Fraudulent credit card transaction detection using soft computing techniques

Aishwarya Priyadarshini[1], Sanhita Mishra[2], Debani Prasad Mishra[3], Surender Reddy Salkuti[4], Ramakanta Mohanty[5]
[1]Department of Computer Science and Engineering, IIIT Bhubaneswar, India
[2]Department of Electrical Engineering, KIIT Deemed to be University, India
[3]Department of Electrical Engineering, IIIT Bhubaneswar, India
[4]Department of Railroad and Electrical Engineering, Woosong University, Republic of Korea
[5]Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, India

Article Info

Article history:

Received Dec 8, 2020
Revised Jul 29, 2021
Accepted Aug 4, 2021

Keywords:

Data distribution
Exploratory data analysis
Fraud detection
Outliers

ABSTRACT

Nowadays, fraudulent or deceitful activities associated with financial transactions, predominantly using credit cards have been increasing at an alarming rate and are one of the most prevalent activities in finance industries, corporate companies, and other government organizations. It is therefore essential to incorporate a fraud detection system that mainly consists of intelligent fraud detection techniques to keep in view the consumer and clients' welfare alike. Numerous fraud detection procedures, techniques, and systems in literature have been implemented by employing a myriad of intelligent techniques including algorithms and frameworks to detect fraudulent and deceitful transactions. This paper initially analyses the data through exploratory data analysis and then proposes various classification models that are implemented using intelligent soft computing techniques to predictively classify fraudulent credit card transactions. Classification algorithms such as K-Nearest neighbor (K-NN), decision tree, random forest (RF), and logistic regression (LR) have been implemented to critically evaluate their performances. The proposed model is computationally efficient, light-weight and can be used for credit card fraudulent transaction detection with better accuracy.

Corresponding Author:

Surender Reddy Salkuti
Department of Railroad and Electrical Engineering
Woosong University
17-2, Jayang-Dong, Dong-Gu, Daejeon-34606, Republic of Korea
Email: surender@wsu.ac.kr

1. INTRODUCTION

For years, fraud and illegal transactions have been significant problems in banking, medicine, and insurance, among others. Because of the increased reliance on the internet, the amount of total online credit card transactions has increased dramatically across various payment methods such as PhonePe, Gpay, Paytm, and others. Creating a secure framework for safer online transactions with advanced authentication services that can help prevent fraud and other fraudulent transaction practices has become an increasingly difficult job, given that no system is perfect. There is always the likelihood of a flaw. Fraudulent or deceptive credit card purchases may be interpreted as uncertified or unlawful credit card use, which is rare [1]-[8]. Fraud is a complex problem that involves engaging in illegal or illicit financial gain while violating laws, regulations, and policies [9], [10]. Random forest (RF), K-Nearest neighbor (KNN), multilayer perceptron (MLP), extreme learning machine (ELM), and bagging classifier were among the machine learning algorithms used to define and classify credit card transaction data into fraudulent and non-fraudulent categories [11]-[14]. Fraud detection in credit card transactions has piqued researchers' interest. The number of techniques for detecting fraudulent or deceptive activity in the realm of financial transactions, whether online or offline, has grown dramatically. Ghosh [15] suggested the method of fraud detection using neural networks, which consisted of a large dataset that included stolen or lost credit cards, phishing, cyber-attacks, non-received issue (NRI) frauds, and so on. Carcillo *et al.* [16] merged the commonly used supervised learning and unsupervised learning algorithms to identify fraudulent patterns in credit card transactions. The automated method [17] was used to examine all current transactions and assign each one a valid or fraudulent ranking. These fraud detection systems are based on expert-driven rules that form a complete collection of fraud detection rules and patterns when combined with a reliable data source. The rules are based on patterns found in the data using various intelligent machine learning algorithms, such as logistic regression (LR) and support vector machines (SVM) [18], [19].

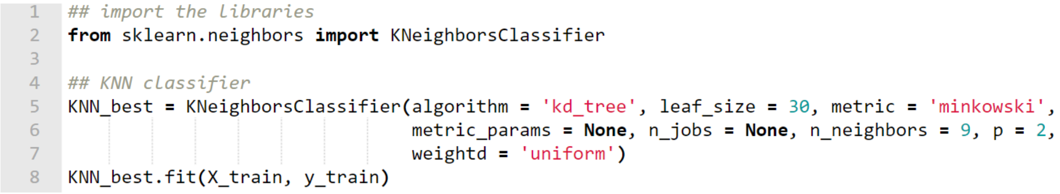
Andrea *et al.* [20] present a significantly challenging task for machine learning algorithms for several reasons, i.e., seasonality, and skewness. Johannes *et al.* [21] address the problem of fraud detection that primarily the used long short term memory (LSTM) networks as a machine-learning algorithm to detect fraudulent transactions and tell the genuine and fraud ones apart. Sam *et al.* [22] proposed a model that combines neural network classifiers and bayesian networks to differentiate a fraudulent credit card transaction from a legitimate one. Similarly, Bolton and Hand [23] conducted research that describes various fraudulent credit card activities and keeps up-to-date with newer deception techniques that fraudsters can employ. Zareapoor, Seeja, and Alam [24] proposed a comparative study based on the performance of various machine learning techniques. It was primarily to review the different credit card fraud techniques.

This paper aims to improve the predictive accuracy of distinguishing fraudulent financial transaction data from legitimate financial transaction data. Gain a comprehensive understanding of the latest research papers describing the application of various machine learning and artificial intelligence techniques to identify credit card frauds and employ an efficient classification model with higher accuracy and higher faster prediction rate for fraud detection for the financial industry, and government organizations. Ultimately, these soft computing techniques and the proposed methodology put together aids in achieving perfect accuracy, and this eliminated the need to use heavier learning algorithms, hence reducing the huge computational overhead and increasing the predictive performance of the detection system.

2. OVERVIEW OF MACHINE LEARNING TECHNIQUES APPLIED

2.1. K-Nearest neighbour classifier

K-Nearest neighbour is a supervised and pattern classification learning algorithm that helps us figure out the class to the new input (test value). When k nearest neighbors are chosen, distance is determined between them. It tries to predict the conditional distribution of Y given X and assign a given observation (test value) to the class with the highest estimated probability. It calculates the distance between all of those categories after identifying the k points in the training data nearest to the test value. The test result would fall into the group with the shortest distance [25]. Even though KNN is computationally expensive, it achieves very good output reliably, without any analytical assumptions relevant to the distributions in which the training samples are fetched properly, out of the different techniques that have been used to identify fraudulent activities in credit card transactions in the literature. If the bordering or closer neighbor is recognized as a fraudulent transaction, it is termed as fraud one [26]. The code snippet for training the KNN classifier has been presented in Figure 1.



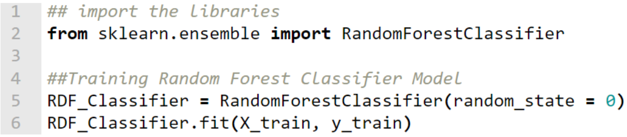
```
## import the libraries
from sklearn.neighbors import KNeighborsClassifier

## KNN classifier
KNN_best = KNeighborsClassifier(algorithm = 'kd_tree', leaf_size = 30, metric = 'minkowski',
                                metric_params = None, n_jobs = None, n_neighbors = 9, p = 2,
                                weightd = 'uniform')
KNN_best.fit(X_train, y_train)
```

Figure 1. Training KNN classifier model

2.2. Random forest classification technique

It is also a supervised learning technique that can be efficiently used for solving classification and regression problems. It is handy for computing solutions for situations where the dataset must be classified [14], [27]. A set of decision trees (DT) are a helpful way of obtaining the prediction values. A group of decision trees (DT) is built, which are then used for predicting the class. The voting technique of the majority can be used for obtaining the final predictive output from the tree that resembles closely. The computational efficiency of Random Forest classifiers lies in the fact that the construction of each tree does not depend on others [28]. The code snippet for training the random forest classifier has been presented in Figure 2.



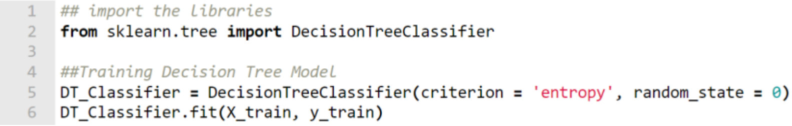
```python
## import the Libraries
from sklearn.ensemble import RandomForestClassifier

##Training Random Forest Classifier Model
RDF_Classifier = RandomForestClassifier(random_state = 0)
RDF_Classifier.fit(X_train, y_train)
```

Figure 2. Training random forest classifier model

2.3. Decision tree

The primary purpose of a decision tree is classification. The decision tree consists of arcs and nodes, which makes its resemblance with a data structure. The nodes of the decision trees imply a decision, and the arcs denote the outcome of that decision [29]. The design of a decision tree is primarily done using a top-down approach. Feature Selection measures can be used to find the best possible splitting point. The function values are compared to the decision tree's nodes or values. The code snippet for training the decision tree classifier has been presented in Figure 3.



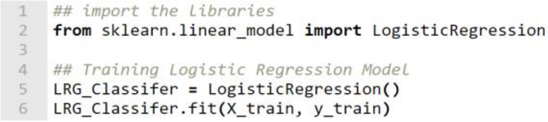
```python
## import the Libraries
from sklearn.tree import DecisionTreeClassifier

##Training Decision Tree Model
DT_Classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
DT_Classifier.fit(X_train, y_train)
```

Figure 3. Training decision tree classifier model

2.4. Logistic regression

Logistic regression (LR) is mainly a predictive analysis model used in a regression analysis where the classification output variable is binary. The output dependent variable is estimated using a series of relevant parameters among the independent variables. The code snippet for training the logistic regression classifier has been presented in Figure 4.



```python
## import the Libraries
from sklearn.linear_model import LogisticRegression

## Training Logistic Regression Model
LRG_Classifer = LogisticRegression()
LRG_Classifer.fit(X_train, y_train)
```

Figure 4. Training logistic regression classifier model

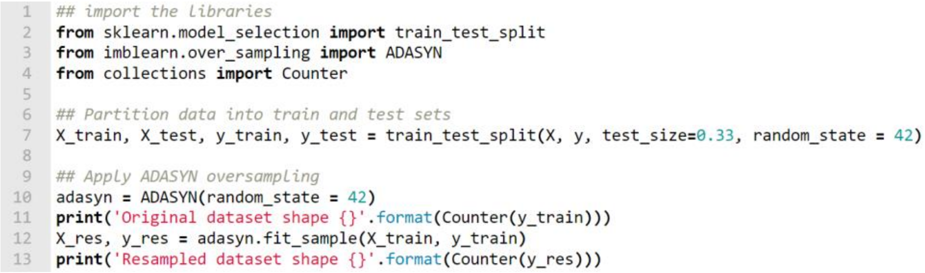
3. DATASET PREPARATION

This research work uses a publicly available dataset [14] that contains only 492 fraudulent credit card transactions out of a whopping total of 284,315 transactions made within the two days. It shows that the dataset is highly skewed or imbalanced with the class 1 or the fraudulent credit card transaction, accounting for a total of 0.172% of the total number of credit card transactions. The dataset features V_1,V_2,V_3,…,V_28, are all numeric values that were obtained by applying principal component analysis (PCA) transformation on the original records to maintain the confidentiality of the users and credit cardholders. The only features that are not transformed and pre-existing numeric values are “Time” and “Amount”. Features “Time” and “Amount” are the time elapsed in seconds between each transaction and the total transaction amount, in $ respectively. The dataset details have been presented in Table 1.

Table 1. Description of dataset

Normal transactions	Fraudulent transactions	Features	Number of transactions
284,315	492	30	284,807

Our main objective in carrying out the exploratory data analysis is to analyze and understand the complete distribution of the data and identify correlation and dependency among various input features. Starting, with understanding the dataset, as explained above, only two of the many input features are known, i.e., “Amount” and “Time”. These columns are already scaled and are ready to be used in our experiment. Our dataset is highly imbalanced, where only 492 (0.172%) out of 284,807 transactions are fraudulent, and the rest 283,823 (99.83%) are genuine transactions. This dataset is highly imbalanced, and to compensate for the high imbalance in the dataset, the adaptive synthetic (ADASYN) oversampling [16] method has been used to resample the dataset. Following which, the Machine learning algorithms were applied to the resampled dataset. The code snippet for implementation of ADAYSN oversampling has been presented in Figure 5.



```
## import the Libraries
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import ADASYN
from collections import Counter

## Partition data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state = 42)

## Apply ADASYN oversampling
adasyn = ADASYN(random_state = 42)
print('Original dataset shape {}'.format(Counter(y_train)))
X_res, y_res = adasyn.fit_sample(X_train, y_train)
print('Resampled dataset shape {}'.format(Counter(y_res)))
```

Figure 5. Code snippent for implementation of ADASYN oversampling

After resampling, the genuine and fraudulent class distributions, the evaluation and estimation of the various features significant towards models’ input features are carried out. Even though, the features in the data are not revealed due to confidentiality issues, it is highly appropriate that the features, V_1,V_2,V_3,…,V_28, which are PCA normalized, while “time”, “amount”, and “class” are non-PCA normalized features, be tested for their correlation and linear dependency on the rest of features and how including or excluding a feature can affect the overall outcome. Therefore, the exploratory data analysis (EDA) is carried out to analyze and understand the complete distribution of the data and identify correlation and dependency among various input features. Starting, with understanding the dataset, as explained above, only two of the many input features are known, i.e., “Amount” and “Time”. These columns are previously scaled and are ready to be used in our experiment. Our dataset is highly imbalanced, where only 492 (0.172%) out of 284,807 transactions are fraudulent and the rest 283,823 (99.83%) are genuine transactions. The 2-D scatter plot was plotted to accurately visualize and realize the credit card transaction in both classes, i.e., class 0 and class 1, representing genuine credit card transactions and fraudulent card transactions, respectively. The 2-D scatter plot has been presented in Figure 6. Again, to accurately verify the transaction distribution, we plot the 3-D scatter plot and pair plot of all the combinations of the features “Time” and “Amount”, presented in Figure 7.

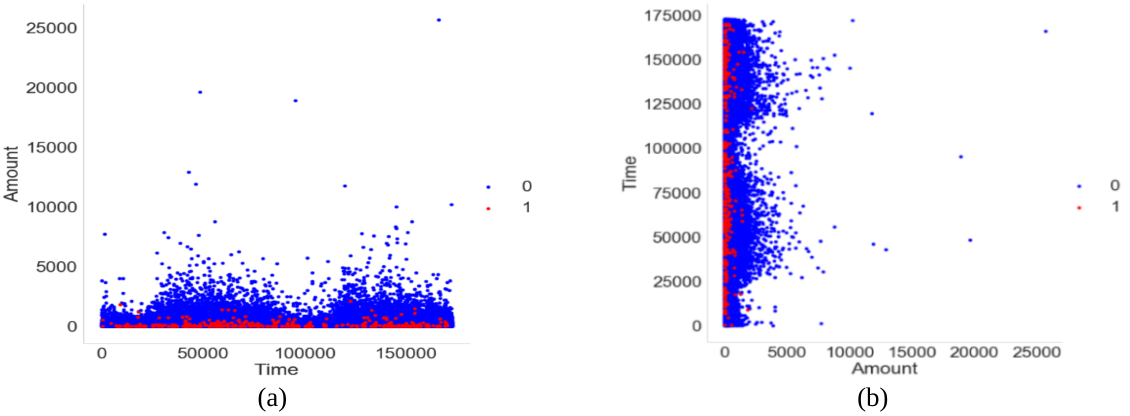


Figure 6. 2-D Scatter plot representing; (a) Distribution of fraudulent transactions vs amount ($), (b) Distribution of fraudulent transactions vs time (sec)

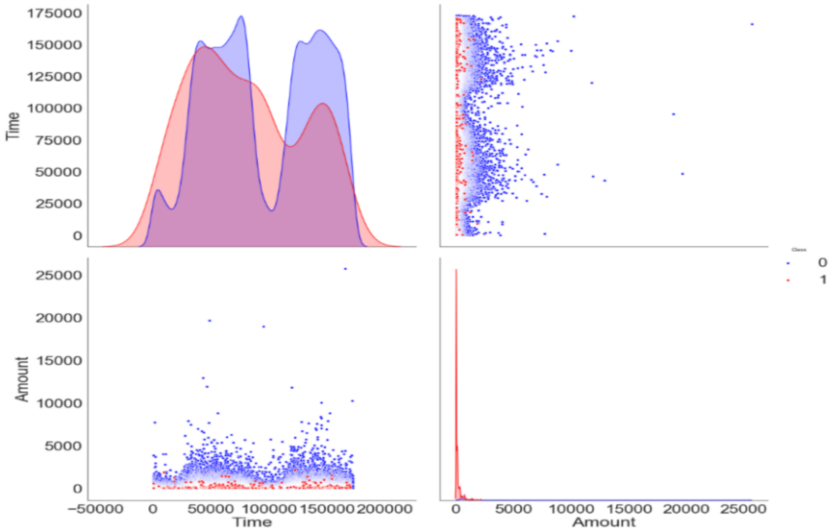


Figure 7. 3-D scatter plot and pair plot of the transaction vs time and transaction vs amount respectively

From the above 3-D scatter plot, first considering plotting the credit card transaction distribution versus the feature “Time”, one can see that whether the credit card transaction is genuine or fraudulent, it has been evenly distributed throughout the time. There is no specific distinction made when it comes to time, i.e. it indicates no significant pattern relevant to fraudulent credit card transactions within the specified time. Similarly, the 3-D scatter plot of the transaction distribution versus “Amount”, can be analyzed that most or all of the fraudulent transactions have been carried out on amounts less than $2500. On computation, out of a total of 284,807 transactions, 284,357 have a transaction amount less than or equal to $ 2500, accounting for 99.84 percent of the total transactions. In contrast, no fraudulent transaction has a transaction amount more significant than $ 2500. Similarly, another important observation being the transaction, whether fraudulent or genuine, was spread evenly throughout time, and there is no clear distinction. There is a heavy overlap of genuine and fraudulent transactions throughout the time and it can be observed that there is no clear distinction. So, there is no particular pattern about fraudulent behavior specific to any part of the day. It is evenly spread throughout 2 days. The histogram, pair plot along distplot for it has been presented in Figure 8.

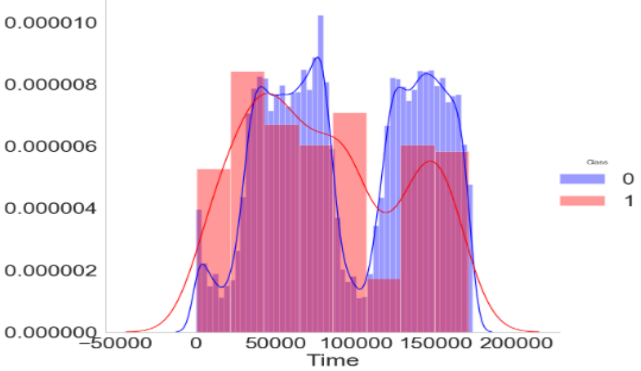


Figure 8. Histogram, pairplot, and distplot to show a heavy overlap of genuine and fraudulent transactions with no clear distinction

For the distplot, the paper uses kernel density estimator (KDE). KDE helps in discerning out the crucial features of the data, such as skewness, bimodality, and central tendency. However, it has its pitfalls. KDE tends to represent the underlying data poorly in certain situations, as it assumes that the underlying distribution is smooth and unbounded. This assumption fails when a variable (here, time) is naturally bounded and there are observations or transactions elapsed at around the starting of the day (assumed to be 0 sec), KDE curve here extends to unrealistic values, resulting in time above being plotted to negative values, which cannot be negative.

Now, the imbalance in our dataset has been analyzed. Considering the skewness of the dataset, to effectively fit our classifier models and generate efficient classifications, the most common techniques are either applying under-sampling or oversampling to even out the imbalance in the dataset. Here, however, ADASYN oversampling technique has been considered [30]. Assuming random under-sampling is implemented on our dataset, the samples from the majority class in the data are deleted randomly, and there is no way whatsoever one can preserve the essential or information-rich samples from the data. Nevertheless, as already proposed in our methodology, ADASYN oversampling technique was implemented to resample the dataset to efficiently train the classifier algorithms.

4. METHODOLOGY

Our study used the dataset [14], which is highly imbalanced, and the balancing is carried out using the ADASYN oversampling method. Following this, the dataset is cross-validated using strategic ten-fold cross-validation. The classifier models were evaluated and tested for their performance on the data using various performance metrics and parameters such as accuracy, recall, precision, and F1-score, which were computed using the confusion matrix. The estimated performance parameters are then compared with every model used in this study to classify the data into class 0 and class 1, i.e., genuine class and fraudulent class credit card transactions, respectively. A confusion matrix, in general, computes the performance of the classifier model while assigning an input to the labels. For binary classification, it is a 2×2 table or a 2-dimensional matrix designed to compute and represent the quantity of all four results of a binary classifier and denoted as TN, FN, TP, and FP [12]. The entries of the confusion matrix have been briefly explained below:

- True – Positives: Correctly Classified Fraudulent Credit Card Transactions
- False – Positives: Erroneously Classified Fraudulent Credit Card Transactions
- True – Negative: Correctly Classified Non-Fraudulent or genuine credit card Transactions
- False – Negative: Erroneously Classified Non-Fraudulent or genuine credit card Transactions

Accuracy: Accuracy is the performance parameter used to calculate to determine the extent to which the classification algorithm can predict the classes correctly.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Recall: Recall is a performance parameter that represents how efficient the model or the classifier is in detecting the actual fraudulent credit card transaction.

$$\text{Recall} = \frac{TP}{FP+FN} \tag{2}$$

Precision: This is the performance parameter that measures the reliability of a model or the classifying algorithm.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

***Recall/Precision Tradeoff*:** It is a very significant trade-off between precision versus recall. It indicates that, as the precision with which the classifier predicts increases, it will detect a lesser number of fraudulent cases, viz. suppose that the classifier model has a precision of 95%, with only five cases of fraudulent credit card transactions. Suppose one tries adding another five fraudulent transaction cases to it. In that case, our classifier model considers 90% precision, i.e., the lower the precision, our classifier model would be able to predict more number of such cases.

F1–Score or F1–measure: F1-Score, a performance parameter estimated to find the testing accuracy of the fraudulent credit card detection classifier model or the classifying algorithm. To calculate the score, it considers evaluating the harmonic mean of two performance metrics: precision and recall of the test samples [13].

$$\text{F1-score} = \frac{2*TP}{2*TP+FP+FN} \tag{4}$$

5. RESULTS AND DISCUSSION

The generated resampled dataset was then divided into training and testing sets, with the testing set having a split ratio of 0.33. To further increase the efficiency and performance of our learning algorithms, we employed stratified k-fold cross-validation with k=10. This will helps us with model results being less biased and less optimistic towards the minority class and any other outliers in the data. The correlation matrix was used to identify which features have high positive and negative correlation or dependency concerning fraudulent credit card transactions. The correlation matrix has been plotted and presented in Figure 9.

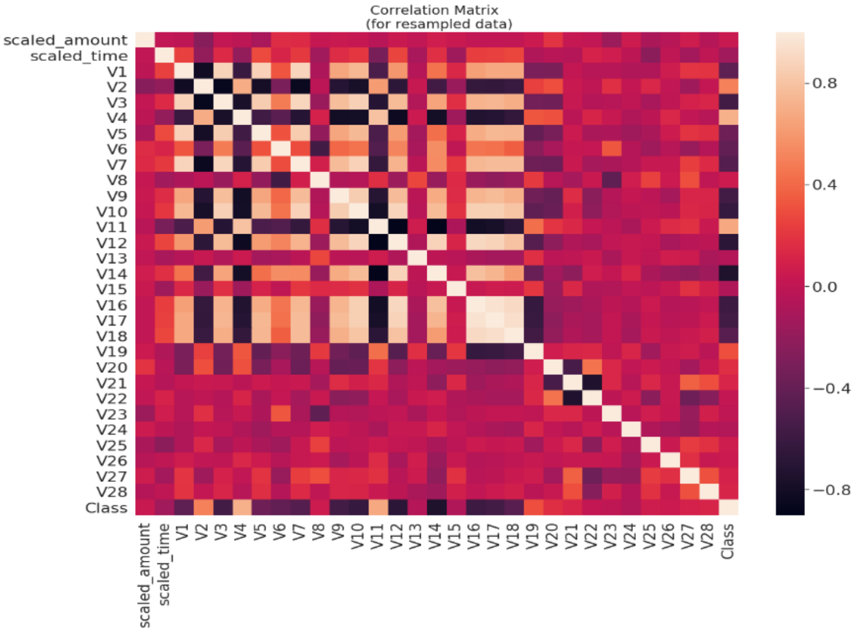


Figure 9. Correlation matrix for resampled data

Correlation matrix for resampled data can be represented by:

- *Features with Negative correlation:* Features $V_{10}, V_{12}, V_{14}, V_{17}$ are negatively correlated. This indicates that the lesser these values would be, the end outcome would be most likely a fraudulent transaction.
- *Features with Positive correlation:* Features V_2, V_{11}, V_4, V_{19} are positively correlated, i.e., the higher the values of these features, the more probable the end outcome would be a fraudulent transaction.

Furthermore, it can be observed that only a handful of the dataset features are correlated, i.e., the attribute “class” is independent of both the features “amount” and “time”. It is also clear from the below correlation matrix that the transaction class, whether genuine (class 0) and fraudulent (class 1), essentially depends on the PCA normalized attributes.

The model performance on the original dataset and the ADASYN resampled dataset are then compared. It can be observed that, there has been a significant improvement in the performance of classifier models, after applying ADASYN oversampling technique. The estimated results of the various classifier models on the original dataset before applying ADASYN are presented in Table 2. The estimated performance parameters of the various classifier models as applied to the ADASYN resampled dataset are presented in Table 3.

Table 2. Performance metrics estimated for various classifier models before applying ADASYN oversampling

Classifier models’ performance metrics	Decision tree	Random forest	KNN	Logistic regression
Accuracy	97.08%	99.98%	80.33%	97.10%
Precision	0.981	0.998	0.823	0.983
F1-score	0.942	0.999	0.801	0.943

Table 3. Performance metrics estimated for various classifier models after applying ADASYN oversampling

Classifier models’ performance metrics	Decision tree	Random forest	KNN (k = 9)	Logistic regression
Accuracy	100.0%	100.0%	83.33%	90.10%
Precision	1.0	1.0	0.99	0.91
Recall	1.0	1.0	0.833	0.89
F1-score	1.0	1.0	0.998	0.90

From Table 3, we employed KNN, which gives us an accuracy score of 83.33% on the data with the best value of k, which was auto evaluated by our model to be k=9. The second model, the logistic regression classifier, gives a better result as compared to that of the KNN algorithm with an overall accuracy score of 90.10%. The other two classifier models, i.e., the random forest classifier model, and the decision tree classifier model, gave an absolute 100.0% accuracy, which is an interesting estimation of accuracy, given the data set was highly imbalanced or skewed and it was resampled using one of the most prominent resampling techniques, ADASYN. Further, the precision, recall, and F1-score for decision tree and random forest are 1.0, unique. Also, the precision, recall and F1-score for KNN and logistic regression are 0.99, 0.833, 0.998 and 0.91, 0.89, 0.90, respectively.

Table 4. Performance of different classifiers for Prusti and Rath [13]

Performance Metrics	ELM	MLP	KNN	Random Forest	Ensemble Method
Accuracy	78.25	80.38	81.43	81.92	83.83
Precision	86.00	87.68	88.52	89.12	94.50
F1-score	86.63	88.03	88.84	89.22	90.31

We compare our simulated results with Prusti and Rath [13], we found that our simulated results outperformed all classifiers and ensembles used by them presented above in Table 4. Subsequently, the authors in the literature applied various standalone soft computing techniques to the pre-processed data to obtain accurate and precise classifications of the data into the said categories, namely the genuine class of credit transactions (class 0) and the fraudulent or deceptive class of credit card transactions, as described in the literature (class 1). A confusion matrix was used to estimate accuracy and other performance parameters showing the models would classify the data into class 0 or class 1. The objective of this study has been entirely dedicated to improving the model performance by varying and fine-tuning the model parameters that can help us render the best results. Given the data, if we can get such a good estimation concerning the model's performance in classifying the fraudulent transactions into either genuine or fraudulent classes, which is significantly better, there is no specific need of having heavier learning algorithms to be employed when the standalone models can give such a greater accuracy.

6. RESULTS AND DISCUSSIONS

This study analyzed numerous multifaceted and critical tasks that included detecting deceitful and fraudulent activities in the credit card in a relatively higher skewed (imbalanced) environment. We applied standalone soft computing and intelligent techniques such as the random forest, decision tree, KNN, and logistic regression classifier to accurately and predictively estimate the classifier's performance metrics to efficiently detect fraud and effectively tell them apart from the genuine credit card transactions and achieved significant improvements in the performance metrics of the classifier models. In the future, an open direction of the work would include working with data that represents the real-world scenario, i.e., the dataset is properly balanced out in terms of the minority class, the fraudulent credit card transactions. Also, improving the model's efficiency by enhancing the model's fitting parameters would be the primary objective. Here, even though our work has been limited to the dataset, PCA transformed numerical values, yet considering more generic scenario, it would be stimulating to extend our work with datasets having text value and sentimental statements.

ACKNOWLEDGEMENTS

This research work was funded by "Woosong University's Academic Research Funding-2021".

REFERENCES

[1] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37-48, 2008, doi: 10.1109/TDSC.2007.70228.

[2] S. S. Dhok, and G. R. Bamnote, "Credit card fraud detection using hidden Markov model," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 1, pp. 231-237, 2012.

[3] A. Prakash and C. Chandrasekar, "An optimized multiple semi-hidden markov model for credit card fraud detection," *Indian Journal of Science and Technology*, vol. 8, no. 2, pp. 176-182, Jan. 2015, doi: 10.17485/ijst/2015/v8i2/58081.

[4] V. Bhusari and S. Patil, "Application of hidden Markov model in credit card fraud detection," *International Journal of Distributed and Parallel Systems*, vol. 2, no. 6, pp. 203-211, Nov. 2011, doi: 10.5121/ijdps.2011.2618.

[5] N. B. Khandare, “Credit card fraud detection using hidden Markov model,” *International Journal of Advance Scientific Research and Engineering Trends*, vol. 1, no. 4, Jul. 2016.

[6] A. Singh and D. Narayan, “A survey on hidden markov model for credit card fraud detection,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 1, no. 3, pp. 49-52, 2012.

[7] S. Subudhi and S. Panigrahi, “Use of fuzzy clustering and support vector machine for detecting fraud in mobile telecommunication networks,” *International Journal of Security and Networks*, vol. 11, no. 1-2, pp. 3-11, 2016, doi: 10.1504/IJSN.2016.075069.

[8] B. Mehdi, C. Hasna, and O. Tayeb, “Intelligent credit scoring system using knowledge management,” *IAES International Journal of Artificial Intelligence (IJAI)*, vol. 8, no. 4, pp. 391-398, Dec. 2019, doi: 10.11591/ijai.v8.i4.pp391-398.

[9] A. Guha, “Prediction of Bankruptcy using Big Data Analytics based on Fuzzy c-means Algorithm,” *IAES International Journal of Artificial Intelligence (IJAI)*, vol. 8, no. 2, pp. 168-174, Jun. 2019, doi: 10.11591/ijai.v8.i2.pp168-174.

[10] M. A. Febriantono, S. H. Pramono, R. Rahmadwati, and G. Naghdy, “Classification of multiclass imbalanced data using cost-sensitive decision tree C5.0,” *IAES International Journal of Artificial Intelligence (IJAI)*, vol. 9, no. 1, pp. 65-72, Mar. 2020, doi: 10.11591/ijai.v9.i1.pp65-72.

[11] H. E. Mostafa and F. Benabbou, “A deep learning based technique for plagiarism detection: a comparative study,” *IAES International Journal of Artificial Intelligence (IJAI)*, vol. 9, no. 1, pp. 81-90, Mar. 2020, doi: 10.11591/ijai.v9.i1.pp81-90.

[12] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” *Decision support systems*, vol. 50, no. 3, pp. 559-569, 2011, doi: 10.1016/j.dss.2010.08.006.

[13] D. Prusti, and S. K. Rath, “Fraudulent Transaction Detection in Credit Card by Applying Ensemble Machine Learning techniques,” In *10th International Conference on Computing, Communication and Networking Technologies*, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944867.

[14] Machine Learning Group–ULB, Credit card Fraud Detection, Kaggle, 2018. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud.

[15] S. Ghosh, Reilly, “Credit card fraud detection with a neural-network,” In *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, 1994, pp. 621-630, doi: 10.1109/HICSS.1994.323314.

[16] F. Carcillo, Y. A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, “Combining unsupervised and supervised learning in credit card fraud detection,” *Information Sciences*, vol. 557, pp. 317-331, 2019, doi: 10.1016/j.ins.2019.05.042.

[17] F. Carcillo, Y. A. Le Borgne, O. Caelen., and G. Bontempi, “Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization,” *International Journal of Data Science and Analytics*, vol. 5, no. 4, pp. 285-300, 2018, doi: 0.1007/s41060-018-0116-z.

[18] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Feature engineering strategies for credit card fraud detection,” *Expert System with Applicatopns*, vol. 51, pp. 134-142, 2016, doi: 10.1016/j.eswa.2015.12.030.

[19] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” *Decision Support Systems*, vol. 50, no. 3, pp. 602-613, 2011, doi: 10.1016/j.dss.2010.08.008.

[20] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, “Credit card fraud detection: a realistic modeling and a novel learning strategy,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8. pp. 3784-3797, 2017, doi: 10.1109/TNNLS.2017.2736643.

[21] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P. E. Portier, L. He-Guelton, and O. Caelen, “Sequence classification for credit-card fraud detection,” *Expert Systems with Applications*, vol. 100, pp. 234-245, 2018, doi: 10.1016/j.eswa.2018.01.037.

[22] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, “Credit card fraud detection using Bayesian and neural networks,” In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, 2002, pp. 261-270.

[23] R. J. Bolton, and D. J Hand, “Unsupervised profiling methods for fraud detection,” *Credit scoring and credit control VII*, pp. 235-255, 2001.

[24] M. Zareapoor, K. R. Seeja, and M. A. Alam, “Analysis on credit card fraud detection techniques: based on certain design criteria,” *International journal of computer applications*, vol. 52, no. 3, pp. 35-42, 2012, doi: 10.5120/8184-1538.

[25] A. A. Akinyelu, and A. O. Adewumi, “Classification of phishing email using random forest machine learning technique,” *Journal of Applied Mathematics*, 2014, doi: 10.1155/2014/425731.

[26] S. Kiran, J. Guru, R. Kumar, N. Kumar, D. Katariya, and M. Sharma, “Credit card fraud detection using Naïve Bayes model based and KNN classifier,” *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 3, pp. 44-47, 2018.

[27] E. N. Osegi and E. F. Jumbo, “Comparative analysis of credit card fraud detection in simmulated annealing trained artificial neural network and hierarchical temporal memory,” *Machine Learning with Applications*, vol. 6, 2021, doi: 10.1016/j.mlwa.2021.100080.

[28] C. Nuno, G. Figueira, and M. Costa, “A data mining-based system for credit-card fraud detection in e-tail,” *Decision Support Systems*, vol. 95, pp. 91-101, 2017, doi: 10.1016/j.dss.2017.01.002.

[29] A. Colin, “Building Decision Trees with the ID3 Algorithm”, *Dr. Dobbs Journal*, 1996.

[30] H. He, Y. Bai, E. A Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.