RESEARCH ARTICLE

DOI: 10.34256/ijcci2511

International Journal of Computer Communication and Informatics

Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms

Gopinath Krishnaraj [a,*], Chandru Ravi [a], Mohammed Bilal Althaf Ahmed [a]

[a] Department of Computer Applications, Sona College of Arts and Science College, Salem-636005, Tamil Nadu, India

* Corresponding Author: vengatgopinath@gmail.com

Received: 27-11-2024, Revised: 15-03-2025, Accepted: 21-03-2025, Published: 03-04-2025

Abstract: Credit card fraud is a major concern for both financial institutions and consumers, leading to significant financial losses and a decline in trust. With the rise in online transactions and increasingly sophisticated fraudulent schemes, there is a pressing need for strong and effective fraud detection systems. This research explores how machine learning and deep learning algorithms, particularly Random Forest (RF) and K-Nearest Neighbors (KNN), can be applied to detect credit card fraud. The main goal is to assess and compare how well these algorithms perform in accurately spotting fraudulent transactions while keeping false positives to a minimum. To carry out this research, we use a publicly available dataset of credit card transactions, which is marked by an imbalanced class distribution, where fraudulent transactions are far fewer than legitimate ones. We apply various preprocessing techniques, such as data cleaning, feature scaling, and addressing class imbalance through resampling methods like SMOTE (Synthetic Minority Over-sampling Technique), to improve data quality and model performance. Random Forest is a powerful ensemble learning method that uses a collection of decision trees to boost prediction accuracy and cut down on overfitting. K-Nearest Neighbors (KNN) is a straightforward, instance-based learning algorithm that classifies transactions by looking at the majority class of their k-nearest neighbours in the feature space. To evaluate how well both algorithms perform, we look at various metrics like precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). The findings show that Random Forest typically outshines K-Nearest Neighbors in overall accuracy and F1-score, especially when dealing with imbalanced datasets. This research emphasizes the need to tackle class imbalance and choose the right evaluation metrics for effective fraud detection.

Keywords: Credit card, Random Forest, K-Nearest Neighbors, Fine-tuning, AUC-ROC, Precision, Recall, F1-score.

1. Introduction

In this Credit card fraud is generally threat to financial institutions because it accounts for billions of lost dollars annually, and in an increasingly digital world where most transactions are performed online, banks, credit-card firms, and merchants would need real-time detection of these fraudulent transactions. Conventional approaches to fraud detection rely on pre-defined rules and visual inspections, both of which drag the detection process and are rarely effective with significant errors because of inefficiencies. Greater need in automation, which leads the company's scale, will arise since both transactions in volume and the level of complexity continue rising.

ML/DL algorithms have shown incredibly power in countering those difficulties as they autonomously learn the patterns on extensive data and predict what follows thereafter from the pattern identified, mainly based on history of its occurrences. A few famous and extremely popular algorithms for their results which deal with the data sets unbalancedness and accuracy on noisy data, or so-called 'Noisy' data are the Random Forest, and the popularly known algorithm K nearest Neighbor.

This research will utilize the latest machine learning techniques in fraud detection using Random Forest and KNN to analyse fraudulent credit card transactions. Main goals include improved metrics for fraud detection systems, reduced false positives, and a scalable solution that may be offered to financial institutions. This paper will test and compare these algorithms as to their performance and outline how they might be utilized in real-world fraud-detection scenarios.

2. Literature Survey

The potential of supervised learning techniques to classify transactions based on historical data [1]. Applied Bayesian networks to model probabilistic relationships in transaction data. Effective for small datasets but struggled with scalability [2]. Compared decision trees and SVM for fraud detection, emphasizing feature selection and handling imbalanced datasets [3]. Highlighted the effectiveness of Random Forests and cost-sensitive learning for fraud detection in imbalanced datasets [4]. Use of gradient boosting algorithms like XGBoost for improved classification accuracy [5]. The use of autoencoders to identify anomalies in transaction data [6]. GANs Proposed to generate synthetic data for addressing class imbalance, enhancing model performance [7]. RNNs is used to detect sequential and temporal patterns in transaction data [8]. Real-time fraud detection system developed for using Apache Spark, integrating machine learning models for high-speed analysis [9]. Online learning algorithms introduction to adapt models dynamically to evolving fraud patterns [10].

3. Proposed Methodology

3.1 Data Collection

The Kaggle Credit Card Fraud Detection dataset is the standard benchmark dataset used to assess and improve fraud detection models. The dataset consists of 284,807 credit card transactions by European cardholders during two days. Of all the records, only 492 have been tagged as fraud - a tiny number, which makes up about 0.172% of the total data, which is typical in many actual fraud detection challenges. It consists of 28 anonymized variables from PCA, for privacy considerations, besides two other features: Time that denotes in terms of seconds how far each transaction is between that transaction and the first. The amount denotes the amount of the transaction. Class: target; the Class is a binary target hence for fraud, 1 while for genuine 0. This dataset is publicly available in Kaggle under the name "Credit Card Fraud Detection" and has been widely used for testing purposes on machine learning and deep learning algorithms to address the challenges like imbalanced data and identification of the fraudulent pattern. This creates a more realistic environment for researchers in designing and optimizing fraud detection systems.

3.2 Data Pre-processing and Feature Engineering

Data preprocessing is all about making sure that your dataset is tidy and ready to go for machine learning models like Random Forest and K-Nearest Neighbors. Missing values are addressed and normalized to enhance distance-based calculations for KNN. Class imbalance is managed using SMOTE to make the fraudulent and legitimate transactions equal. Feature engineering selects key attributes such as transaction time, amount, and PCA transformed features. Surveys show that pre-processing really boosts the performance of Random Forest by improving how decision trees split, and it also fine-tunes KNN by making the distance metrics more precise, especially when it comes to detecting fraud in imbalanced datasets.

3.3 Model Selection and Evaluation

Random Forest and KNN have been chosen for credit card fraud detection due to their comparative strengths. The ensemble learning method of Random Forest will be preferred because it is robust for complex datasets and can process high-dimensional data. In contrast, KNN is simple to use and identifies patterns based on proximity in feature space. Accuracy, precision, recall, F1-score, and Prediction time would be used for model evaluation. Random Forest performs better in imbalanced datasets due to its feature importance mechanism. KNN is more sensitive to scaling and parameters. Cross-validation is a crucial aspect that validates such models. Generally, Random Forest is expected to perform better than KNN for fraud detection, as it is flexible and an ensemble model.

3.4 Real-Time Prediction and Deployment

In real-time fraud detection, the two models are used for prediction purposes on new transactions. The Random Forest model is really good for real-time prediction since it infers very fast and can work with very large datasets and multiple features. The KNN is simple, but computationally costly in real-time as it must calculate distances for every transaction. This must be optimized for deployment so that KD-Trees and Ball Trees are used, which can speed up nearest-neighbor searches. Both models combine into a real-time system deployed against updated and retrained models capable of learning new fraud patterns in real time.

3.5 System Flow Diagram

Figure 1 summarizes the flow of data and operations in the fraud detection system.

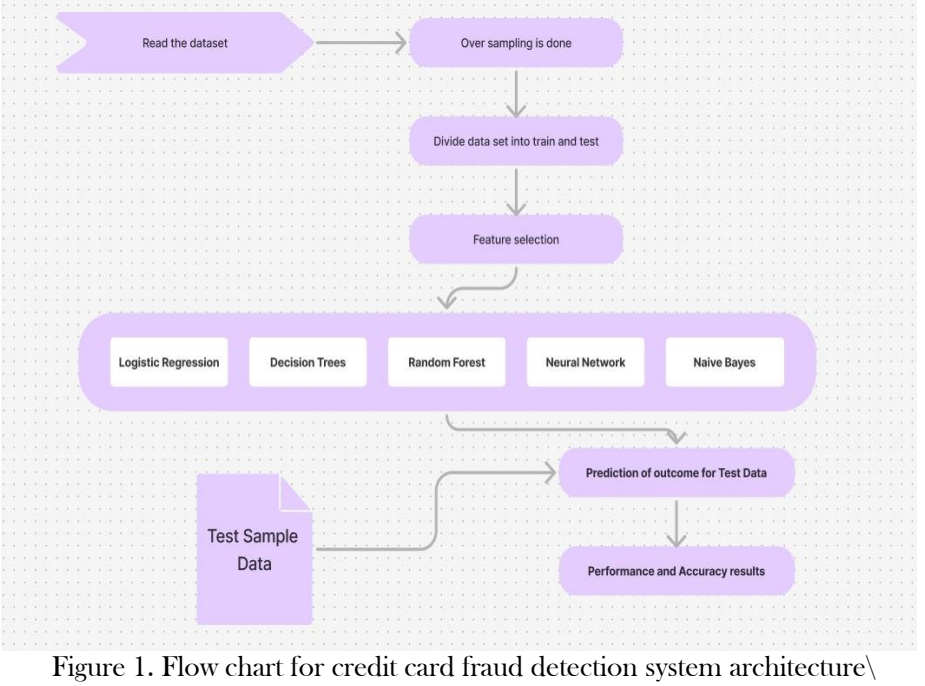

Figure 1. Flow chart for credit card fraud detection system architecture\

3.6 Implementation

The pre-processing of the data is the starting point for the implementation of the credit card fraud detection system. The raw data set-for example, the Kaggle Credit Card Fraud Detection Data Set-is cleansed with regard to handling missing and inconsistent values, normalization of the features, with this being pivotal for the algorithm based on the KNN that will be based on distance computations. SMOTE, which is the acronym for Synthetic Minority

Oversampling Technique, is applied to fraud detection for balancing the class in an over-underclass imbalance by generating artificial samples for the under-class.

For KNN, it is configuration includes setting k, number of neighbour's, while making use of a distance metric for classifying transactions- Euclidean distance and then majority class dictates the label as it compares every transaction to its K nearest neighbors. The efficient but computationally expensive thus would require optimizations such as KD trees for large data sets.

Random Forest are trained an ensemble of decision trees on bootstrapped subsets of the data. For the final prediction, each tree classification will be considered using a majority vote. The mechanism to calculate feature importance is robust, with this approach being an overfitting-resistant method but also performs very well with imbalanced datasets. Metrics like accuracy, precision, recall, and Prediction time are used to evaluate the models for accurate fraud detection.

4. Methods

4.1 Random Forest

Random Forest is an ensemble learning algorithm that builds multiple decision trees and combines their predictions to improve metrics and reduce overfitting. It works by aggregating the results of individual trees through majority voting for classification tasks or averaging for regression tasks. Each tree is trained on a random subset of the dataset and a random subset of features, which enhances generalization.

Random Forest Prediction

For classification, the final prediction is determined by majority voting among T trees.

$$\boldsymbol{y = mode(f1(x), f2(x), \ldots, fT(x))} \quad (1)$$

Where

- *fi(x):Prediction from the i-th tree.*
- *y:Final prediction.*
- *T:Total number of trees in the forest.*

This approach is particularly effective for high-dimensional or imbalanced datasets.

For regression

$$\hat{y} = \frac{1}{T}\sum_{i=1}^{T} f_i(x) \quad (2)$$

Random Forest excels in handling high-dimensional data and is less prone to overfitting compared to individual decision trees.

4.2 K-Nearest Neighbour's (KNN)

KNN is a simple, instance-based learning algorithm that classifies a new data point based on the majority class of its k-nearest neighbors in the feature space. It uses a distance metric (e.g., Euclidean distance) to find the closest neighbors.

The Euclidean distance d*(x, x')* between two points *x* and *x'* is given by:

$$d\ (x, x^{/}) = \sqrt{\sum_{i=1}^{n}(x_i - x_i^{/})^{2}} \qquad (3)$$

Where

- *xi, x'i: Values of the i-th feature for points x and x'.*
- *n:Number of features.*

For regression

$$\hat{y} = \frac{1}{T}\sum_{i=1}^{K} y_i \quad (4)$$

KNN is a versatile and easy-to-implement algorithm that performs well on small datasets with clear patterns. However, it is computationally expensive for large datasets and sensitive to irrelevant features. Proper scaling, careful selection of K and feature optimization can improve its performance for real-world applications.

4.3 Comparison of Random Forest and KNN

Random Forest (RF) and K-Nearest Neighbors (KNN) are two popularly used machine learning algorithms, both differing in their characteristics. Random Forest is an ensemble-based learning method, which involves training multiple decision trees and aggregates the predictions from all of them. It is more robust against overfitting, fast during prediction, as the trees are pre-trained, and efficient for high-dimensional and large datasets. Additionally, Random Forest provides the feature importance score, and hence it is easier to understand the significance of the input variables. Hyperparameters such as the number of trees and tree depth can be adjusted to obtain optimal performance, making it highly reliable for imbalanced datasets.

On the other hand, KNN is an instance-based learning algorithm that classifies the data based on proximity to its nearest neighbors. Although simple and intuitive, KNN requires distance calculations for each prediction, which makes it computationally expensive and slower, particularly on large datasets. It does not provide feature importance scores like Random Forest, and it is prone to overfitting if the number of neighbors, denoted by (K), is small. Proper tuning of (K) and balancing of data are very important to improve its performance. Overall, Random

Forest is to be preferred for complex and large-scale problems, while KNN works best on the smaller datasets with clear patterns.

5. Evaluation and Continuous Learning

Evaluation: Both KNN and Random Forest models are evaluated using metrics like accuracy, precision, recall, F1-score, and Prediction time to assess their performance in detecting fraud as shown in Table 1. Random Forest tends to perform better on imbalanced datasets due to its feature importance mechanism, while KNN is sensitive to feature scaling and parameter tuning.

Continuous Learning: In machine learning, therefore, ensures that models like Random Forest and KNN maintain their effectiveness over time as fraud patterns are changing. Periodic retraining of models with new transaction data allows them to adapt to emerging trends and anomalies. Random Forest handles incremental learning in an efficient manner by updating trees; while KNN requires the recomputation of distances with updated datasets. Automated pipelines for the collection, pre-processing, and retraining of data allow models to maintain accuracy and reliability in real-time systems for fraud detection.

Merits: KNN is simple to implement and intuitive, making it suitable for small to medium-sized datasets. It does not require explicit training, as it makes predictions based on stored data. KNN works well with non-linear data distributions and can adapt to new patterns without retraining. Additionally, it is versatile, as it can be used for both classification and regression tasks, providing flexibility in solving various problems.

Random Forest is robust to overfitting due to its ensemble learning nature, making it highly effective for large, complex datasets. It automatically handles missing values and performs well on imbalanced datasets by assigning feature importance. Random Forest can handle both categorical and continuous variables, making it a versatile algorithm for different data types. It also provides reliable performance and good generalization, even when the dataset contains noise

Demerits: KNN can be computationally expensive, especially for large datasets, as it requires calculating distances for each new data point. It is sensitive to irrelevant or redundant features, and its performance heavily relies on the choice of the distance metric and k-value. KNN can struggle with high-dimensional data, as the "curse of dimensionality" increases the complexity and reduces its effectiveness.

Random Forest, while robust, can be computationally intensive during training, especially with large numbers of trees and features. It is less interpretable compared to simpler models like decision trees, making it harder to explain the reasoning behind predictions. Additionally, Random Forest may require significant memory and time for large-scale data, making real-time prediction challenging.

6. Result

I. **Dataset Size:** 100,000 transactions (98,000 legitimate, 2,000 fraudulent).

II. **Features:** Transaction Amount, Time, Location, Merchant ID, etc.

III. **Random Forest:** Trained with 100 trees and max depth of 10.

IV. **KNN:** Trained with k = 5 (5 nearest neighbour's) using Euclidean distance.

Table 1. Performance metrics

Metric	Random forest	KNN
Accuracy	99.2%	96.5%
Precision	92.8%	87.1%
Recall	89.5%	81.3%
F1-Score	91.1%	84.1%
Prediction Time	0.01 seconds	1.2 seconds

6.1 Bar Chart

Figure 2 shows the bar chart comparing the performance of Random Forest and KNN for various metrics.

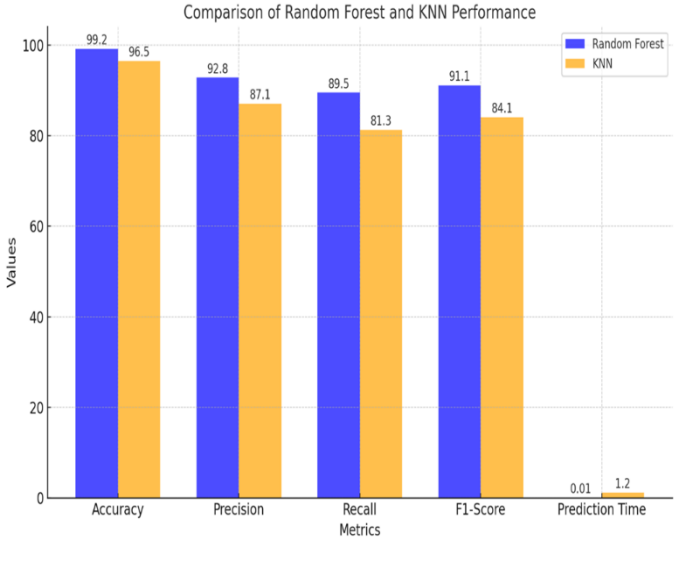

Figure 2. Comparison bar chart for both Random Forest and KNN performance

7. Conclusion

KNN and Random Forest are both good algorithms for detecting credit card fraud; the latter one has more robustness and accuracy, Precision, Recall, F1-Score & Prediction time especially when applied to imbalanced datasets. Although KNN is intuitive and easy to understand, it has high sensitivity toward tuning and pre-processing. The two models benefit from continuous learning and periodic retraining for better performance.

References

[1] S. Ghosh, D. Reilly, (1994). Credit Card Fraud Detection with Neural Networks. *Proceedings of the 27th awaii International Conference on System Sciences*, IEEE, USA. https://doi.org/10.1109/HICSS.1994.323314

[2] K.J. Ezawa, S.W. Norton, (1996). Constructing Bayesian networks to predict uncollectible telecommunications accounts. *IEEE Expert*, *11*(5), 45-51.

[3] Y. Sahin, E. Duman, (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. *Proceedings of the International Journal of Advanced Computer Science and Applications,* IEEE, Turkey. https://doi.org/10.1109/INISTA.2011.5946108

[4] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, "Credit card fraud detection and concept-drift adaptation with delayed supervised information," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-8, doi: http://doi.org/10.1109/IJCNN.2015.7280527

[5] C. Whitrow, D.J. Hand, P. Juszczak, D. Weston, N.M. Adams, (2009). Transaction aggregation as a strategy for credit card fraud detection. Data mining and knowledge discovery, 18, 30-55. https://doi.org/10.1007/s10618-008-0116-z

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

[7] C. Yu, Y. Xu, J. Cao, Y. Zhang, Y. Jin, M. Zhu, Credit card fraud detection using advanced transformer model. In 2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom) (pp. 343-350). IEEE. https://doi.org/10.1109/MetaCom62920.2024.00064

[8] A. Madhavi, T. Sivaramireddy, (2021). Real-Time Credit Card Fraud Detection Using Spark Framework. In: Mai, C.K., Reddy, A.B., Raju, K.S. (eds) Machine Learning Technologies and Applications. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-33-4046-6_28

Funding

No funding was received for conducting this study.

Conflict of interest

The Author have no conflicts of interest to declare that they are relevant to the content of this article.

About The License