International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Credit Card Fraud Detection - A Machine Learning Perspective

Shahana Fathima[1], Leena C Sekhar[2], Jaseena K U[3]

[1]Department of Computer Applications, MES College, Marampally, Aluva, Kerala, India
Email: *shahanafathima820[at]gmail.com*

[2]Department of Computer Applications, MES College, Marampally, Aluva, Kerala, India
Email: *leena[at]mesmarampally. org,*

[3]Department of Computer Applications, MES College, Marampally, Aluva, Kerala, India
Email: *jaseena[at]mesmarampally.org*

Abstract: ***The increasing prevalence of credit card fraud in today's digital economy poses a significant challenge to financial institutions and consumers alike. To combat this threat, there is a growing need for robust and efficient fraud detection systems. This paper presents a comprehensive machine learning approach for credit card fraud detection, leveraging advanced techniques and models to enhance the accuracy and reliability of fraud detection mechanisms. Our methodology encompasses data pre - processing, feature engineering, and model selection to construct a highly effective fraud detection pipeline. We explore various machine learning algorithms, including K Nearest Neighbour, Support Vector Machine, Random Forest, Decision Trees and Artificial Neural Networks, to build a predictive model that can distinguish between legitimate and fraudulent credit card transactions. The dataset used for training and evaluation is sourced from historical credit card transaction records, encompassing a wide range of transaction attributes and labels for fraudulent and non - fraudulent activities. We apply rigorous performance metrics, such as precision, recall, F1 - score, to assess the models' efficacy. The proposed model achieves high accuracy rates while minimizing false positives, thus enhancing the overall security of credit card transactions.***

Keywords: Credit Card Fraud, Fraud Detection, Feature Selection, RandomForest, K - Nearest Neighbours, Artificial Neural Network.

1. Introduction

The digital age has brought remarkable convenience to our lives, allowing us to conduct financial transactions with the click of a button. Credit cards, in particular, have become an essential part of modern commerce, enabling seamless payments and online shopping. However, this convenience comes with a significant challenge - the rise of credit card fraud. As technology evolves, so do the tactics of fraudsters, making it imperative for financial institutions to stay one step ahead in the battle against fraudulent activities. Detecting and preventing such fraud necessitates the development of an efficient model. Credit card fraud detection falls within the realm of machine learning, aiming to mitigate various forms of fraudulent activities. This paper introduces key features to enhance fraud detection accuracy and speed using diverse classifiers. This study employs machine learning, including K - Nearest Neighbours, Artificial Neural Networks, Support Vector Machines, Random Forests, and Decision Trees, to identify and prevent fraudulent transactions.

Credit card fraud typically involves unauthorized use of another person's card information, including PINs, passwords, and credentials, with or without the physical card. Detecting and preventing such fraud is crucial, and machine learning and deep learning - based fraud detection modules have proven very effective in this regard. Machine learning represents a potent technology that empowers computers to enhance their performance through experiential learning, devoid of the need for explicit programming. Deep learning, a subset within the realm of machine learning, leverages neural networks to emulate the data processing and decision - making capacities of the human brain. A multitude of deep learning techniques, including artificial neural networks, convolutional neural networks, autoencoders, recurrent neural networks, and restricted Boltzmann machines, play a significant role in enhancing fraud detection systems.

2. Related Works

Ileberi, E. et al. [1] proposed a credit card fraud detection engine using a genetic algorithm (GA) for feature selection and after feature selection a combination of machine learning classifiers such as Random Forest, Decision Trees, Artificial Neural Networks, Naive Bayes, and Logistic Regression. The genetic algorithm was utilized on a dataset of credit card transactions from European cardholders, resulting in a remarkable accuracy of 99.98% when employing the GA - RF model. Asha R. B et al [2] used multiple machine learning algorithms to identify the occurrence of fraud. The results demonstrate that ANN achieves close to 100% accuracy, outperforming other techniques and highlighting the potential of deep learning for fraud detection. E. N Osegi et al [3] introduced a novel approach to credit card fraud detection using Hierarchical Temporal Memory based on Cortical Learning Algorithms (HTM - CLA). The study conducts a comparison between HTM - CLA and Self - Adaptive Artificial Neural Networks alongside Long Short - Term Memory Artificial Neural Networks. The results indicate that HTM - CLA outperforms both SA - ANN and LSTM - ANN by a significant margin, showcasing its potential for improving credit card fraud detection. A hybrid approach of genetic algorithm with a Multilayer Neural Network (MNN) trained with spectral

clustering for credit card fraud detection is proposed by Ojugo et al [4]. The hybrid model achieves a sensitivity of 90% and specificity of 19%, demonstrating its effectiveness in distinguishing legitimate from fraudulent transactions.

Fayas Ittoo et al [5] proposed a comparative analysis of various machine learning algorithms such as Logistic Regression (LR), Naive Bayes (NB), and k - Nearest Neighbor (KNN) across various data proportions. It is observed that the LR based model is more accurate for the prediction of fraudulent. Olowookere, T. A., & Adewale, O. S [6] proposes a framework that combines meta - learning ensemble techniques and cost - sensitive learning for fraud detection. Through the integration of cost - sensitive learning within the ensemble learning process, the model attains exceptional Area Under the Receiver Operating Characteristic curve (AUC) values and maintains consistent performance across varying fraud rates within the dataset. Saad M. Darwish [7] introduces an intelligent system architecture based on the Artificial Bee Colony (ABC) optimization method and k - means for fraud detection in online payment systems. The architecture includes a rule engine to filter the dataset and combines semantic K - means and ABC algorithm fusion, resulting in improved fraud detection performance. V N Dornadula and Geetha [8] proposed a method to design and develop a novel fraud detection for Streaming Transaction Data to analyse the past transaction details of the customers and extract the behavioural patterns. Various classifiers are applied and the classifier with better rating score is selected as one of the best methods to predict fraud.

Pumsirirat, A., & Liu, Y. [9]create a model of deep Auto - encoder and restricted Boltzmann machine (RBM) that can reconstruct normal transactions to find anomalies from normal patterns. Deep learning based on autoencoders (AE) is an unsupervised learning algorithm that utilizes backpropagation by configuring the input to be equivalent to the output. [9]. Awoyemi et al [10] addresses the issue of imbalanced datasets in credit card fraud detection by applying a hybrid technique of under - sampling and over - sampling on skewed data. n the study, the performance of Naive Bayes, K - nearest Neighbor, and Logistic Regression models is compared, with K - nearest Neighbor demonstrating superior performance over the other models.

In summary, these studies highlight the importance of using advanced ML and AI techniques for credit card fraud detection. Researchers have explored a range of approaches, including genetic algorithms, deep learning, cost - sensitive learning, and hybrid models, to enhance the accuracy and effectiveness of fraud detection systems. These approaches hold promise for improving security in online payment systems and safeguarding financial transactions from fraudulent activities.

The paper's remaining structure unfolds as follows: Section 3 provides an overview of the dataset, and Section 4 delves into the methods utilized. Section 5 gives various evaluation parameters, section 6 explains the proposed framework, section 7, the experimental set up and the conclusion and future scope in section 8.

3. Data Set Description

The dataset encompasses a total of 492 instances of fraudulent transactions out of 284, 807 transactions. To safeguard confidentiality, the dataset represents transaction details in the form of numeric values, categorized as positive and non - positive. The dataset comprises 31 distinct features, identified by labels V1 through V28. Additionally, two key features are disclosed: "Time," signifying the elapsed time in seconds since the initiation of the first transaction on day 1, and "amount, " which exhibits positive values for deposits and non - positive values for withdrawals. Figure 1 shows the screenshot of the dataset employed in this study.

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791
1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336
-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136
-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775
-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819
-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653
1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199
-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822
-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765
-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677
1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344
0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985
1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779
1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805
-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265
-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999
1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591
-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047

Figure 1: Screenshot of the dataset

4. Methods Employed

In pursuit of the research goals, an extensive array of machine learning algorithms has been utilized in this investigation, encompassing K - Nearest Neighbors, Random Forest, Support Vector Machines, Decision Trees, and Artificial Neural Networks.

4.1 K - Nearest Neighbours Algorithm (KNN)

The K - Nearest Neighbours algorithm, commonly referred to as KNN or k - NN, is a supervised learning classifier that relies on proximity to make classifications or predictions. Although it can be employed for both classification and regression tasks, it is predominantly used as a classification algorithm. This preference stems from its core principle, which assumes that similar data points are likely to be found close to one another. For classification problems, KNN selects a class label based on a majority vote, where the label most frequently occurring near a particular data point is assigned. Majority voting typically implies a majority exceeding 50%, which may not be necessary when dealing with multiple classes. Notably, KNN is a non - parametric technique, making no underlying assumptions about the data distribution. It is often referred to as a "lazy learner" algorithm, as it retains the training dataset rather than immediately learning from it

The KNN classifier operates by calculating the distance between an unknown data pattern and all the data patterns within the training dataset to identify the nearest neighbours. This operation is divided into two primary components:

a) Calculating Distance: The distance between the unknown pattern and each data pattern in the training dataset is computed.
b) Finding Nearest Neighbors: The nearest neighbors for the input pattern are determined.

The architecture of the KNN classifier consists of several components, which include storing the elements of data patterns in ROM, subtractors, square units (SQ), adders, a winner - takes - all (WTA) circuit comprising comparators and registers, and a pipelining strategy for efficient computation. The WTA circuit identifies the nearest neighbors, and this process repeats until all distances are processed. Figure 2 shows the architecture of KNN Classifier.

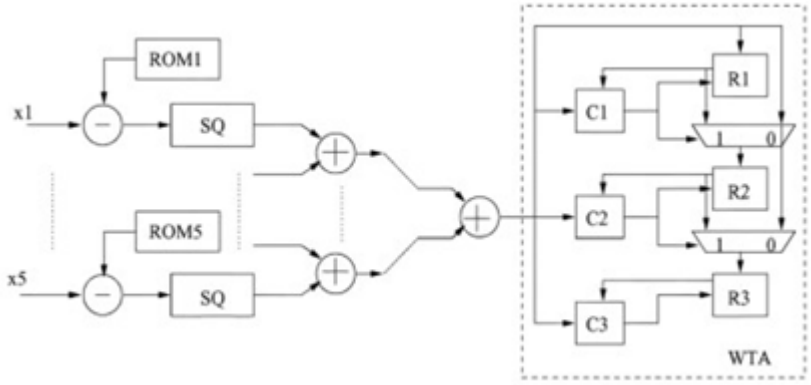

Figure 2: Architecture of a KNN Classifier

The workflow of the KNN algorithm can be summarized through the following steps:
Step 1: Determine the value of K, representing the number of neighbors to consider.
Step 2: Calculate the Euclidean distance for K neighbors.
Step 3: Select the K nearest neighbors based on the computed Euclidean distance.
Step 4: Among these K neighbors, tally the number of data points within each category.
Step 5: Assign the new data point to the category with the highest neighbor count.
Step 6: The KNN model is now prepared for classification.

4.2 Random Forest (RF)

The Random Forest algorithm, a widely acclaimed machine learning technique, is an integral part of supervised learning methodology. It is applicable to addressing machine learning challenges spanning both classification and regression tasks. Grounded in the principle of ensemble learning, Random Forest is crafted to amalgamate multiple classifiers for tackling intricate problems and augmenting model performance.

The Random Forest classifier, as its name suggests, operates by constructing numerous decision trees on various subsets of the provided dataset and subsequently averaging their predictions to improve the accuracy of predictions. In contrast to relying on a single decision tree, Random Forest aggregates forecasts from each tree and makes predictions based on the majority of these predictions. This approach, often referred to as ensemble learning, facilitates improved accuracy and mitigates the overfitting issue. Random Forest, finds widespread application across various industries, including banking and e - commerce. A Random forest classifier is shown in figure 3.

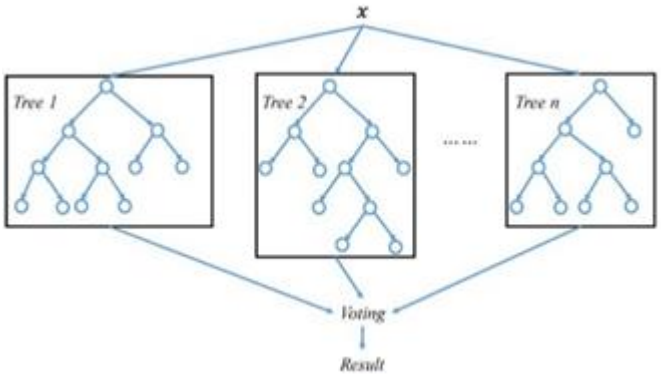

Figure 3: Random forest Classifier

The Random Forest algorithm harnesses the power of multiple decision trees. This algorithm constructs a forest by employing bagging, also known as bootstrap aggregation. Bagging is an ensemble meta - algorithm crafted to enhance

the predictive accuracy of machine learning algorithms. The algorithm subsequently derives the final result by considering the predictions made by these decision trees, typically through methods like averaging or voting on the outcomes. As the number of trees increases, the accuracy of the result proportionally improves. To ensure the efficacy of a Random Forest classifier, two key assumptions must be met:

- The feature variable of the dataset should contain actual values to enable the classifier to make accurate predictions instead of conjectured results.
- The predictions from each tree should exhibit minimal correlation with one another.

4.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is among the most widely adopted supervised learning algorithms, serving dual purposes for both Classification and Regression tasks. The core objective of the SVM algorithm is to establish an optimal line or decision boundary capable of segregating an n - dimensional space into distinct classes. This decision boundary is referred to as a hyperplane. SVM accomplishes this by selecting extreme points or vectors pivotal in the creation of the hyperplane. These critical instances are referred to as support vectors.

The following are some of the features of SVM algorithm:

- Computationally Intensive: SVMs can become computationally intensive, particularly when dealing with large datasets or complex kernel functions. The training time can be protracted, rendering them less suitable for real - time or interactive applications.
- Memory Consumption: SVMs necessitate the storage of support vectors and their associated information in memory during both training and prediction phases. This can lead to substantial memory consumption, especially when handling voluminous datasets or high - dimensional feature spaces.
- Sensitivity to Noise: SVMs exhibit sensitivity to noisy data and outliers present in the training set. Outliers can exert a significant influence on the positioning of the decision boundary and overall model performance.
- Challenging to Tune: Despite having fewer hyperparameters compared to some other algorithms, SVMs still entail parameter tuning, including the choice of kernel function, kernel hyperparameters, and the regularization parameter (C). Determining the optimal combination of these parameters can be a demanding and time - consuming task.
- Limited Scalability: SVMs might not scale well to exceptionally large datasets due to their computational complexity. Training on millions of samples can be impractical in terms of both time and memory.
- Imbalanced Data: SVMs may encounter difficulties when dealing with imbalanced datasets where one class significantly outnumbers the other. This imbalance has the potential to result in the majority class dominating the decision boundary, which can lead to suboptimal performance for the minority class.

4.4 Decision Tree (DT)

Decision Trees (DT) are a classification method that excels in distinguishing records with numerous features by scrutinizing specific properties from the root to the terminal nodes within a tree - like structure. Each terminal node, also known as a leaf, is associated with a class label. These properties are often evaluated based on the presence or absence of specific attributes or conditions, such as the occurrence of particular words or features.

The tree undergoes successive partitions until a minimal number of records remains in each leaf. Decision Trees are a powerful tool in machine learning and data analysis, offering a clear and interpretable way to make decisions based on data. The key strengths of Decision Trees include their ability to handle both numerical and categorical data, their interpretability, and their effectiveness in capturing complex decision boundaries. However, Decision Trees are prone to overfitting, especially when they are deep and overly complex. To mitigate these issues, various techniques such as pruning and setting minimum sample size criteria for splitting nodes can be applied. Figure 4 depicts a decision tree classifier.

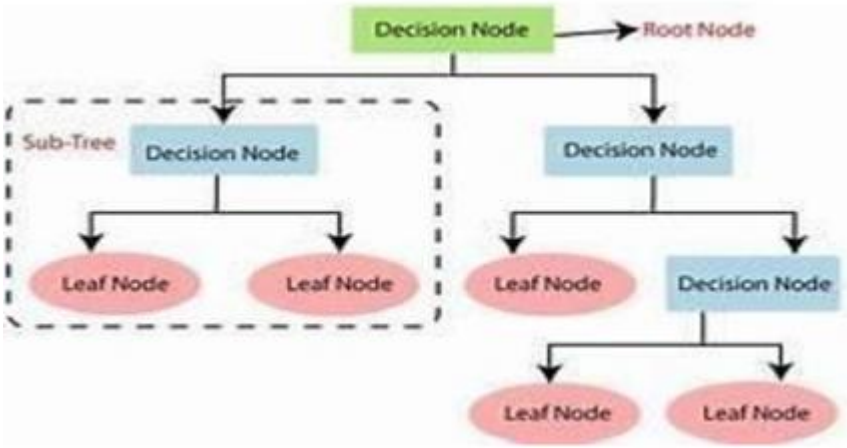

Figure 4: Decision tree classifier

Decision Trees offer a transparent and intuitive approach to decision - making in classification and regression tasks. However, careful consideration of tree depth and complexity is essential to prevent overfitting and ensure optimal model performance.

4.5 Artificial Neural Networks (ANNs)

Artificial Neural Networks represent a pivotal paradigm in the realm of machine learning, profoundly inspired by the intricate structure and functionality of the human brain's neural networks. The fundamental structure of an Artificial Neural Network (ANN) consists of interconnected nodes, commonly known as neurons, arranged into specific layers: an input layer, one or more hidden layers, and an output layer.

The essence of ANNs resides in their capability to process and analyse data as it traverses through these layers, with every connection representing a weight that modulates the information flow. The crux of ANN's learning ability lies in the dynamic adjustment of these weights, a process facilitated by exposure to input data and the corresponding desired output. This iterative weight adaptation empowers ANNs to discern complicated patterns, make predictive inferences, and accomplish diverse tasks, including image recognition, natural language processing, and complex decision - making. An architecture of a three layer ANN is shown in figure 5.

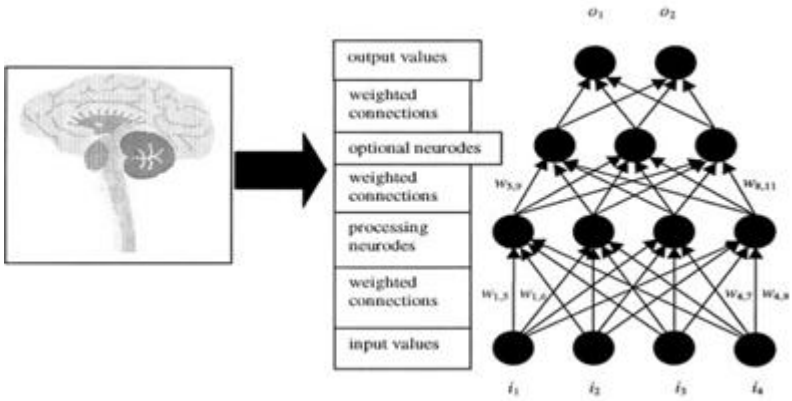

Figure 5: Artificial Neural Networks

ANNs have fostered the emergence of deep learning, an illustrious subset of machine learning, which has urged remarkable advancements across multifarious domains. By enabling computers to learn and glean profound insights from vast datasets, ANNs have catalysed transformative progress and indicated a new era of artificial intelligence and data - driven decision - making.

5. Evaluation Parameters

In the pursuit of assessing the performance of the various machine learning models for credit card fraud detection, we employ a set of crucial evaluation parameters that provide insights into their effectiveness. These parameters serve as metrics for gauging the quality of our models' predictions. The key evaluation parameters include:

Accuracy: Accuracy quantifies the proportion of correctly predicted instances compared to the total number of predictions.

Precision: Precision is the ratio of true positive predictions to the total number of positive predictions made by our models and is defined in equation (1). A higher precision signifies a lower false positive rate, highlighting the ability to make accurate positive predictions.

$$Precision = \frac{Number\ \ of\ retrieved\ \ and\ relevant\ \ observations}{Total\ \ retrieved\ \ observations} = \frac{TP}{TP+FP} \quad (1)$$

Recall: Recall, also known as sensitivity, calculates the proportion of actual positive instances correctly identified by our models and is defined in equation (2). It provides an indication of our models' capacity to capture genuine positive cases.

$$Recall = \frac{Number\ \ of\ retrieved\ \ and\ relevant\ \ observations}{Total\ \ relevant\ \ observations} = \frac{TP}{TP+FN} \quad (2)$$

F1 - Score: The F1 - score is a harmonious blend of recall and precision, offering a balanced assessment of our models' performance and is defined using equation (3). It provides a weighted average that considers both false positives and false negatives, thus encompassing the trade - off between precision and recall.

$$F1\ Score = 2 * \frac{(recall\ * precision\ )}{(recall + precision\ )} \quad (3)$$

These evaluation parameters play a pivotal role in quantifying the effectiveness of the machine learning techniques in credit card fraud detection, aiding in the selection of the most suitable models.

6. Proposed Framework

In this section, we present a robust framework for credit card fraud detection, leveraging the available dataset and harnessing the advancements in technology. The proposed framework encompasses a series of well - defined steps to ensure the development of an effective fraud detection model. The architectural overview of the proposed system is illustrated in Figure 6. The key steps involved in the proposed framework are as follows:

1) Collecting Data:
It is imperative to ensure that the dataset is sourced from a reputable and trustworthy origin to maintain accuracy and precision. The reliability of the dataset directly impacts the model's performance, and therefore, the procurement of high - quality data is crucial.

2) Preparing Data:
Data preparation involves several critical sub - steps:

- Aggregating and organizing collected data to eliminate biases.
- Randomizing the data to ensure even distribution, reducing the impact of data order.
- Cleaning the data, which includes removing irrelevant or redundant information, handling data types, and applying transformations like label encoding to render the dataset suitable for input to the model.
- Visualizing the data through techniques such as heatmaps and scatter plots to gain a comprehensive understanding of the dataset's characteristics.
- Splitting the data into training and testing sets, with the testing data reserved for model evaluation.

3) Choosing a Model:
Model selection is a pivotal step influenced by the nature of the reprocessed data. The choice is guided by the effectiveness of various models as revealed in the review of available models and their compatibility with the pre - processing techniques applied.

4) Training the Model:
The pre - processed training dataset is utilized to train the machine. During this phase, the machine learns from the input data, identifies patterns, and becomes capable of making predictions.

5) Evaluating the Model:
The trained model undergoes rigorous evaluation using a range of metrics suitable for the specific objectives of the proposed model. The evaluation process is performed on the training set.

6) Parameter Tuning:
Post - evaluation and parameter tuning is conducted to enhance model accuracy. Adjustments are made to various parameters, aiming to maximize accuracy levels for each parameter.

7) Making Predictions:
With the model now refined and optimized, it can be deployed for credit card fraud detection on unseen data, ensuring accurate predictions.

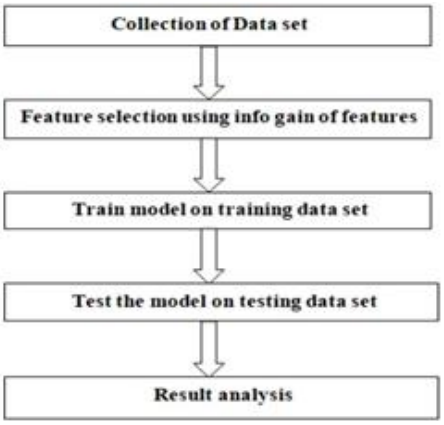

Figure 6: Frame work of proposed methodology

7. Experimental Setup

In this study, we have implemented and assessed the performance of five distinct machine learning algorithms using Python: K Nearest Neighbour, Support Vector Machine, Random Forest, Decision Tree, and Artificial Neural Network. The parameter setting of the model employed in this study are outlined in table 1.

Table 1: The parameters of the model used for comparison

Classification Model	Parameters	Number or Type
SVC	Kernel function	Radial basis function (RBF)
ANN	Number of layers	3
	Number of neurons in the hidden layer	256, 128, 64
	Activation function of the hidden layers	Relu
	Activation function of the output layer	Softmax
	Optimizer	Adam
DT	Criterion	gini
	max_depth	5
RF	n_estimators	100
	max_depth	5
	Criterion	gini
KNN	leaf_size	30
	n_neighbors	3

Table 2 summarises the performance of the various machine learning models. The experimental results reveal significant variations in the performance metrics of these classifiers. Specifically, the ANN classifier demonstrates an exceptional accuracy rate of 96.28%, indicating its robust predictive capabilities. Following closely behind is the Random Forest classifier and KNN with a noteworthy accuracy of 95%. The Support Vector Machine achieves an accuracy rate of 94%, showcasing its effectiveness in classification tasks. However, the Decision Tree classifier lags behind with an accuracy of 75.32%,

Table 2: Performance comparison of various models

Model Employed	Accuracy	Precision	Recall	F1 - score
KNN	95.2702	0.9608	0.9074	0.9333
RF	95.2703	0.9896	0.8796	0.9314
SVM	94.5946	0.9894	0.8611	0.9207
ANN	**96.2838**	**0.9896**	**0.9074**	**0.9463**
DT	92.9054	0.9307	0.8704	0.8995

To provide a comprehensive evaluation, we consider additional performance metrics, including Precision, Recall, and F1 Score. ANN excels not only in accuracy but also in Precision (0.98), Recall (0.90), and an F1 Score of 0.94. Random Forest maintains a high Precision of 0.9896, Recall of 0.8796, and an F1 Score of 0.9314. SVM demonstrates a strong Precision of 0.9894, Recall of 0.8611, and an F1 Score of 0.9207, reaffirming its classification prowess. KNN delivers an accuracy of 95.27% along with Precision (0.96), Recall (0.90), and an F1 Score of 0.93. Decision Tree, while having an accuracy of 92.9%, maintains a Precision (0.93), Recall (0.87), and an F1 Score of 0.89. Figure 7 shows performance of various models based on accuracy.

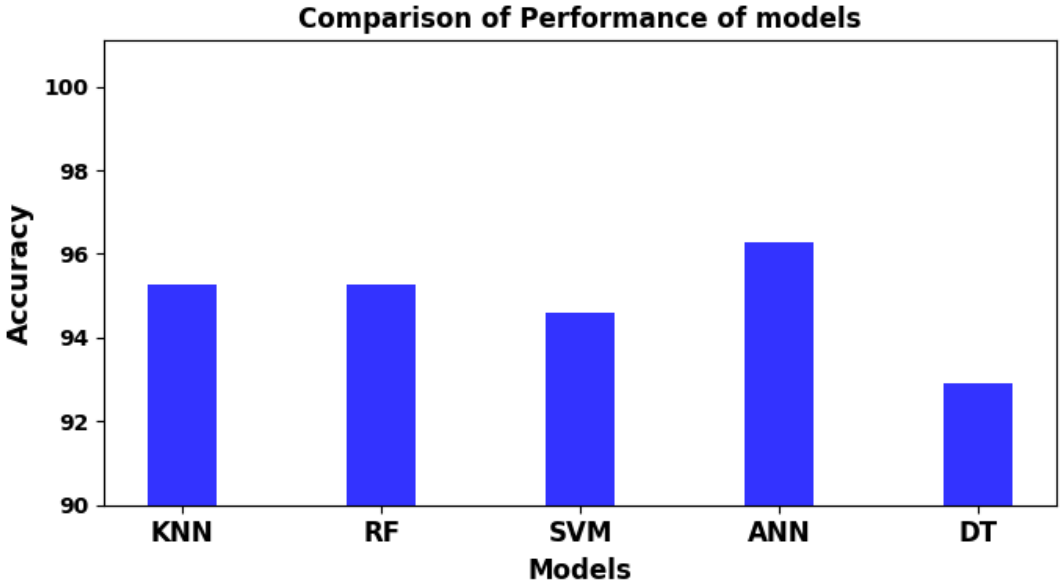

Figure 7: Plot of accuracy of models

In view of these comprehensive assessments, it is evident that all the machine learning models employed in this study exhibit satisfactory prediction accuracy. However, the ANN classifier emerges as the most efficient among the classifiers, boasting not only the highest accuracy but also Precision, Recall, and F1 Score values compared to the other classifiers under consideration.

8. Conclusion and Future Scope

The research has delved into the efficacy of five unique machine learning algorithms: K Nearest Neighbour, Support Vector Machine, Random Forest, Decision Trees, and Artificial Neural Networks. After comprehensive analysis and model evaluation, it was evident that ANN, a supervised learning algorithm, outperformed the other models with an outstanding accuracy rate of 96.28%. In comparison, RF achieved an accuracy of 95%, while SVM attained an accuracy of 94%. This highlights the superiority of the ANN classifier in efficiently detecting fraudulent transactions. The proposed model underwent rigorous testing across various transaction types, demonstrating its promising capability to successfully identify fraudulent transactions. Furthermore, a comparative analysis was conducted with existing methods, showcasing the superior performance of the proposed approach. By categorizing transactions as legal (class 0) or fraudulent (class 1), the proposed model significantly enhanced detection accuracy.

Future research in credit card fraud detection offers several promising avenues for further advancement. Advanced deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can be explored to improve accuracy and uncover complex fraud patterns. The development of explainable AI models is crucial for regulatory compliance and user trust, allowing stakeholders to understand model decisions. Real - time detection systems should be a focus, enabling the prevention of fraud as it occurs. Blockchain integration and the use of cryptocurrencies present opportunities for secure transactions.

References

[1] 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), 1 - 17.
[2] Asha, R. B., & KR, S. K. (2021). Credit card fraud detection using artificial neural network. Global Transitions Proceedings, 2 (1), 35 - 41.
[3] Osegi, E. N., & Jumbo, E. F. (2021). Comparative analysis of credit card fraud detection in Simulated Annealing trained Artificial Neural Network and Hierarchical Temporal Memory. Machine Learning with Applications, 6, 100080.
[4] Ojugo, A. A., & Nwankwo, O. (2021). Spectral - cluster solution for credit - card fraud detection using a genetic algorithm trained modular deep learning neural network. JINAV: Journal of Information and Visualization, 2 (1), 15 - 24.
[5] Itoo, F., & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. International Journal of Information Technology, 13 (4), 1503 - 1511.
[6] Olowookere, T. A., & Adewale, O. S. (2020). A framework for detecting credit card fraud with cost -

sensitive meta - learning ensemble approach. Scientific African, 8, e00464.

[7] Darwish, S. M. (2020). An intelligent credit card fraud detection approach based on semantic fusion of two classifiers. Soft Computing, 24 (2), 1243 - 1253.

[8] Dornadula, V. N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithms. Procedia computer science, 165, 631 - 641.

[9] Pumsirirat, A., & Liu, Y. (2018). Credit card fraud detection using deep learning based on autoencoder and restricted boltzmann machine. International Journal of advanced computer science and applications.

[10] Awoyemi, J. O., Adetunmbi, A. O., &Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. In 2017 international conference on computing networking and informatics (ICCNI) (pp.1 - 9). IEEE.