test set, which we avoid it with the machine learning algorithms. And in [38], the logistic regression performs well with a precision of 0.99 and recall of 0.9704 due to utilizing SMOTE strategy with several KNN; The proposed method results will improve by tuning the hyperparameters. Also, in [40], the results are high; however, the number of under-sampled data is significantly less and cannot be generalized with a high confidence level [41]. The deep learning model at [46,47] performed better than our method because the training set is not raw data. Our study's limitations concerning time and many comparisons complicate it from getting a chance with multiple hyperparameter tunning.

4. RESEARCH CONTRIBUTIONS

Credit card fraud detection is a dynamic and evolving problem, and while various researchers have attempted to address the issue, a reliable and efficient method remains elusive. This study tried to solve the problem by introducing a new method known as the hybrid approach, which uses a combination of wrapper-based and filter-based methods to produce a robust selection of features to achieve high performance, and this addresses the challenges in modeling credit card fraud detection and providing a system with higher accuracy than its competitors. The proposed approach performed better than all the other feature selection methods across four different algorithms on the precision-recall AUC score and the F1 score, which are the key metrics to consider when dealing with unbalanced datasets because the precision-recall curve does not assess the negatives. Hence, only the fraudulent cases (in which we have interest) would be evaluated, and the F1 score is the harmonic mean between precision and recall, which strikes a nice balance. Our hybrid technique was then applied to several sampling techniques to improve the results by catching a nice balance between the two classes. Therefore, our proposed technique is a robust and high-performing feature selection technique that can see fraudulent cases with such high performance. The Random under sampler (RUS) appears to have low performance across all algorithms; this is because under sampling reduces the dataset to match the minority classes; hence there is a reduction in data, which leads to a reduction in the information that the model can learn from to make good predictions, and also because it would lead to applying sampling on a train data that is smaller than the testing data. Finally, we applied LSTM and experimented with various levels of minority classes using the SMOTE sampling technique because it improved the result for each algorithm. Increasing the minority classes fifteen times achieved the best result by running a hyperparameter search on the LSTM; an improvement in the results would be achieved since it would be optimally combining the best hyperparameters, which would yield better results.

5. REFERENCES

1. Han, Q. and Cho, D., Characterizing the Technological Evolution of Smartphones: Insights From Performance Benchmarks, in proceedings of the 18th Annual International Conference on Electronic Commerce: E-Commerce in Smart Connected World, 2016, 1-8.

2. Abdallah, A., Maarof, M.A. and Zainal, A., Fraud Detection System: A Survey, Journal of Network and Computer Applications, 68 (2016) 90-113.

3. Prabowo, H.Y., Building Our Defense Against Credit Card Fraud: a strategic view, Journal of Money Laundering Control, 14, 4 (2011) 371-386

4. Lucas, Y., and Jurgovsky, J., Credit Card Fraud Detection Using Machine Learning: A Survey, arXiv preprint arXiv:2010.06479, 2020.

5. Jordan M.I., and Mitchell, T.M., Machine Learning: Trends, Perspectives, and Prospects, Science, 349, 6245 (2015) 255-260.

6. Liu, A., Ghosh, J., and Martin, C. E., Generative Oversampling for Mining Imbalanced Datasets, in DMIN, 2007, 66-72.

7. Chawla, N.V., Data Mining for Imbalanced Datasets: An Overview, Data mining and knowledge discovery handbook, 209 (2005) 875-886.

8. Zeng, M., Zou, B., Wei, F., Liu, X. and Wang, L., Effective Prediction of Three Common Diseases by Combining SMOTE with Tomek Links Technique for Imbalanced Medical Data, in 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), 2016, 225-228.

9. Lin, W.-C., Tsai, C.-F., Hu, Y.-H. and Jhang, J.-S., Clustering-Based Undersampling in Class-Imbalanced Data, Information Sciences, 409 (2017) 17-26.

10. Sarkar, S., Khatedi, N., Pramanik, A. and Maiti, J., An Ensemble Learning-Based Undersampling Technique for Handling Class-Imbalance Problem, in Proceedings of ICETIT Springer, 2020, 586-595.

11. Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W. P., SMOTE: Synthetic Minority Over-Sampling Technique, Journal of artificial intelligence research, 16 (2002) 321-357.

12. He, H., Bai, Y., Garcia, E.A. and Li, S., ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, in 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), 2008, 1322-1328.

13. Shorten, C. and Khoshgoftaar, T. M., A Survey on Image Data Augmentation for Deep Learning, Journal of big data, 6, 1 (2019) 1-48.

14. Bahnsen, A.C., Aouada, D. Stojanovic, A. and Ottersten, B., Feature Engineering Strategies for Credit Card Fraud Detection, Expert Systems with Applications, 51 (2016) 134-142.

15. Lee, Y.-J., Yeh, Y.-R. and Wang, Y.-C. F., Anomaly Detection Via Online Oversampling Principal Component Analysis, IEEE transactions on knowledge and data engineering, 25, 7 (2012) 1460-1470.

16. Pradhan, S.K., Rao, N.K., Deepika, N., Harish, P., Kumar, M. P. and Kumar, P.S., Credit Card Fraud Detection Using Artificial Neural Networks and Random Forest Algorithms, in 2021 5th International Conference on Electronics, Communication, and Aerospace Technology (ICECA), 2021, 1471-1476.

17. Rathore, A.S., Kumar, A., Tomar, D., Goyal, V., Sarda, K. and Vij, D., Credit Card Fraud Detection using Machine Learning, in 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, 167-171.

18. Gürsoy, G. and Varol, A., Risks of Digital Transformation: Review of Machine Learning Algorithms in Credit Card Fraud Detection, in 2021 2nd International Informatics and Software Engineering Conference (IISEC), 2021, 1-6.

19. Hussain, S.S., Reddy, E.S.C., Akshay, K.G. and Akanksha, T., Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms, in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2021, 1013-1017.

20. Tyagi, R., Ranjan, R. and Priya, S., Credit Card Fraud Detection Using Machine Learning Algorithms, in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021, 334-341.

21. Uttam, A.K. and Sharma, G., A., Comparison of Data Balancing Techniques for Credit Card Fraud Detection using Neural Network, in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021, 1136-1140.

22. Jenipher, V.N., Rose, J.D., Sabharam, M. and Nithin, M., Learning Algorithms with Data Balancing in Credit Card Fraud Detection Application, in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021, 1-6.

23. Krishnan, M. and Srinivasan, M.K., Credit Card Fraud Detection: An Exploration of Different Sampling Methods to Solve the Class Imbalance Problem, in Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences, 2022, 825-837.

24. Kumar, S., Gunjan, V.K., Ansari, M.D. and Pathak, R., Credit Card Fraud Detection Using Support Vector Machine, in Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications, 2022, 27-37.

25. Singh, P., Chauhan, V., Singh, S., Agarwal, P. and Agrawal, S., Model for Credit Card Fraud Detection using Machine Learning Algorithm, in 2021 International Conference on Technological Advancements and Innovations (ICTAI), 2021, 15-19.

26. Arafath, Y., Roy, A.C., Shamim Kaiser, M. and Arefin, M.S., Developing a Framework for Credit Card Fraud Detection, in Proceedings of the International Conference on Big Data, IoT, and Machine Learning, 2022, 637-651.

27. Abosamra, G. and Faloudah, A., Machine Learning Based Marks Prediction to Support Recommendation of Optimum Specialization and Study Track, International Journal of Computer Applications, 181, 49 (2019) 975- 8887.

28. Juszczak, P., Tax, D. and Duin, R.P., Feature Scaling in Support Vector Data Description, in Proc. asci, Citeseer, 2002, 95-102.

29. Oshiro, T.M., Perez, P.S. and Baranauskas, J.A., How Many Trees in A Random Forest?, in International workshop on machine learning and data mining in pattern recognition, Springer, 2012, 154-168.

30. Qi, Y., Random Forest for Bioinformatics, in Ensemble machine learning: Springer, 2012, 307-323.

31. Weston, J. and Watkins, C., Support Vector Machines for Multi-Class Pattern Recognition, in Esann, 99 (1999) 219-224.

32. Wright, R.E., Logistic Regression, In L. G. Grimm, & P. R. Yarnold (Eds.),1995, 217-244.

33. Murphy, K.P., Naive Bayes Classifiers, University of British Columbia, 18, 60 (2006) 1-8.

34. Lewis, D.D., Naive (Bayes) at forty: The Independence Assumption in Information Retrieval, in European conference on machine learning, Springer, (1998) 4-15.

35. Pérez, A., Larrañaga, P. and Inza, I., Bayesian Classifiers Based on Kernel Density Estimation: Flexible classifiers, International Journal of Approximate Reasoning, 50, 2 (2009) 341-362.

36. Ayorinde, K., A Methodology for Detecting Credit Card Fraud Kayode Ayorinde, Thesis Master's in Data Science Minnesota Sta, Cornerstone A Collection of Scholarly and Creative Works for Minnesota State University, Mankato, 2021.

37. Muaz, A., Jayabalan, M. and Thiruchelvam, V., A Comparison of Data Sampling Techniques for Credit Card Fraud Detection, Int. J. Adv. Comput. Sci. Appl., 11, 6 (2020) 477–485.

38. Sriram Sasank, J.V.V.G., Sahith, Abhinav, R. K. and Belwal, M., Credit Card Fraud Detection Using Various Classification and Sampling Techniques: A Comparative Study, Proc. 4th Int. Conf. Commun. Electron. Syst. ICCES, Icces, 2019,1713–1718.

39. Erdoğan, T., Credit Card Fraud Detection Using Machine Learning, Thesis Master, M.E.F University, İstanbul, 2021.

40. Mahesh, K.P., Afrouz, S.A. and Areeckal, A.S., Detection of Fraudulent Credit Card Transactions: A comparative analysis of data sampling and classification techniques, J. Phys. Conf. Ser., 2161, 1, 2022.

41. Kaur, K., Credit Card Fraud Detection using Imbalance Resampling Method with Feature Selection, Int. J. Adv. Trends Comput. Sci. Eng., 10, 3 (2021) 2061–2071.

42. Shakya, R., Application of Machine Learning Techniques in Credit Card Fraud Detection, UNLV Theses, Diss. Prof. Pap. Capstones, University of Nevada, Las Vegas, 2018.

43. Nguyen, T.T., Tahir, H., Abdelrazek, M. and Babar, A.i., Deep Learning Methods for Credit Card Fraud Detection, 2020, [Online]. Available: http://arxiv.org/abs/2012.03754.

44. RB, A. and KR, S.K., Credit Card Fraud Detection Using Artificial Neural Network, Glob. Transitions Proc., 2, 1 (2021) 35–41.

45. Tyagi, R., Ranjan, R. and Priya, S., Credit Card Fraud Detection Using Machine Learning Algorithms, Proc. 5th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2021, 9, 07 (2021) 334–341.

46. Benchaji, I., Douzi, S., El Ouahidi, B. and Jaafari, J., Enhanced Credit Card Fraud Detection Based on Attention Mechanism and LSTM Deep Model, J. Big Data, 8, 151 (2021).

CURRICULUM VITAE

She received a Bachelor's degree in Computer Engineering from Kirkuk Technical University, IRAQ 2016-2017. she worked as a teaching assistant at Kirkuk Technical University. In 2018, She came to Trabzon TURKEY and enrolled for a Master's degree in Karadeniz Technical University Computer Engineering Department.