FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 8 No. 4, August, 2024, pp 8 - 16
DOI: https://doi.org/10.33003/fjs-2024-0804-2435

AN ENHANCED FEATURE ENGINEERING TECHNIQUE FOR CREDIT CARD FRAUD DETECTION

[1]Hadiza Hassan, *[2]Muhammad Aminu Ahmad and [3]Rabi Mustapha

[1]Department of Informatics, Kaduna State University, Kaduna, Nigeria
[2]Department of Secure Computing, Kaduna State University, Kaduna, Nigeria
[3]Department of Intelligent Computing, Kaduna State University, Kaduna, Nigeria

*Corresponding authors' email: muhdaminu@kasu.edu.ng

ABSTRACT

As the world is becoming a cashless society with increasing use of online transactions, the number of credit cards users has also increased substantially. This led to credit card fraud, which is among the major cybercrimes faced by users with consequential damages to financial institutions. Therefore, credit card fraud detection is crucial due to the increasing number of credit card transactions. Machine learning based credit card fraud detection systems exist, but machine learning approaches have problems with imbalanced data and the need to selected best features for effective classification. Imbalance classification occurs when there are small number of observations of the minority class compared with the majority in a dataset. This study addresses the challenges of feature selection and data imbalance in credit card fraud detection through an enhanced feature engineering method. We propose a technique that uses wrapper to select the best features and mitigate data imbalance using a hybrid approach that combines SMOTE, random oversampling and under-sampling techniques. Five popular machine learning classifiers—Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine—are used with balanced and imbalanced datasets to evaluate the technique. The results show significant improvements in accuracy, precision, recall, F1-score, and Kappa score with the enhanced method. Specifically, and K Nearest Neighbor, Random Forest and Support Vector Machine achieve perfect accuracy with the balanced data.

Keywords: Machine learning, Sampling techniques, Credit card fraud detection. Classification

INTRODUCTION

Credit card fraud is an ongoing and increasing cause of significant loss in financial institutions. The COVID pandemic increased the use online transactions, which resulted in an increase in the number of online users. This increasing number of credit card use and cashless transactions resulted in a higher number of financial frauds, which necessitates the need for strong prevention and detection techniques (Murli, 2015). Therefore, efficient and real-time detection of fraudulent activities in the financial transactions is needed in order to mitigate the huge financial losses (Debachudamani et al., 2020).

Financial institutions are using machine learning based techniques to detect fraudulent activities in their transactions. There exist numerous machine learning techniques to detect credit card fraud, which can be mainly classified into supervised learning, unsupervised learning and reinforcement learning (Zareapoor, 2015). The evolution of numerous and different machine learning techniques, such as, classification and clustering, and their application in the fraud detection has shown the need for the use of such algorithms and techniques in detecting frauds of credit card transactions, although using machine learning often come with difficulties particularly when dealing imbalanced data (Alkhatib, 2021; Maikano, 2024).

Several machine learning models and techniques have been used by researchers to detect credit card fraud, such as classification (Tran & Dang, 2021; Ileberi et. al., 2021; Singh et. al., 2022) Bayesian Model (Akila & Reddy, 2018), Auto Encoder (Misra et al., 2019) Decision Tree and Fuzzy Logic (Askari & Hussain, 2020), Hidden Markov Model (Lucas et al., 2020) and Ensemble techniques (Carcillo et al., 2019). However, Yazici (2020) noted that the common problems of using machine learning techniques in detecting credit card fraud are imbalanced data, real-time detection and feature engineering method. The author further established that the imbalanced data problem occurs due to the higher number of benign transactions than fraudulent transaction in the data. This shows that the use of effective feature engineering methods is substantial as the features obtained in financial data are limited. Similarly, the author also stated that adapting the detection system to real time scenarios is a challenging task, since the number of credit card transaction in a limited time period is very high.

Ileberi et. al (2022) suggested that the use of effective pre-processing techniques has the potential to improve the performance of machine learning classifiers. In addition, Dornadula & Geetha (2019) and Varmedja et al., (2019) adopted SMOTE for class imbalance but there are various drawbacks, such as noise and probability of overlapping between the classes which eventually results in overfitting of the model. Moreover, the works of Tran & Dang (2021), Ileberi et. al. (2021) and Singh et. al. (2022) showed that using a combination of oversampling and under sampling techniques for data balancing enhance classification performance.

As observed from previous works, various experiments that were performed on the credit card fraud lack feature selection. Therefore, this work will utilize a Wrapper approach (El Aboudi & Benhlima, 2016) to select related and coherent features in the dataset for maximum performance. Moreover, data imbalance and heterogeneity of the credit card fraud dataset are one of the major problems affecting the models from yielding higher accuracy. Therefore, this work hybridizes the techniques of data balancing by combining SMOTE, oversampling and under-sampling approach. The improved feature engineering technique (Wrapper + Hybridized Sampling) has the potential to enhance the performance of credit card fraud detection.

The rest of this article is structured as follows. Section 2 describes the literature review. Section 3, demonstrated the techniques used in the credit card fraud detection system and

explains the implementation of the proposed technique and techniques used in machine learning performance evaluation methods. In Section 4 discuss the results and Section 5 we concluded
the article.

MATERIALS AND METHODS

Approach

The approach used comprises three main stages namely Data Collection, Data Preprocessing and Training and Classification ash shown in Figure 1.

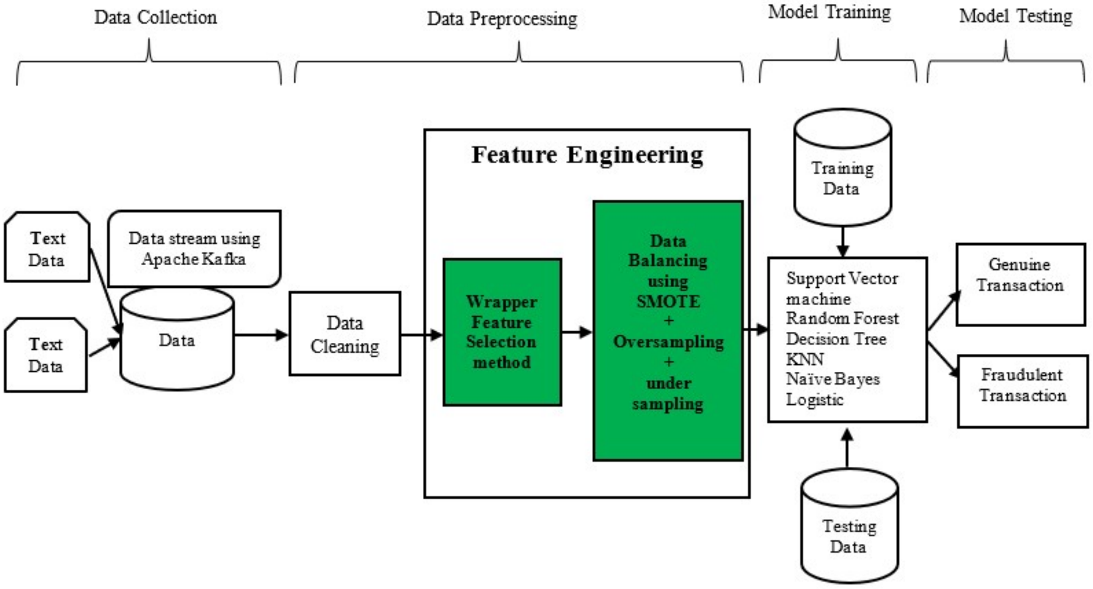

Figure 1: The enhanced credit card fraud detection technique

Data Collection

The research used an open dataset from Kaggle as source of data. The research used Apache Kafka for data streaming (Raptis & Passarella, 2023). Apache Kafka is an open-source software that is used to stream, process, read and analyze data.

Description of Data

The dataset used comprises credit card transactions information over a two-day period in September 2013 from cardholders in Europe with a total number of 284,807 transactions. There are 492 fraudulent transactions, which is approximately 0.172% of the dataset. This is a typical scenario of an imbalanced dataset.

The dataset comprises numerical input variables obtained through a Principal Components Analysis (PCS) transformation and anonymized as V1 through V17 for identities protection except two feature 'Time', 'Amount' and 'Class'. These features represent the number of seconds elapsed between the given transaction and the first transaction in the dataset, the transaction amount and fraudulent transaction/non-fraudulent transactions respectively. Some of the anonymized features include transaction ID, transaction date, transaction time, transaction day, terminal ID, customer id, mean amount, standard deviation of amount, mean number of transactions per day x coordinate of customer ID, y coordinate of customer ID and gender.

Data Preprocessing

At the data preprocessing stage, the collected credit card fraud dataset underwent data cleaning, normalization, and handling of missing values. A Wrapper approach, based on El Aboudi & Benhlima (2016), was employed to select the most relevant and coherent features from the dataset while eliminating any irrelevant data. This helps to improve the efficiency and effectiveness of the subsequent classification process. The selected features, such as transaction id, transaction date, transaction time, transaction day, terminal id, customer id, mean amount, standard deviation of amount, mean number of transactions per day, x coordinate of customer id and y coordinate of customer id were used for model training and evaluation.

The cleaned and selected data in the processed dataset was imbalanced, with a majority of normal transactions and a minority of fraudulent credit card transactions. To tackle the problem of an imbalanced dataset, the preprocessed wrapper data is subjected to a combination of Synthetic Minority Oversampling Technique (SMOTE) (Zhu, et al., 2024), oversampling and under-sampling techniques (Wongvorachan et al., 2023). This hybrid preprocessing technique helps to balance the dataset by generating synthetic instances of the minority class, resampling the majority class, and reducing the size of the majority class. The proposed model incorporates this hybrid preprocessing technique and applies it to train and test various classification algorithms.

Model Training and Classification

At this stage, the balanced dataset was split into training and testing sets. The training set accounted for 80% of the data, while the testing set comprised the remaining 20%. Five popular machine learning classifiers, SVM, KNN, Naïve-Bayes, Random Forest, and Decision Tree, were implemented and trained on the balanced training dataset. These algorithms had been widely used in credit card fraud detection research. The models learned from the training data and created a decision boundary to classify transactions as either fraudulent or normal. Furthermore, the trained models were evaluated using the testing dataset. Experiments were conducted with the imbalanced and balanced datasets. The classification accuracy of each model was recorded and analyzed with and without data balancing to determine the best-performing model based on the highest accuracy achieved. Other evaluation metrics, such as precision, recall, and F1-score, were also used to assess the performance of the models. The results obtained from the evaluation step were analyzed to determine the effect of the enhanced feature engineering technique on classification accuracy.

Evaluation

Evaluation experiments were conducted using the five selected machine learning algorithms in two forms. The first experiment was performed with the original dataset without preprocessing of feature selection and hybrid data balance approach while the second experiment was conducted with the wrapper feature selection and hybrid data balancing approach.

The experiment used confusion matrix to compared the performances of the algorithms used. A confusion matrix describes the performance of a classification model on set of test data for known true values in tabular form. In the context of this research, it provides valuable insights into an algorithm's performance, allowing for assessment of its ability to accurately classify transactions as fraudulent or benign. The rows of a confusion matrix represent the actual classes while its columns represent the predicted classes. Table 1 shows the confusion matrix for a two-class classifier (Valero-Carreras et. al., 2023).

Table 1: Confusion Matrix for Two Class Classifiers

ACTUAL		PREDICTED	
		Positive	Negative
	Positive	A (TP)	B (FN)
	Negative	C (FP)	D (TN)

TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative

After obtaining a confusion matrix for each of the machine learning models, the Accuracy, Sensitivity, Specificity Recall and Error rate values are derived from as follows;

***Accuracy**:* This is the ratio of number of benign transactions that are correctly classified to the total number of transactions. This is defined as shown in Equation 1.

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + FN + FP + TN}} \qquad (1)$$

Precision: This is the ratio of positively predicted transactions among the retrieved instances. This is defined as shown in Equation 2.

$$\text{Precision} = \frac{\text{TP}}{\text{TP +FP}} \qquad (2)$$

False Positive rate (FPR). This measures the rate of wrongly classified transactions. A low FP-rate signifies that the classifier is a good one. This is defined as shown in Equation 3.

$$\text{FPR} = \frac{\text{FP}}{\text{FP + TN}} \qquad (3)$$

True Positive Rate: This is the proportion of positives that are correctly identified. This is defined as shown in Equation 4.

$$TPR = \frac{TP}{TP + FN} \qquad (4)$$

Recall: This is the ratio of positively predicted instances among all the instances. This is defined as shown in Equation 5.

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \qquad (5)$$

F1-Score: This is a measure of the harmonic mean of precision and recall. This is defined as shown in Equation 6.

$$\text{F1} - \text{Score} = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Rec}} \qquad (6)$$

Kappa Score: It is a measure of agreement between the predicted and actual classes, taking into account the agreement that could occur by chance alone. This is defined as shown in Equation 7.

$$\text{kappa} = \frac{2*\text{TP}*\text{TN}-\text{FN}*\text{FP}}{(\text{TP}+\text{FP})*(\text{FP}+\text{TN})+(\text{TP}*\text{FN})*(\text{FN}*\text{TN})} \qquad (7)$$

Receiver Operating Characteristic (ROC) curve. This is a plot that shows true positive rate against the false positive rate.

RESULTS AND DISCUSSION

Feature Selection

The feature selection process was used to identify the features that exhibited strong correlation and contributed significantly to the classification and detection of fraud in the dataset. By selecting these relevant features, our goal was to improve the accuracy and efficiency of the classification models. We initially generated a heatmap to find features that are related to classes before using the wrapper algorithm to select the appropriate features. Figure 2 displays the inter-correlated features in the dataset, while Figure 3 reveals the selected features that exhibited strong correlation with the target variable (class).

After analyzing the visual representations in Figure 2, it is evident that the V14 and V17 features demonstrate a notable correlation with the target class. These specific features, V14 and V17, exhibit a stronger relationship with the Class variable in comparison to other features within the dataset. Seven features were ultimately selected for further analysis and model training out of the thirteen features initially considered.

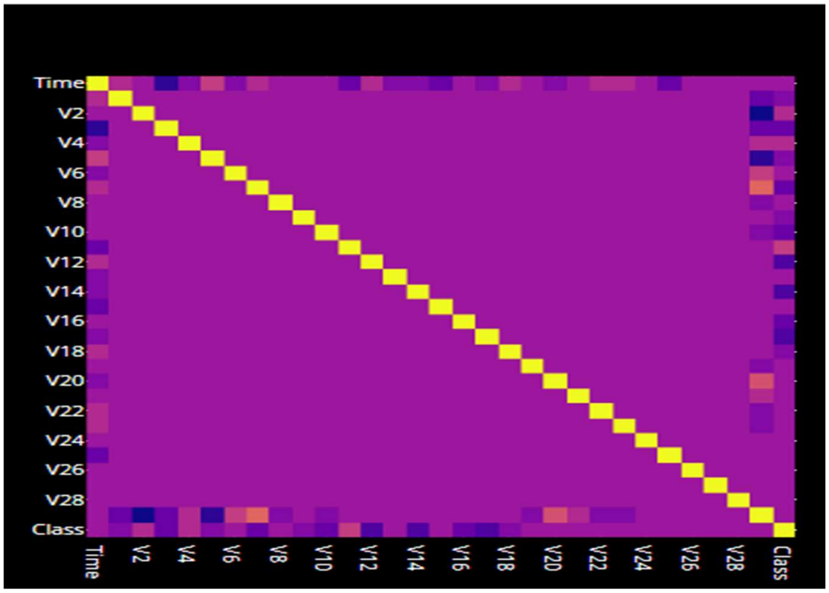

Figure 2: Inter-correlated features

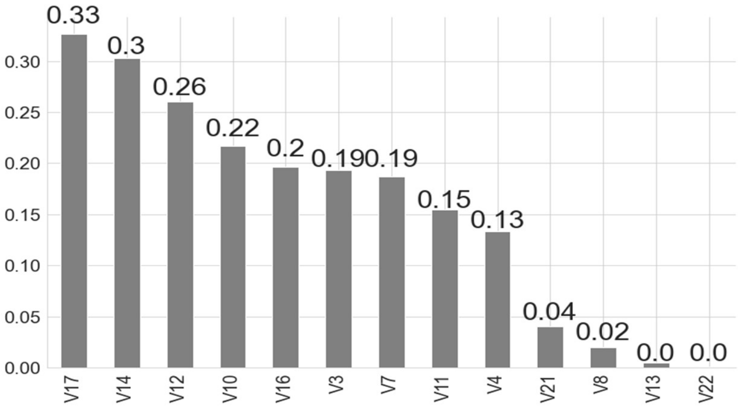

Figure 3: Features selected by the Feature selection engineering

Data Class Distribution

We examined the data class distribution and discovered that it was highly imbalanced. Figure 4 illustrates the significant disparity between the number of fraud and normal transactions. Obviously, the percentage distribution of fraud to normal transactions is highly incoherent. The class imbalance is challenging because it will impact the performance of the classification algorithms by prioritizing the majority class and neglecting the minority class. Therefore, SMOTE, random oversampling and under sampling were used to balance the credit card fraud detection data. The results obtained from each sampling technique (SMOTE, random oversampling then random under sampling) were combined to get a more effective class balance. This is to achieve a balanced representation of the fraud and non-fraud classes in the dataset and compare with the outcome of the imbalanced data analysis. By comparing the results after applying the hybrid balancing technique, it is easier to observe and analyze the differences and effects resulting from the application of data balancing, which will lead to improved performance of the classification models. Figure 5 illustrates the outcome of this combined approach that balanced the fraudulent and benign transactions. The result in Figure 5 shows that combining of SMOTE and oversampling and under sampling techniques has successfully resolve the data imbalance.

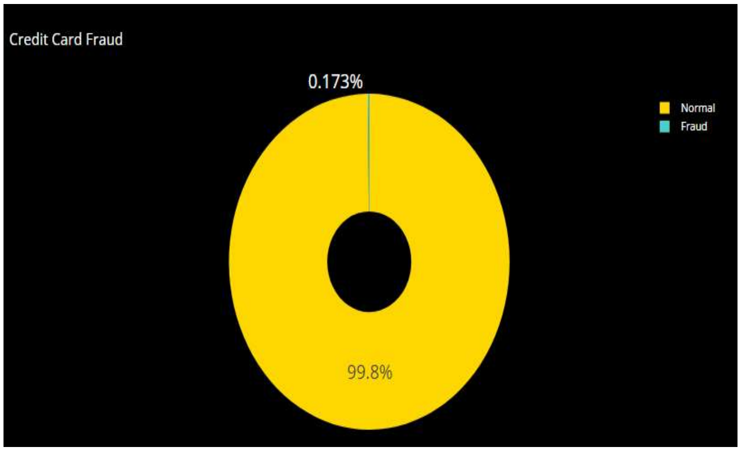

Figure 4: Data Class Distribution

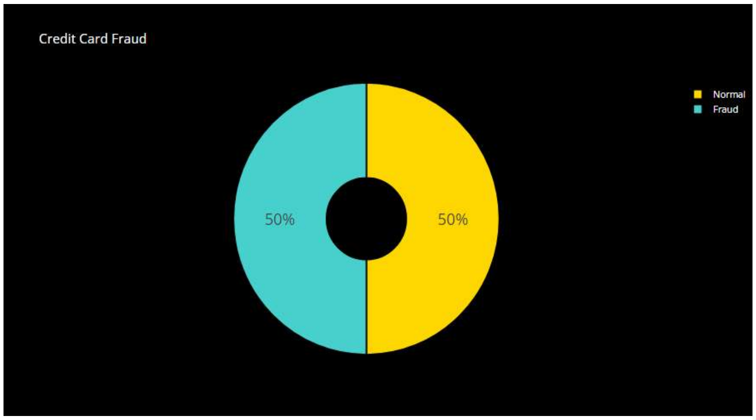

Figure 5: Balanced Dataset

Credit Card Fraud Classification

This section presents the results of the experiments conducted with the five selected machine learning algorithms with the balanced and imbalanced datasets. By balancing the dataset, we aimed to provide a fair representation of both the benign and fraudulent transactions, enabling the algorithms to make accurate predictions without being biased towards the majority class. Table 2 summarizes the results of confusion matrices for each of the five machine learning classifiers.

For the balanced data, the results shows that the algorithms predicted 85,289, 83,130, 84,999, 85,115 and 84,439 instances of the negative class (normal transactions) for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively. For the positive class (fraudulent transactions), the algorithms predicted 72,734, 71,015, 73,345, 69,414 and 77,347 instances for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively. On the other hand, the results show misclassifications of six, 2,165, 296, 180 and 856 normal transactions as fraudulent for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively. In addition, the algorithms also misclassified 12,561, 14,280, 11,950, 15,881 and 7,948 fraudulent transactions as normal for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively.

Table 2: Summary of Confusion Matrices Results for Credit Card Fraud Detection

Dataset	Classifier	True Positive	True Negative	False Positive	False Negative
Balanced	Random Forest	72,734	85,289	6	12,561
	Naïve Bayes	71,015	83,130	2,165	14,280
	K Nearest Neighbor	73,345	84,999	296	11,950
	Decision Tree	69,414	85,115	180	15,881
	Support Vector Machine	77,347	84,439	856	7,948
Imbalanced	Random Forest	106	86,291	4	42
	Naïve Bayes	121	83,399	1907	27
	K Nearest Neighbor	88	85,289	6	60
	Decision Tree	112	85,267	28	36
	Support Vector Machine	50	85280	15	98

The two categories of misclassifications highlight the limitations of the algorithms in accurately distinguishing between normal and fraudulent transactions. To sum up, the results show that Random Forest comparatively has the lowest number of misclassifications of normal transactions as fraudulent (False Positives), while Support Vector Machine has the lowest number of misclassified fraudulent transactions (False negative) for the balanced data.

For the imbalanced data, the results shows that the algorithms predicted 86,291, 83,399, 85,289, 85,267 and 85280 instances of the negative class (normal transactions) for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively. For the positive class (fraudulent transactions), the algorithms predicted 106, 121, 88, 112 and 50 instances for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively. On the other hand, the results show misclassifications of 4, 1907, 6, 28 and 15 normal transactions as fraudulent for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively. In addition, the algorithms also misclassified 42, 27, 60, 36 and 98 fraudulent transactions as normal for Random Forest, Naïve Bayes, K Nearest Neighbor, Decision Tree and Support Vector Machine respectively.

Table 3 summarizes the results obtained from the evaluation of the five classifiers after applying data balancing techniques and feature engineering for the balanced and imbalanced data. The performance metrics used are accuracy, precision, recall, F1-score and Kappa score. The summary provides an overview of the impact of the applied techniques on the classification models, showcasing the improvements in model performance achieved with and without data balancing.

Table 3: Classification Performance of the Five Algorithms Used

Dataset	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Kappa Score (%)
Balanced	Random Forest	0.93	0.94	0.93	0.93	0.85
	Naïve Bayes	0.90	0.91	0.90	0.90	0.81
	K Nearest Neighbor	0.93	0.94	0.93	0.93	0.86
	Decision Tree	0.91	0.92	0.91	0.91	0.81
	Support Vector Machine	0.95	0.95	0.95	0.95	0.90
Imbalanced	Random Forest	1.00	0.98	0.86	0.91	0.82
	Naïve Bayes	0.98	0.53	0.90	0.55	0.11
	K Nearest Neighbor	1.00	0.97	0.80	0.86	0.72
	Decision Tree	1.00	0.90	0.88	0.89	0.77
	Support Vector Machine	1.00	0.88	0.67	0.73	0.46

The classification results for the balanced data show that Random Forest achieved an accuracy of 93%, with a precision of 94%, recall and F1-score values of 93% each, and Kappa score 85%. The results also show that Naïve Bayes attained an accuracy of 90%, with a precision of 91%, recall and F1-score values of 90% each, and a Kappa score of 81%. The results further show that K Nearest Neighbor achieved an accuracy of 93%, with a precision of 94%, recall and F1-score values of 93% each and a Kappa score of 86%. Additionally, Decision Tree attained an accuracy of 91%, with a precision of 92%, recall and F1-score of 91% each, and a Kappa score of 81%. Finally, the result show that SVM achieved the highest accuracy of 95%, with precision, recall, and F1-score of 0.95 each, and a Kappa score of 90%.

From the results, it is evident that SVM performed the best among the classifiers, achieving best performance in terms of accuracy, precision, recall, and F1-score. Random Forest and K Nearest Neighbor also showed good performance with similar scores across the metrics. Naïve Bayes and Decision Tree had slightly lower scores but still achieved acceptable performance. Thus, these results demonstrate the effectiveness of the Random Forest algorithm in accurately classifying credit card transactions as normal or fraudulent, with high precision, recall rates and Kappa score.

Furthermore, the classification results for the imbalanced data show that Random Forest achieved an accuracy of 100%, with a precision of 98%, recall value of 86%, F1-score values of 91% and Kappa score 82%. The results also show that Naïve Bayes attained an accuracy of 98%, with a precision of 53%, recall value of 90% F1-score values of 55% and a Kappa score of 11%. The results further show that K Nearest Neighbor achieved an accuracy of 100%, with a precision of 97%, recall value of 80%, F1-score values of 86% and a Kappa score of 72%. Additionally, Decision Tree attained an accuracy of 100%, with a precision of 92%, recall value of 88%, F1-score of 89% each and a Kappa score of 77%. Finally, the result show that SVM achieved an accuracy of 95%, with a precision of 88%, recall of 67%, F1-score of 73% and a Kappa score of 46%.

Table 4 shows the differences in the performances of the classifiers between balanced and imbalanced data.

Table 4: Performance Differences of Classifiers Between Balanced and Imbalanced Data

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Kappa Score (%)
Random Forest	-0.07	-0.04	0.07	0.02	0.03
Naïve Bayes	-0.08	0.38	0	0.35	0.7
K Nearest Neighbor	-0.07	-0.03	0.13	0.07	0.14
Decision Tree	-0.09	0.02	0.03	0.02	0.04
Support Vector Machine	-0.05	0.07	0.28	0.22	0.44

A positive value shows increase in performance after using imbalanced data, while a negative indicates reduction in the performance. The results show general slight reduction in accuracies and increase in recall, F1-score and Kappa values of the classifiers except Naïve Bayes. In addition, there is reduction in the precision of Random Forest and K Nearest Neighbor and increase in the precision of Naïve Bayes, Decision Tree and Support Vector Machine.

Table 5 presents a comparative analysis of the performances of the proposed method and the work of Singh et. al. (2022) that use SMOTE and Tomek Link sampling techniques. The performance metrics used are accuracy, precision, recall and F1-score.

Table 5: Classification Performance of the Proposed Technique and Singh et. al (2022)

Dataset	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed Technique	Random Forest	0.93	0.94	0.93	0.93
	K Nearest Neighbor	0.93	0.94	0.93	0.93
	Decision Tree	0.91	0.92	0.91	0.91
	Support Vector Machine	0.95	0.95	0.95	0.95
Singh et. al. (2022)	Random Forest	1.00	1.00	1.00	0.99
	K Nearest Neighbor	0.99	1.00	1.00	0.99
	Decision Tree	1.00	0.99	0.99	0.99
	Support Vector Machine	0.95	0.98	0.92	0.95

Furthermore, Figure 6 shows the ROC curves for the five classification algorithms used. An ROC curve visually displays the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) as the classification threshold is varied. The closer the curve is to the top-left corner of the plot, the better the algorithm's performance in distinguishing between the positive and negative classes. A higher AUC value suggests better discriminatory power, with values closer to 1 indicating a stronger performance. Therefore, the ROC curves in Figure 6 provide a visual representation of the performances of five classifiers in distinguishing between fraudulent and normal transactions with balanced and imbalanced data.

For the balanced data, the ROC curve for Random Forest demonstrates a steep rise at the beginning, indicating a high true positive rate while maintaining a low false positive rate with highest area under curve that is, 0.99. The ROC curves for Support Vector Machine, Naive Bayes, K Nearest Neighbor and Decision Tree indicate reasonably good performances with a significant area under the curve, that is, 0.97, 0.96, 0.94, and 0.91 respectively.

The results suggest that the algorithms effectively identify a significant portion of fraudulent transactions while minimizing the number of false positives with Random Forest classifier has the highest value for area under curve. This shows that Random Forest classifier comparatively has the highest level of differentiating fraudulent and normal credit card transaction. In the case of imbalanced data, the Support Vector Machine demonstrates the highest AUC of 0.97, surpassing Naïve Bayes, Random Forest, Decision Tree, and K Nearest Neighbor (KNN) with AUC values of 0.96, 0.94, 0.88, and 0.87, respectively. Notably, the ROC scores obtained under imbalanced data conditions are observed to be lower compared to the achievements when the data were balanced.

Overall, the results show that using wrapper feature selection technique with the hybrid data balancing method has a positive impact on the performance of the machine learning algorithms, which enhanced the performance of credit card fraud detection because the recall, F1-score, Kappa-Score and AUC obtained by the enhanced technique were superior to values obtained with imbalanced dataset. This shows that appropriate use of feature selection technique and hybrid data balancing methods enhance the performance of credit card fraud classification.

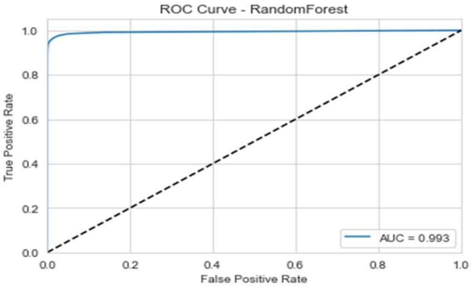

Figure 6.1: RF ROC (Balanced)

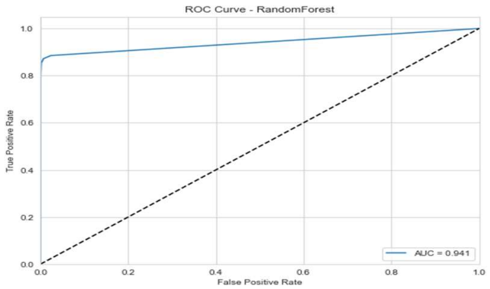

Figure 6.2: RF ROC (Imbalanced)

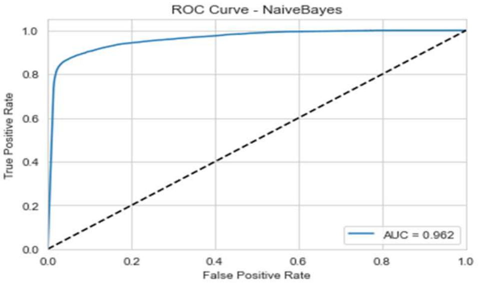

Figure 6.3: NB ROC (Balanced)

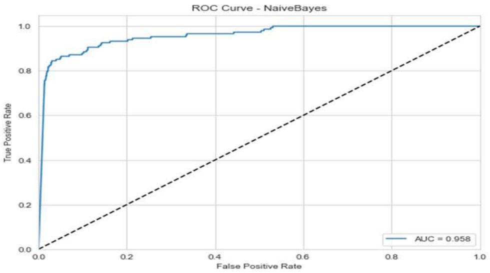

Figure 6.4: NB ROC (Imbalanced)

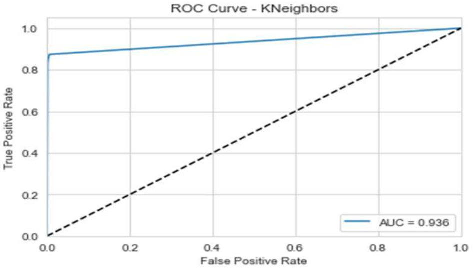

Figure 6.5: KNN ROC (Balanced)

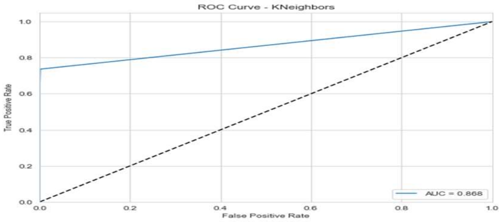

Figure 6.6: KNN ROC (Imbalanced)

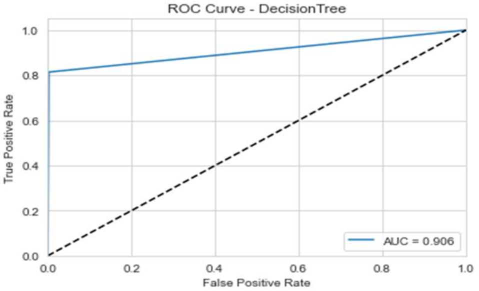

Figure 6.7: DT ROC (Balanced)

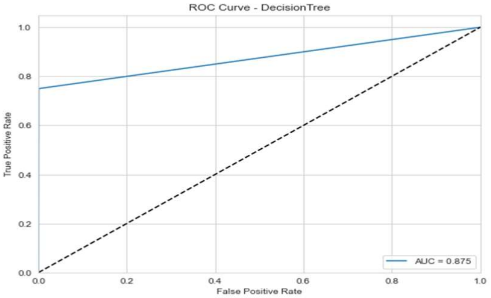

Figure 6.8: DT ROC (Imbalanced)

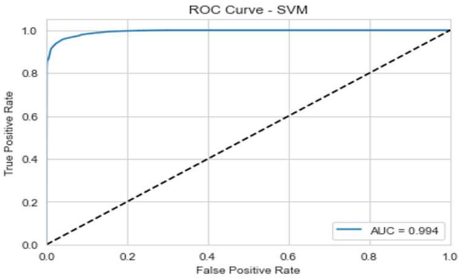

Figure 6.9: SVM ROC (Balanced)

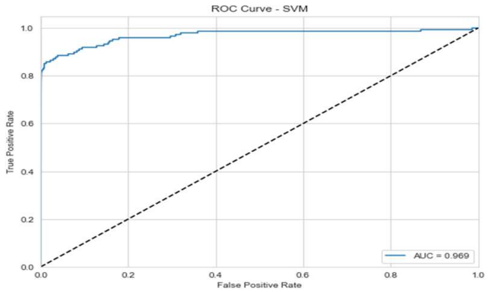

Figure 6.10: SVM ROC (Imbalanced)

Figure 6: ROC curves for the five classification algorithms

CONCLUSION

This research addressed the challenges of feature selection and data imbalance in credit card fraud detection. By employing a combination of wrapper feature selection and hybrid data balancing methods to evaluate the performance of different machine learning classification algorithms, we sought to enhance the accuracy and effectiveness of fraud detection models. Through the evaluation of five popular classification algorithms, namely Random Forest, Naive Bayes, K Nearest Neighbor, Decision Tree, and SVM, this work obtained valuable insights into their performance in detecting fraudulent transactions. The results demonstrated that the combination of feature engineering techniques, such as wrapper feature selection, and data balancing methods, such as SMOTE, Resample, and Under-sampling, significantly improved the classification models' accuracy and effectiveness. These techniques helped address the issue of data imbalance and enabled the models to better handle the detection of fraudulent transactions. Although Support Vector Machine demonstrated the highest accuracy, precision, recall, F1-score, and Kappa score, indicating its effectiveness in detecting credit card fraud, it is important to note that the choice of the most suitable algorithm depends on various factors, including the specific dataset, the desired balance between accuracy and computational efficiency, and the interpretability of the results. Moreover, the wrapper feature selection and hybrid data balancing methods demonstrated a good performance in comparison with the performance of SMOTE + Tomek Link (Singh et al., 2022), which shows overfitting, particularly with Random Forest, K Nearest Neighbor and Decision Tree.

FUTURE WORK

Based on the results and findings obtained, there are several potential areas for further work. The effectiveness of ensemble methods, such as stacking or boosting, can be investigated in improving the overall performance of the classification models. Ensemble methods have the potential to combine the strengths of multiple algorithms and enhance the accuracy and robustness of the fraud detection system. Additionally, incorporating anomaly detection methods can help identify unusual patterns or behaviors that deviate from normal transactions, providing an additional layer of security for detecting fraud. Investigating the explainability of the classification model is also important in order to understand and justify the decisions made by the models to gain the trust of stakeholders and regulatory bodies.

REFERENCES

Akila, S. & Reddy, U. S., 2018. Cost-Sensitive Risk Induced Bayesian Inference Bagging (RIBIB) for Credit Card Fraud Detection. *Journal of Computational Science,* Volume 27, pp. 247-254.

Alkhatib, K. I.-A. (2021). Credit Card Fraud Detection Based on Deep Neural Network Approach. 12th International Conference on Information and Communication Systems (ICICS) (pp. 153-156). IEEE.

Askari, S. M. S., & Hussain, M. A. (2020). IFDTC4. 5: Intuitionistic fuzzy logic based decision tree for E-transactional fraud detection. *Journal of Information Security and Applications*, *52*, 102469.

Carcillo, F. et al., 2019. Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection. *Information Sciences,* pp. 10-11.

Debachudamani Prusti, S. S. Harshini Padmanabhuni, Santanu Kumar Rath (2020) Safety, Security, and Reliability of Robotic Systems, 1st Edition, 2020, Imprint CRC Press. eBook ISBN 9781003031352.
Dornadula, V. N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithms. *Procedia computer science*, *165*, 631-641.

El Aboudi, N., & Benhlima, L. (2016, September). Review on wrapper feature selection approaches. In 2016 international conference on engineering & MIS (ICEMIS) (pp. 1-5). IEEE.

Maikano, F. A. (2024). Machine Learning Approaches for Cyber Bullying Detection In Hausa Language Social Media: A Comprehensive Review And Analysis. FUDMA Journal of Sciences, 8(3), 344-348.

Ileberi, E., Sun, Y., & Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. IEEE Access, 9, 165286-165294.

Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9(1), 24.

Kaggle (2024) [Online]. Available at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud [Accessed 2 6 2024]

Lucas, Y., Partier, P.-E., Laporte, L. & He, L., 2020. Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. *Future Generation Computer System,* Volume 167, pp. 393-402.

Misra, S., Thankur, S., Ghosh, M. & Saha, S. K., 2020. An Autoencoder Based Model for Detecting Fraudulent Credit Card Transaction. *Procedia Computer Science,* Volume 102, pp. 254-262.

Murli, D. J. (2015). Credit card fraud detection using neural networks. International Journal of Students' Research in Technology & Management, 2(2), 84-88.

Raptis, T. P., & Passarella, A. (2023). A survey on networked data streaming with Apache Kafka. IEEE access, Volume 11, pp. 85333 - 85350

Singh, A., Ranjan, R. K., & Tiwari, A. (2022). Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. Journal of Experimental & Theoretical Artificial Intelligence, 34(4), 571-598.

Tran, T. C., & Dang, T. K. (2021, January). Machine learning for prediction of imbalanced data: Credit fraud detection. In 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM) (pp. 1-7). IEEE.

Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the confusion matrix. Computers & Operations Research, 152, 106131.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019, March). Credit card fraud detection-machine learning methods. In 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-5). IEEE.

Wongvorachan, T., He, S., & Bulut, O. (2023). A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. Information, 14(1), 54.

Yazici, Y. (2020). Approaches to Fraud detection on credit card transactions using artificial
intelligence methods. *arXiv preprint arXiv:2007.14622*..

Zareapoor, M. &. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science, 48, 679-685.

Zhu, M., Zhang, Y., Gong, Y., Xu, C., & Xiang, Y. (2024). Enhancing Credit Card Fraud Detection A Neural Network and SMOTE Integrated Approach. arXiv preprint arXiv:2405.00026.