Credit Card Fraud Detection Using Machine Learning

Sabnam Kumari, Pooja Sapra, Divyanshu Pratap Singh, Heramb Sharma, Devesh Kajla, Rohit Gupta
Bhagwan Parshuram Institute of Technology, Delhi, India
poojasapra@bpitindia.com, dpsingh2292k@gmail.com, sharma.heramb28@gmail.com, thedeveshkajla@gmail.com, rohit.gupta17069@gmail.com

Abstract—The surge in e-commerce and digital payments has led to a corresponding rise in credit card fraud, presenting new challenges for financial institutions and consumers. Traditional fraud detection methods are proving inadequate in the face of increasingly sophisticated fraud schemes, highlighting the need for smarter, more adaptive solutions. This paper investigates the use of machine learning techniques, including logistic regression, random forest, and AdaBoost, in enhancing the detection of fraudulent credit card transactions. Logistic regression is recognized for its clarity and ease of interpretation, while random forest leverages ensemble learning to boost predictive performance, and AdaBoost improves accuracy by correcting prior classification errors. By examining transaction data, we assess these models based on key performance indicators such as accuracy, precision, recall, and computational demands. The findings demonstrate that combining multiple machine learning approaches can substantially improve fraud detection, providing financial institutions with a more reliable and efficient solution for real-time fraud prevention.

Keywords—optimization, metaheuristic optimization, local search, global search

I. INTRODUCTION

The continued rise in online retail, mobile payments, and digital financial transactions has made credit cards a ubiquitous part of daily life. However, this increasing reliance on credit cards has heightened the risk of fraud. The growing sophistication of credit card fraud schemes is a serious concern for both consumers and financial institutions, as criminals constantly seek out new vulnerabilities in digital payment systems.

In earlier days, fraud detection systems operated on simple, rule-based models that flagged unusual transactions—such as those involving large sums or conducted in unfamiliar locations. Although these rule-based approaches were effective for a time, they now struggle to keep up with today's rapidly evolving fraud tactics. Machine learning has emerged as a transformative tool, capable of processing massive amounts of transaction data, recognizing subtle patterns, and adapting to the ever-changing landscape of fraudulent activity in real time.

Credit card fraud has emerged as a significant threat to the financial ecosystem, with conventional detection mechanisms often proving inadequate against modern tactics. Static, rule-based systems are no longer sufficient to identify the increasingly complex strategies fraudsters use. These systems often suffer from high rates of false positives, causing inconvenience to legitimate customers, and false negatives, allowing fraudulent transactions to slip through undetected.

In the fast-paced digital economy, financial institutions must implement fraud detection systems that can quickly and accurately identify fraudulent activity without negatively impacting the customer experience. While machine learning models offer a promising solution to these challenges, determining which models offer the best trade-offs between speed, accuracy, and user impact is critical. Thus, comparing the performance of multiple machine learning models is essential to developing a more robust and efficient fraud detection system.

The objective of this study is to implement and evaluate three machine learning models—logistic regression, random forest, and AdaBoost—for credit card fraud detection, each offering different strengths in performance and interpretability.

- Logistic regression is widely used for binary classification problems, such as distinguishing between fraudulent and legitimate transactions. It provides transparency by showing how different variables influence the probability of fraud.

- Random forest enhances predictive accuracy by constructing an ensemble of decision trees, which can capture complex, non-linear relationships in the data that simpler models may overlook.
- AdaBoost is an ensemble technique that improves weak classifiers by iteratively focusing on previously misclassified cases, offering increased accuracy in detecting fraud.

This research compares these models based on metrics such as precision, recall, and accuracy, with a focus on reducing both false positives and false negatives. The aim is to identify the most effective model, or a combination of models, for use in a real-time fraud detection system.

The structure of the paper is as follows: Section 2 reviews existing literature on fraud detection, emphasizing both traditional methods and the integration of machine learning. Section 3 outlines the methodology used in this study, detailing data preprocessing and the development of the logistic regression, random forest, and AdaBoost models. Section 4 presents the results, comparing the models using various performance metrics. Finally, Section 5 concludes the study with a discussion of the findings and potential areas for future research.

II. LITERATURE REVIEW

A. Traditional Methods for Detecting Fraud

Over time, the way we catch credit card fraud has changed. Early systems were simple and based on rules set by businesses. These rules flagged transactions that seemed unusual, like purchases over a certain amount or those made in a strange location. Although easy to use, these systems struggled to detect more complicated fraud. They also made a lot of mistakes, wrongly flagging normal transactions as suspicious, which frustrated customers when their legitimate payments were blocked.

B. Growing Use of Smart Technology in Fraud Detection

The rise of machine learning has made fraud detection smarter. These methods can look at large amounts of past data to find patterns and decide if a new transaction is safe or suspicious. Some of the most common types of machine learning used in fraud detection include:

- Logistic Regression: A basic model that predicts whether a transaction is fraudulent based on certain characteristics. It's simple and easy to interpret.
- Decision Trees: A model that sorts data by breaking it down into smaller parts to help decide if a transaction is suspicious. It's easy to understand but can sometimes make mistakes.
- Random Forest: This model combines several decision trees to get more accurate results and avoid errors.
- Neural Networks: These models, inspired by the human brain, can detect very complex patterns but require a lot of resources and fine-tuning to work well

C. Problems in Detecting Credit Card Fraud

There are several issues that make credit card fraud detection difficult:

1. Imbalance in Suspicious Cases: Fraud is rare compared to regular transactions, which can make it hard for systems to learn how to catch fraud accurately.

2. Changing Cheating Tactics: Criminals keep finding new ways to cheat, so detection systems

must be updated often to keep up with new tricks.

3. Real-time Detection: Fraud systems need to catch suspicious activity as it happens, which means they must be fast and accurate, often making decisions in milliseconds.

4. Protecting Personal Information: Since credit card data is sensitive, it’s important to ensure that these detection systems keep customer information secure while still being effective at spotting fraud].

III. METHEDOLOGY

- Data Overview: The dataset used for this study comes from Kaggle and includes credit card transactions from September 2013. It contains records for two days, with a total of 284,807 transactions, of which only 492 were found to be fraudulent. This means that fraud makes up just 0.172% of all transactions. To keep the data confidential, some information was converted into numerical values using a method called PCA. However, the "Time" and "Amount" features could not be changed this way. The "Time" feature shows how many seconds have passed since the first transaction, while the "Amount" feature indicates how much money was involved in each transaction. Another important feature, labeled "Class," shows whether a transaction is fraudulent or not: a value of 1 indicates a fraud, while 0 means it is a normal transaction. To evaluate how well our methods work, we look at several measures: accuracy, precision, recall, and F1-score. We also create a confusion matrix, which is a 2x2 table that helps visualize the results. The confusion matrix shows four important outcomes:
    - True Positive Rate (TPR): The number of fraudulent transactions correctly identified as fraudulent.
    - True Negative Rate (TNR): The number of legitimate transactions correctly identified as legitimate.
    - False Positive Rate (FPR): The number of legitimate transactions wrongly identified as fraudulent.
    - False Negative Rate (FNR): The number of fraudulent transactions wrongly identified as legitimate.

By plotting TPR against FPR at different levels, we can create a Receiver Operating Characteristic (ROC) curve, which helps evaluate the performance of our fraud detection model. The area under this curve (AUC) provides a summary measure of the model's ability to distinguish between fraud and non-fraud transactions.

- Data Preparation: Choosing the right features, or characteristics, from the dataset is crucial. This process is called feature selection, and it helps identify the most relevant variables while removing those that are less important. By selecting appropriate features, we can reduce errors, improve accuracy, and save time during training. To help with this selection, we used a tool designed by Will Koehrsen. This tool showed us which features were the most important, and we removed any that did not contribute significantly to the overall importance. After this step, we ended up with 27 features for further experiments. Since the dataset has many more normal transactions than fraudulent ones, it is imbalanced. This imbalance makes it hard for models to learn effectively. To fix this, we need to adjust the data so that both types of transactions are more evenly represented. Common methods include reducing the number of normal transactions (undersampling) or increasing the number of fraudulent ones (oversampling). We used a technique called

SMOTE (Synthetic Minority Oversampling Technique), which helps improve results by creating additional examples of the minority class.

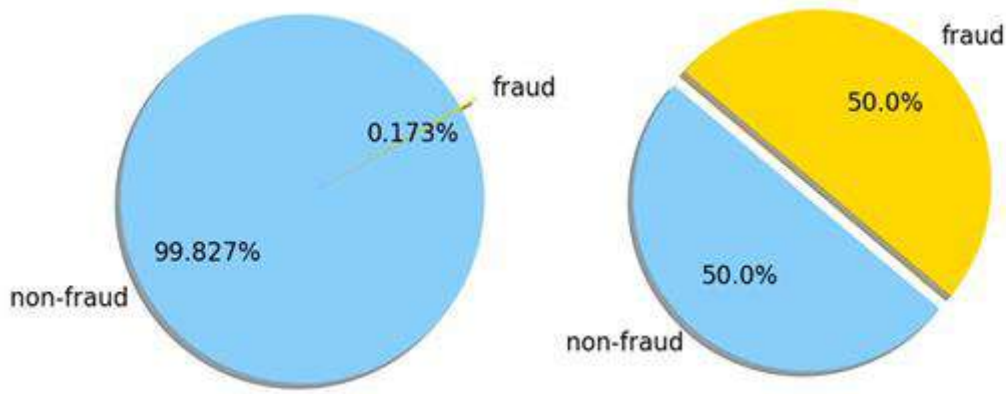


Fig. 1 Class distribution before and after sampling

Many machine-learning algorithms expect the scale of the input. Taking into account that values of time and amount are highly varying, scaling is done in order to bring all features to the same level of magnitudes.

- Experiment: Logistic regression is a popular method for classifying data. It helps us understand the relationship between various factors to predict whether an event will happen. It calculates the likelihood of a transaction belonging to each category based on certain features.

  Naive Bayes is another method we used, which assumes that the features are independent of each other. It relies on Bayes' theorem and can handle different types of data distributions. In this case, we used it to identify fraudulent transactions based on the Bernoulli distribution.

  Random Forest is a method that combines multiple decision trees to make predictions. It generally provides better results, especially when there are more trees in the forest, which helps prevent mistakes.

  We also used a type of artificial neural network (ANN) called a multilayer perceptron. This network has at least three layers: an input layer, hidden layers, and an output layer. Each layer uses a function to decide which connections to keep. In our experiment, we set up the network with four hidden layers containing 50, 30, 30, and 50 units, respectively, using the ReLU activation function. Research shows that deeper networks often perform better than shallower ones, so we started with fewer layers and gradually increased them until we found a suitable architecture.

  To optimize our model, we used an algorithm called Adam. We split the data into training and testing sets in an 80:20 ratio. The model was trained over several cycles until the improvement slowed down significantly, at which point we considered it ready and stopped training.

```
              Time         V1        V2         V3  ...       V26       V27       V28  Amount
100623    67571.0  -0.758469 -0.045410  -0.168438  ... -0.540955  0.150606 -0.117140  549.06
115599    73929.0   1.056257 -0.154476   0.092850  ...  0.582508 -0.123331  0.012196  103.92
54037     46262.0   1.160374 -0.330942  -0.125541  ... -0.540922  0.035393  0.016274   47.37
100435    67514.0   1.254421  0.340401   0.301932  ...  0.094824 -0.024252  0.030651    2.69
120585    75880.0   1.217646  0.649606  -0.466480  ... -0.323737  0.029514  0.043507    1.00
...           ...        ...       ...        ...  ...       ...       ...       ...     ...
209700   137667.0  -0.102716  0.478136   0.055105  ... -0.828485  0.124807  0.122492   45.95
123267    76866.0   1.165077  0.182248   0.586474  ...  0.136185 -0.039259  0.026811   35.37
223618   143456.0  -2.006582  3.676577  -5.463811  ...  0.536905  0.485864 -0.042393    1.00
43061     41353.0 -15.020981  8.075240 -16.298091  ...  0.056031 -1.310888 -0.707403   34.12
209277   137487.0  -1.250341 -2.446004  -2.273570  ... -0.526747  0.115233  0.113342  405.98

[787 rows x 30 columns]
```

Fig. 2 Train and Split Data

IV. RESULTS and DISCUSSIONS

- Classifier Models Overview: In this study, we developed three types of models to classify data: Naive Bayes, k-nearest neighbor (kNN), and logistic regression. To test how well these models work, we used 80% of the dataset for training and kept 20% for testing. We measured their performance using various metrics, including accuracy, sensitivity, specificity, precision, Matthews correlation coefficient (MCC), and balanced classification rate. The accuracy results for the original dataset distribution (0.172% fraud and 99.828% non-fraud), as well as the adjusted distributions (10% fraud and 90% non-fraud, and 34% fraud and 66% non-fraud), are shown in Tables 1, 2, and 3.
    - Performance Comparison: The evaluation of the three models using the 34:66 data distribution is shown in Figure 1. This distribution gave the best overall results. Among the models, the k-nearest neighbor (kNN) method consistently performed better than the others in different tests. It achieved the highest rates for correctly identifying legitimate transactions (specificity) and correctly identifying fraudulent transactions (precision), both at 1.0, meaning it did not misclassify any normal transactions as fraud.

    In contrast, the Naive Bayes model only outperformed kNN in terms of accuracy with the 10:90 data distribution. The logistic regression model had the lowest overall performance among the three. However, it is important to mention that all models showed significant improvement when we compared the two adjusted distributions.

    Since not all previous studies assessed models using the same measures—like accuracy, sensitivity, specificity, precision, MCC, and balanced classification rate—this study also compares its results with other research mainly using true positive rate (TPR) and false positive rate (FPR). Figures 2 and 3 show the TPR and FPR comparisons between our Naive Bayes, kNN, and logistic regression models and those from other studies, with references indicated by numbers in brackets "[ ]".

    True Positive Rate and False Positive Rate Evaluation

    - TPR = True Positive Rate
    - FPR = False Positive Rate
    - Proposed NB = Proposed Naive Bayes classifier
    - Proposed kNN = Proposed k-nearest neighbor classifier
    - Proposed LR = Proposed Logistic Regression classifier

    The proposed kNN model achieved zero false positives for both data distributions (10:90 and 34:66), showing better performance than the other studies we reviewed. When looking at the true positive and false positive rates for the logistic regression model, as shown in Figure 4, we noticed that the two rates were very similar for the 10:90 distribution. This is different from what we saw in Figures 2 and 3, where there was no overlap. These results suggest that the logistic regression model does better with the original dataset than with the two adjusted datasets.

TABLE 1. Accuracy result for un-sampled data distribution

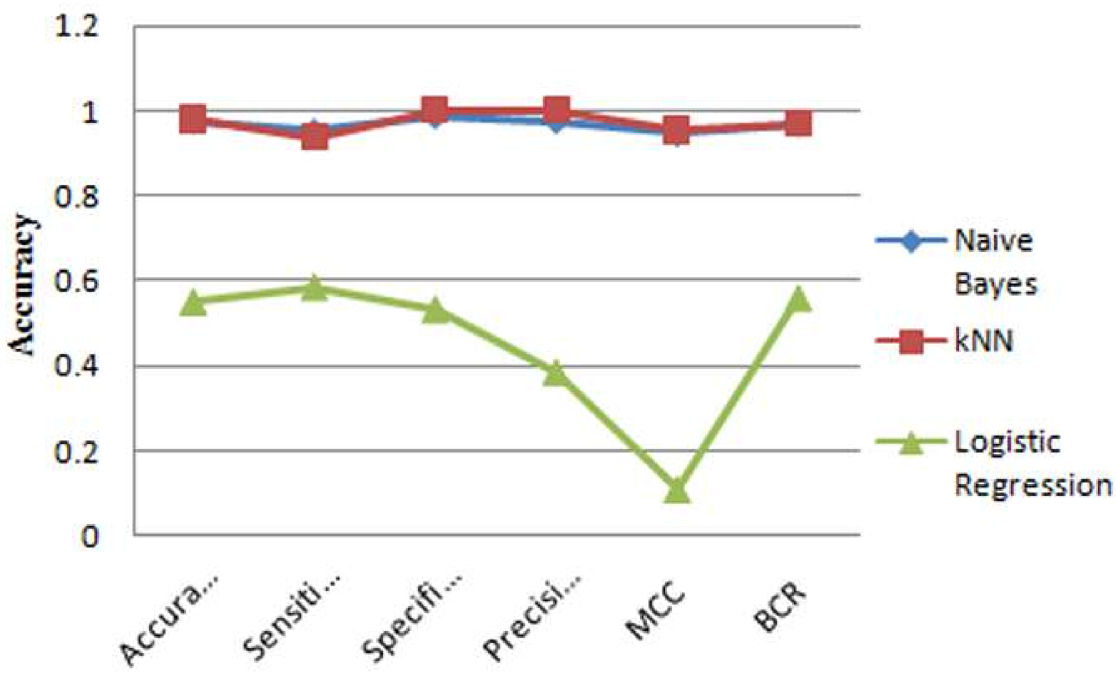
Metrics	**Classifiers**		
	Naïve Bayes	***k-Nearest Neighbour***	***Logistic Regression***
Accuracy	0.9737	0.9691	0.9824
Sensitivity	0.8072	0.8835	0.9767
Specificity	0.9741	0.9711	0.9824
Precision	0.0505	0.4104	0.0873
Matthews Correlation Coefficient	+0.1979	+0.5903	+0.2893
Balanced Classification Rate	0.8907	0.9273	0.9796

TABLE 2. Accuracy result for 10:90 data distribution

Metrics	**Classifiers**		
	Naïve Bayes	***k-Nearest Neighbour***	***Logistic Regression***
Accuracy	0.9752	0.9715	0.3639
Sensitivity	0.8210	0.8285	0.7155
Specificity	0.9754	1.0000	0.2939
Precision	0.0546	1.0000	0.1678
Matthews Correlation Coefficient	+0.2080	+0.8950	+0.0077
Balanced Classification Rate	0.8975	0.9143	0.5047

TABLE 3. Accuracy result for 34:66 data distribution

Metrics	**Classifiers**		
	Naïve Bayes	***k-Nearest Neighbour***	***Logistic Regression***
Accuracy	0.9769	0.9792	0.5486
Sensitivity	0.9514	0.9375	0.5833
Specificity	0.9896	1.0000	0.5313
Precision	0.9786	1.0000	0.3836
Matthews Correlation Coefficient	+0.9478	+0.9535	+0.1080
Balanced Classification Rate	0.9705	0.9688	0.5573



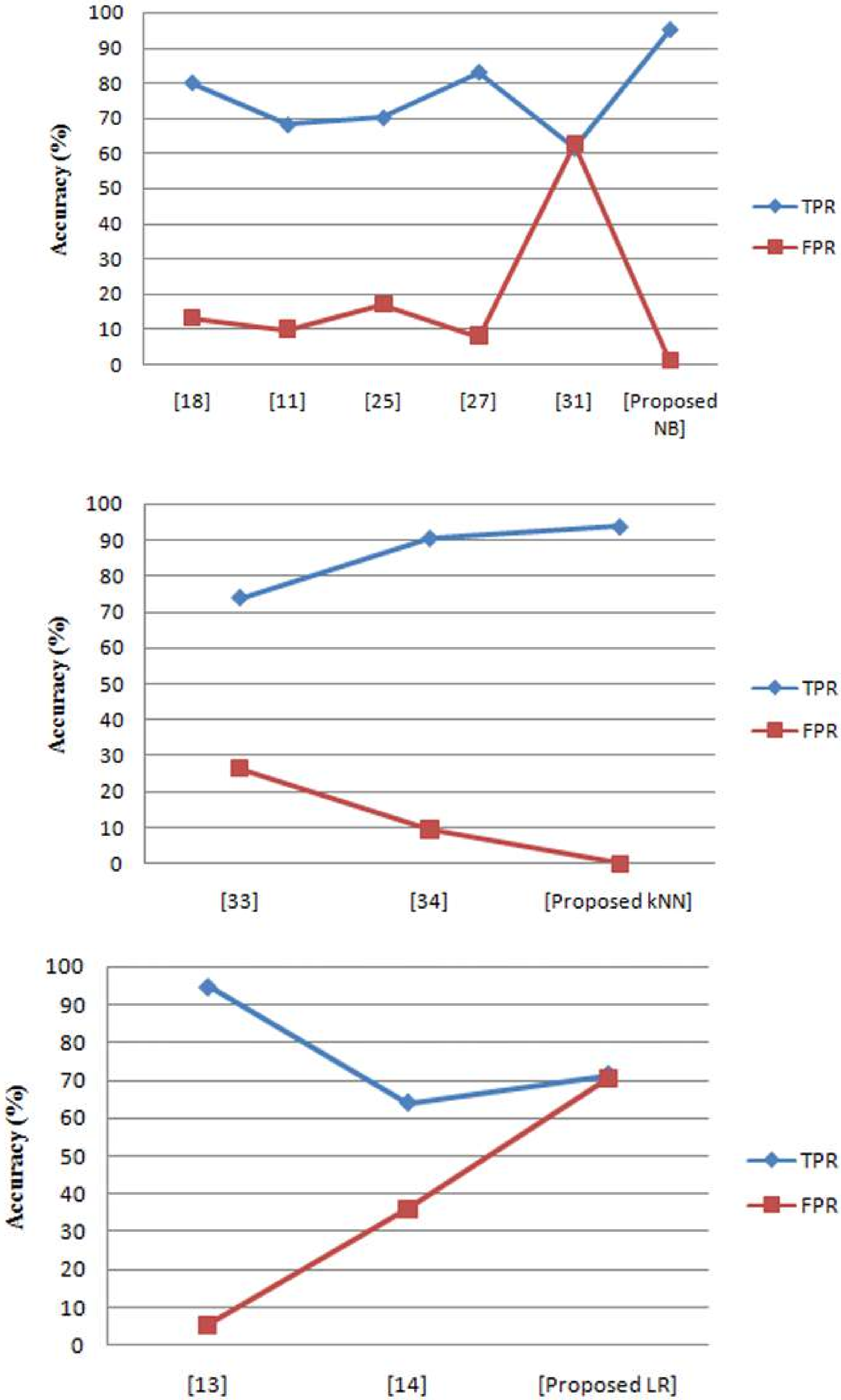


Fig. 3 Performance Analysis

V. CONCLUSION

Machine learning algorithms have shown significant promise in identifying fraudulent credit card transactions, with logistic regression performing well in terms of reliability, precision, and recall. However, fraud detection remains challenging due to the uneven distribution between legitimate and fraudulent transactions, as fraudulent cases are relatively rare. Balancing this disparity has enhanced the models' ability to recognize patterns within the limited fraud data, improving their effectiveness in detecting suspicious activities. In the future, more advanced methods like Convolutional Neural Networks (CNNs) or Reinforcement Learning could further enhance accuracy and efficiency. Integrating these models into real-time systems will be critical for quickly flagging suspicious transactions, reducing losses, and strengthening security

REFERENCES

1. Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10), 4915-4928.
2. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
3. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
4. Liu, Y., & Zhou, H. (2019). Deep learning and credit card fraud detection. *Journal of Finance and Data Science,* 5(2), 74-81.
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357