A FIELD PROJECT REPORT

on

Credit Card Fraud Detection Using Machine Learning Algorithms

Submitted

by

221FA04460	221FA04475
N.DIVYA	J.VASU
221FA04486	221FA04690
Sk.UMAR FAROOQ	B.SRINU BABU

Under the guidance of

Dr. S. Deva Kumar

Designation

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH Deemed to be UNIVERSITY

Vadlamudi, Guntur.

ANDHRA PRADESH, INDIA, PIN-522213.

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the Field Project entitled **Credit Card Fraud Detection Using Machine Learning Algorithms** that is being submitted by 221FA04460 (N.Divya), 221FA4475(J.Vasu), 221FA04486 (Sk.Umar Farooqr)**,** 221FA04690 (B.Srinu) for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Ms. G.NAVYA, M.Tech., Assistant Professor, Department of CSE.

Guide name& Signature	Dr. S.V. Phani Kumar	Dr.K.V. Krishna Kishore
Assistant/Associate/Professor, CSE	HOD,CSE	Dean, SoCI

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

We hereby declare that the Field Project entitled **"extracting text from imag Credit Card Fraud Detection Using Machine Learning Algorithms "** that is being submitted by 221FA04460 (N.Divya), 221FA4475(J.Vasu), 221FA04486 (Sk.Umar Farooqr), 221FA04690 (B.Srinu) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Ms. Dr. N. Sameera., Assistant Professor, Department of CSE.

By
221FA04460(N. Divya)
221FA04475(J. Vasu)
221FA04486(Sk. Umar Farooqr)
221FA04690(B. Srinu)

Date:

ABSTRACT

Abstract—Credit card fraud is a critical issue that leads to significant financial losses for both individuals and financial institutions. This study focuses on detecting credit card fraud by leveraging various machine learning algorithms, including Logistic Regression, Decision Trees, and Random Forest. By evaluating the performance of these algorithms on a dataset containing transaction details, we aim to identify the most effective model for accurately detecting fraudulent transactions. This approach provides a robust and reliable method for fraud detection, potentially aiding financial institutions in reducing losses due to fraudulent activities. Our results highlight the potential of machine learning in enhancing the accuracy and efficiency of fraud detection systems. The model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score to ensure its effectiveness. Results demonstrate that the proposed system significantly reduces false positives while maintaining high detection accuracy. This research aims to contribute to the development of more reliable and scalable fraud detection mechanisms for financial institutions.

***Index Terms*—Credit Card Fraud Detection, Machine Learning, Logistic Regression, Decision Trees, Random Forest, Fraudulent Transactions, Model Evaluation, Support Vector Machine (SVM), Neural Networks (NN), Naive Bayes, K- Nearest Neighbour (KNN), XGBoost**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1 Motivation

The increasing prevalence of credit card fraud poses a significant threat to both consumers and financial institutions. With the rise of online and contactless payments, fraudulent activities have become more sophisticated and difficult to detect. Traditional security measures often fall short, especially in scenarios where attackers gain access to sensitive card information without physical theft. The need for an advanced and reliable fraud detection system, capable of identifying unusual spending patterns in real-time, has never been more crucial. By leveraging machine learning and behavioral analysis, this research aims to enhance fraud detection methods, reducing financial losses and improving the security of credit card transactions.[1]

The detection of credit card fraud has gained significant attention recently, driven by the rapid expansion of big data and AI technologies. This area of research is crucial for financial institutions, as it helps minimize losses resulting from fraudulent transactions. While several methods have shown encouraging outcomes, detecting fraudulent transactions accurately and quickly remains a challenge. This is primarily due to the highly imbalanced nature of the data and the wide variability in fraud patterns.[2]

1.2 Problem Definition

Credit card fraud has become an escalating issue in the financial sector, particularly with the increasing volume of digital transactions. In recent years, fraudulent activities have outpaced the growth of the credit card industry itself, resulting in substantial financial losses for both consumers and financial institutions. Despite advancements in security technologies, fraud schemes such as counterfeit cards, stolen cards, non-receipt of issued cards, and fraudulent applications remain prevalent. Additionally, the rise of mail-order and telephone-order fraud poses significant challenges due to the absence of physical card verification. The complexity and adaptability of these fraud methods highlight the urgent need for advanced detection systems that can accurately and swiftly identify suspicious transactions to mitigate financial losses. This research aims to address these challenges by leveraging machine learning and data analysis techniques to enhance fraud detection mechanisms.[3]

1.3 Constraints

This research will address the following constraints:

Accessibility: Ensuring that the proposed system is available across various platforms and interfaces.

Code: Efficient, scalable code that can handle real-time transactions and adapt to evolving fraud patterns.

Constructability: The system must be easy to deploy in existing financial institutions' infrastructure.

Cost: Minimizing operational and maintenance costs while maximizing detection efficiency.

Extensibility: Allowing for future enhancements and integration with other detection tools.

Functionality: Providing robust and accurate detection mechanisms without affecting user experience.

Interoperability: Ensuring compatibility with different banking systems and transaction processing environments.

Legal Considerations: Compliance with financial regulations, privacy laws, and ethical considerations.

Maintainability: Easy updates and modifications to stay ahead of fraud techniques.

Marketability: The solution should be appealing to financial institutions as a commercial product.

Schedule: Developing and testing the solution within a specified time frame.

Standards: Adhering to industry standards and best practices in fraud detection.

Sustainability: The system must remain effective over time, considering technological advancements and evolving fraud patterns.

Usability: The system should be user-friendly for both analysts and end-users.

Security: Ensuring that the system itself is secure and not vulnerable to exploitation.

Privacy and Ethical Considerations: Protecting customer data and ensuring ethical use of AI in fraud detection.

1.4 Design Standards

The research will follow industry-standard design methodologies, ensuring modularity, scalability, and performance. Standards related to financial data security, will be adhered to, along with ensuring the system meets the criteria for real-time performance and accuracy.

1.5 Major Contributions/Objectives

i. Propose a machine learning-based model for detecting credit card fraud with enhanced accuracy.

ii. Address the challenge of imbalanced data using advanced techniques such as SMOTE (Synthetic Minority Over-sampling Technique).

iii. Implement and evaluate real-time fraud detection to minimize the risk of fraudulent transactions.

iv. Investigate privacy-preserving methods to ensure customer data protection.

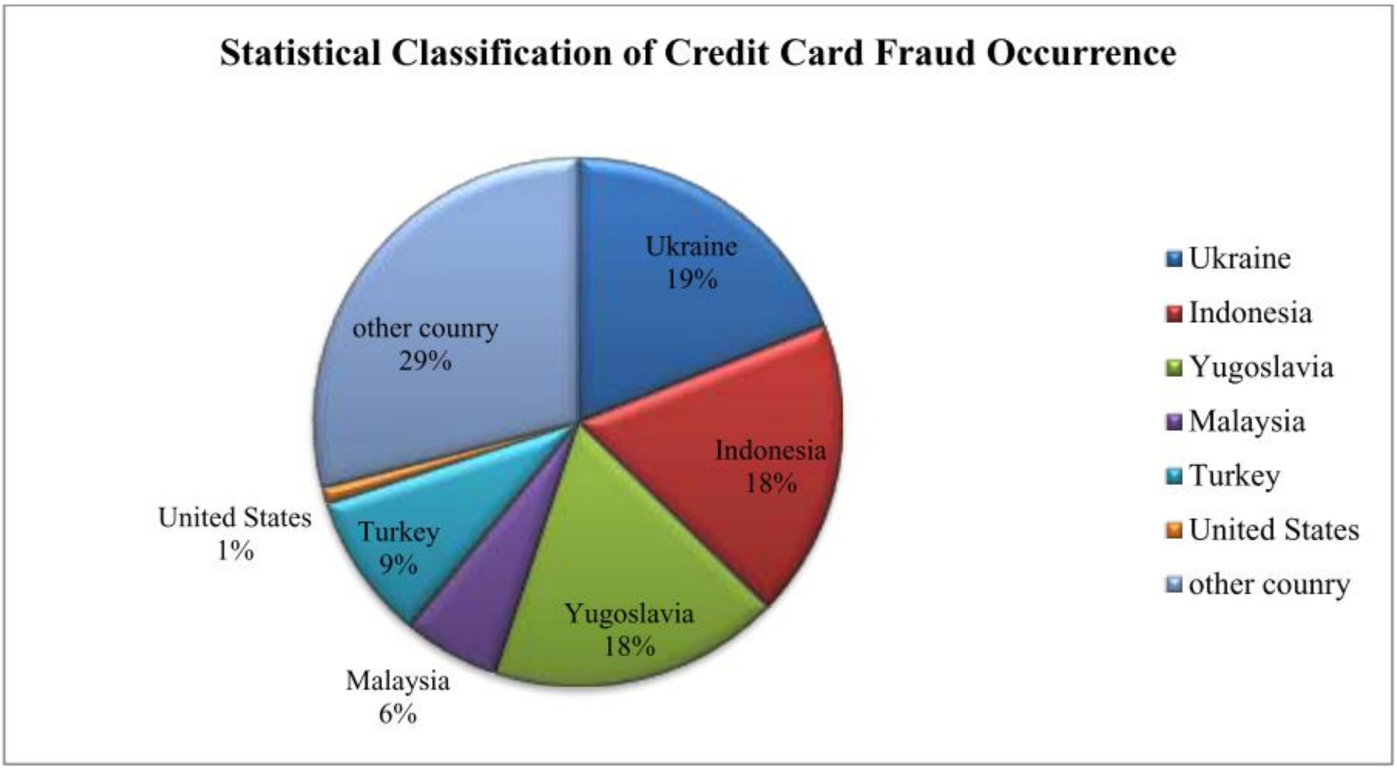


Figure 1. High risk countries facing credit card fraud threat

1.6 Difficulties of Credit Card Fraud Detection

- **Imbalanced Data:** The credit card fraud detection landscape is fraught with challenges primarily due to the imbalanced nature of transaction data, where fraudulent activities represent a minuscule fraction of total transactions, complicating accurate detection.
- **Different Misclassification Importance:** Furthermore, misclassification carries differing implications; erroneously labelling a legitimate transaction as fraudulent can lead to unnecessary investigations, while failing to identify a fraudulent transaction can have severe repercussions.
- **Overlapping Data:** This overlap between legitimate and fraudulent transactions exacerbates the difficulty in maintaining a low false positive and false negative rate.
- **Lack of Adaptability:** Additionally, many classification algorithms struggle to adapt to emerging patterns of fraud and normal behaviour, rendering traditional supervised and unsupervised systems less effective.
- **Fraud Detection Cost:** Finally, the cost of fraud detection must balance the expenses incurred from false positives against the potential losses from undetected fraudulent transactions, making it critical for systems to optimize both detection efficiency and cost-effectiveness.

CHAPTER-2 LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Literature review

Multiple Supervised machine learning techniques are used for fraud detection[4], Various approaches have been explored in the realm of credit card fraud detection, encompassing both supervised. Supervised techniques typically rely on labelled datasets to train models, enabling them to distinguish between legitimate and fraudulent transactions. This approach is particularly useful in scenarios where labelled data is scarce, allowing for the detection of novel fraud schemes that may not be represented in the training set.[5], [6].

In the context of credit card fraud detection, many machine learning techniques approach the problem as a supervised classification task. This method involves utilizing a labelled dataset to train a classifier, allowing it to learn the characteristics of both legitimate and fraudulent transactions. Once trained, the model can then be employed to classify new, unlabelled transaction data as either normal or abnormal. In this section, we will explore six commonly used supervised machine learning algorithms that have proven effective in detecting fraudulent activities in credit card transactions. By understanding the strengths and weaknesses of each approach, we can better assess their suitability for real-world applications in fraud detection.

We know that all fraudulent transactions follow a similar pattern, and by using any pattern recognition system such as Support Vector Machine (SVM), Neural Networks (NN), Naive Bayes, K-Nearest Neighbour (KNN), Decision Trees, we can classify transactions as fraudulent. Additionally, methods like Logistic Regression, Random Forest, Classification and Regression Trees (CART), Synthetic Minority Over-sampling Technique (SMOTE) can also be employed for effective classification. The working of these methods is explained below.[7]

Credit card fraud has emerged as a critical issue in modern financial systems, driven by the growing reliance on online transactions and the increasing sophistication of fraudulent activities. To combat this, numerous fraud detection techniques, primarily leveraging data mining and machine learning, have been proposed.

1. Fraud Detection Techniques

An early review of credit card fraud detection techniques. They categorize fraud into various types, such as counterfeit, lost or stolen cards, and application fraud, and emphasize the use of data mining techniques to combat fraud. According to the authors, the implementation of these methods in European markets has been particularly effective in reducing fraud rates. However, the issue of false positives, where legitimate customers are flagged as fraudulent, remains an ethical and operational challenge. The paper advocates for a combination of detection techniques to optimize both the accuracy of detection and the customer experience. [8]

2. Machine Learning Approaches

In recent years, the application of machine learning (ML) to fraud detection has gained considerable attention. Describe a machine learning-based approach for fraud detection using real-time streaming data. Their study focuses on analyzing historical transaction data and extracting behavioral patterns. They propose clustering cardholders based on transaction amounts, followed by training classifiers such as decision trees, logistic regression, and random forests. The authors highlight the challenge of handling imbalanced datasets, where fraudulent transactions are significantly fewer than legitimate ones. To address this, they employ techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and discuss the problem of concept drift, where the characteristics of fraud evolve over time.[9]

2.2 Credit Card Fraud Detection Machine Learning Techniques

1. Support Vector Machines (SVM)

Support Vector Machines (SVM) is a supervised learning algorithm that classifies data points by determining the best separating hyperplane between different classes. The primary objective of SVM is to identify this optimal hyperplane, which maximizes the margin between classes. While multiple hyperplanes may exist, SVM focuses on finding the one that best differentiates the classes. The data points that lie closest to the hyperplane are referred to as support vectors, and these points are crucial for predicting the classifications of new data. When a new data point arrives, it is evaluated using the hyperplane equation to ascertain its class based on which side of the hyperplane it falls on within the vector space. The SVM model is trained using labelled data, where the outcomes are already known, allowing it to learn the patterns associated with fraudulent and legitimate transactions. Consequently, it can effectively classify new transactions into the appropriate categories.[10], [11].

2. Neural Networks (NN)

Neural networks (NN) are advanced computational models inspired by the human brain, designed to recognize patterns and make predictions based on data. In the context of credit card fraud detection, neural networks are particularly effective due to their ability to process large volumes of transactional data and identify complex relationships among various features, such as transaction amounts, user behaviour, and merchant details. By employing multiple layers of interconnected nodes, these networks can learn from historical data to differentiate between legitimate and fraudulent transactions. Training a neural network involves feeding it vast amounts of labelled data, allowing it to adjust its internal parameters and improve its accuracy over time. This adaptability makes neural networks suitable for dynamic environments where fraud tactics constantly evolve. However, ethical considerations arise, as misclassifying legitimate customers as fraudulent can harm business relationships and lead to lost revenue. Thus, while neural networks enhance fraud detection capabilities, they must be implemented with careful attention to minimizing errors and ensuring fair treatment of all customers.[12]

3. Naive Bayes

Naive Bayes play a significant role in credit card fraud detection by modelling the dependencies between different variables associated with fraudulent and legitimate transactions. These networks help in predicting the probability of fraud even when data is uncertain or incomplete. For instance, fraudulent user behaviour can be modelled using expert knowledge, while legitimate user behaviour is based on historical data from non-fraudulent transactions. As new transaction data emerges, the legitimate behaviour model adapts to the specific user, improving its accuracy. To detect fraud, each new transaction is evaluated against both models, and the Bayesian network helps classify whether the transaction is likely fraudulent or legitimate. This probabilistic approach enhances detection capabilities in the presence of uncertainty and enables efficient identification of fraud patterns in real-time transactions.[13]

4. K-Nearest Neighbour (KNN)

The K-Nearest Neighbour (KNN) algorithm is a powerful tool in the realm of credit card fraud detection, functioning as a supervised learning technique that classifies new transactions based on the categories of their nearest neighbours. When a new transaction occurs, the algorithm assesses its similarity to previously recorded transactions using a defined distance metric, such as Euclidean distance. By evaluating the nearest points in the dataset, KNN can effectively identify whether a transaction is likely to be fraudulent or legitimate. However, the effectiveness of the KNN algorithm is influenced by several factors, including the choice of distance metric, the classification rule applied to determine the category, and the number of neighbours considered. A smaller value of K, typically 1, 3, or 5, can make the model sensitive to noise in the data, while larger values can enhance stability and reduce false positives. By optimizing these parameters, KNN can improve its performance in distinguishing between genuine transactions and potential fraud. when combined with genetic algorithms for distance metric optimization, KNN remains a fast and effective method for enhancing credit card fraud detection efforts. [7], [14].

5. Decision Trees

Decision trees have emerged as a valuable tool in credit card fraud detection due to their intuitive structure and ease of interpretation. They operate by segmenting data into branches based on specific attribute values, effectively simplifying complex decision-making processes. Each branch represents a decision point, leading to further splits or ultimately reaching a leaf node that indicates the classification outcome—whether a transaction is legitimate or fraudulent. This method allows for a clear visual representation of the decision-making process, making it accessible for stakeholders to understand how fraud detection is carried out. Moreover, decision trees do not require assumptions about data distribution, enhancing their flexibility and applicability to diverse datasets.

While decision trees offer significant advantages in fraud detection, they also have certain limitations. One key challenge is the need to evaluate each transaction individually, which can be time-consuming, particularly when dealing with high transaction volumes. Additionally, decision trees can be prone to overfitting, where the model becomes overly complex and tailored to training data, potentially reducing its effectiveness on unseen data. Despite these drawbacks, the adaptability and transparency of decision trees make them a popular choice for identifying fraudulent transactions in the credit card industry, allowing institutions to enhance their fraud prevention strategies.[15]

6. Logistic Regression

Logistic regression is a powerful statistical tool used in credit card fraud detection due to its ability to model binary outcomes, such as whether a transaction is legitimate or fraudulent. By utilizing various predictor variables, such as transaction amount, merchant type, and customer behaviour, logistic regression estimates the probability of fraud occurring in a transaction. This approach allows financial institutions to identify high-risk transactions by quantifying how each predictor influences the likelihood of fraud, making it easier to develop targeted strategies for mitigating risks.

In practice, logistic regression provides a straightforward and interpretable framework for analysing patterns in transactional data. As the model outputs probabilities, it enables banks to establish a threshold for flagging suspicious transactions, balancing the trade-off between false positives and negatives. This is particularly important in fraud detection, where misclassifying a legitimate transaction as fraudulent can lead to customer dissatisfaction, while failing to catch an actual fraudulent transaction can result in significant financial losses. Overall, logistic regression serves as an essential tool for enhancing the effectiveness and efficiency of fraud detection systems in the banking sector.[16]

7. Random Forest

Random Forest (RF) techniques have emerged as a powerful solution for credit card fraud detection, leveraging their bagging approach to enhance the robustness and accuracy of predictions. By combining multiple decision trees, RF effectively reduces overfitting and improves generalization, making it particularly suited for identifying fraudulent transactions. For instance, integrating RF with various sampling methods can further optimize performance, allowing for a more targeted analysis of transaction data.

In addition to traditional RF applications, researchers have explored hybrid models that incorporate other machine learning algorithms to bolster fraud detection capabilities. For example, combining RF with J48 classifiers has shown promising results, outperforming other techniques such as Support Vector Machines and K-Nearest Neighbour’s in certain scenarios. Moreover, the preprocessing of data through feature extraction and cleaning is crucial for enhancing the accuracy of RF models. Variations of the RF classifier, including those focused on record distances and Gini-criterion calculations, also demonstrate the flexibility and adaptability of RF methods in tackling the ever-evolving challenges of credit card fraud detection.[17]

8. Classification and Regression Trees (CART)

Classification and Regression Trees (CART) are powerful tools utilized in credit card fraud detection to analyze transaction data and classify it as either fraudulent or legitimate. Developed by Breiman in 1984, CART builds decision trees based on the Gini index, which measures the impurity of a dataset at each node. This approach allows the model to effectively handle both numeric and categorical attributes, making it versatile for various types of transaction data. Additionally, CART can manage missing values, ensuring that the analysis remains robust even when certain information is unavailable.

In the context of fraud detection, CART provides a clear visual representation of decision paths, helping stakeholders understand the reasoning behind classifications. By constructing new features automatically within each node, CART enhances the model's predictive power. This capability is particularly valuable in dynamic environments, where patterns of fraudulent behaviour can shift over time. By leveraging CART, financial institutions can better identify suspicious transactions, thereby minimizing losses and improving their overall risk management strategies.[18]

9. Synthetic Minority Over-sampling Technique (SMOTE)

In credit card fraud detection, the issue of classification imbalance presents a significant challenge for machine learning models. Fraudulent transactions comprise only a small fraction of the total transactions, leading to a skewed dataset that can result in misclassification when training algorithms. To address this imbalance, researchers often utilize data preprocessing techniques, such as oversampling the minority class of fraudulent transactions while under sampling the majority class of legitimate transactions. This approach helps to create a more balanced dataset, allowing machine learning algorithms to better distinguish between genuine and fraudulent activities.

One effective method for handling imbalanced data is the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic examples of the minority class. Although SMOTE can enhance the performance of fraud detection models, it also introduces challenges, such as the potential for noise and overlapping data points that may lead to overfitting. Alternatives like Adaptive Synthetic (ADASYN) and Ranked Minority Over-sampling in Boosting (RAMO) have been developed, but they can complicate the classification process due to increased iterations. As a result, researchers recommend using ensemble classifiers, which combine multiple models, as they tend to outperform single classifiers when dealing with imbalanced datasets.[19]

Advantages and Disadvantages of Machine Learning Algorithms

Technique	Advantages	Disadvantages
Support Vector Machine (SVM)	-Effective in high-dimensional spaces - Works well with clear margin of separation	- Memory intensive - Poor performance on large datasets
Neural Networks (NN)	- Capable of modeling complex patterns - Can learn from unstructured data	- Requires a lot of data - Difficult to interpret results
Naive Bayes	-Simple and fast -Works well with small datasets - Effective for text classification	- Assumes feature independence - Performs poorly on complex relationships
K-Nearest Neighbour (KNN)	- Simple to understand - No training phase - Can adapt easily to changing data	- Computationally expensive during testing - Sensitive to irrelevant features
Decision Trees	- Easy to interpret - Handles both numerical and categorical data - Can capture non-linear relationships	- Prone to overfitting - Can be unstable due to small data changes

Logistic Regression	- Simple and efficient for binary classification - Outputs probabilities	- Assumes linear relationship - Cannot model complex relationships
Random Forest	- Reduces overfitting - Handles large datasets well - Robust to noise	- Less interpretable than decision trees - Slower to predict due to many trees
Classification and Regression Trees (CART)	- Generates interpretable models - Can be used for both classification and regression	- Prone to overfitting - Can become complex if not pruned
Synthetic Minority Over-sampling Technique (SMOTE)	- Balances class distribution - Helps improve classifier performance on minority classes	- Risk of overfitting - Generated samples may not always be realistic

Figure 2 Advantages and Disadvantages of Machine Learning Algorithms

CHAPTER-3 PROPOSED METHODOLOGY

3. PROPOSED METHODOLOGY

3.1 Proposed Work Flow

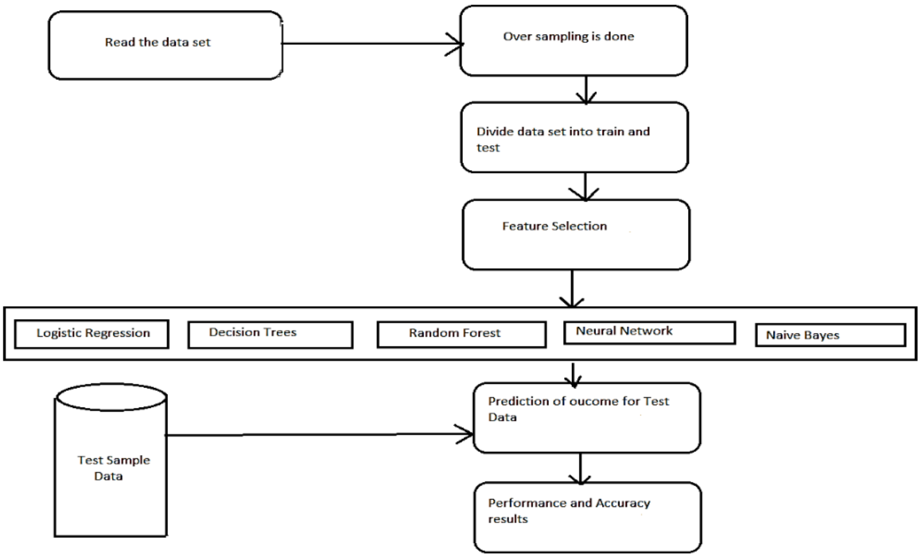


Figure *2* Proposed Work Flow

3.2 Step-wise discussion of each point mentioned in Proposed Flow Diagram

1. **Read the data set:** This involves loading the credit card transaction data into the system. The data likely contains features such as transaction amount, time, location, and other relevant information.
2. **Over sampling is done:** This step addresses the class imbalance problem common in fraud detection. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be used to create synthetic examples of the minority class (fraudulent transactions).[20]
3. **Divide data set into train and test:** The data is split into training and testing sets. A common split is 80% for training and 20% for testing, but this can vary based on the size of the dataset.
4. **Feature Selection:** Important features are selected to improve model performance and reduce dimensionality. Techniques like correlation analysis, mutual information, or feature importance from tree-based models can be used.
5. **Application of multiple machine learning models:**
   - Logistic Regression
   - Naive Bayes
   - SVM
   - Decision Trees

- XGBoost
- Neural Network

6. **Prediction of outcome for Test Data:** Each model makes predictions on the test data set.

7. **ROC curve for each model**

8. **Performance and Accuracy results:** The models' performances are evaluated using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

3.3 Major Design/ Architecture / Algorithm of the Proposed Models

Our methodology involves the implementation and evaluation of three machine learning algorithms: Logistic Regression, Decision Trees, and Random Forest, on a credit card fraud detection dataset. The process is outlined as follows

A. Data Collection and Preprocessing

We utilized a publicly available credit card transaction dataset containing features such as transaction amount, time, and anonymized features V1 to V28. The dataset was pre-processed to handle missing values, outliers, and imbalanced class distribution. Feature selection was performed based on their relevance to fraud detection, and Principal Component Analysis (PCA) was applied to reduce dimensionality and noise.

B. Algorithm Implementation

1. **Logistic Regression:** A binary classification algorithm that estimates the probability of an instance belonging to a particular class. Mathematical model: $P(Y=1|X) = 1 / (1 + e^{\wedge}(-z))$, where $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$..[21]
2. **Naive Bayes:** Based on Bayes' theorem with an assumption of independence between features. P(Class|Features) = P(Features|Class) * P(Class) / P(Features).
3. **A Support Vector Machine (SVM):** algorithm used for classification and regression tasks. It aims to find the optimal hyperplane that best separates different classes in the feature space. In cases where data is not linearly separable, SVM uses a kernel trick to map data to a higher-dimensional space where a linear separation is possible.[22]

$$w \cdot x + b = 0$$

4. **Decision Trees:** A tree-like model of decisions. Each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. Key concept: Information Gain = Entropy(parent) - Weighted Sum of Entropy(children).

5. **XGBoost (Extreme Gradient Boosting)l :** ibrary for gradient boosting decision trees. It improves the performance and speed of traditional gradient boosting algorithms, making it popular for both regression and classification tasks. XGBoost builds models in a sequential way by combining the predictions of many weak learners (usually decision trees) to create a strong predictive model.

6. **Neural Network:** A series of interconnected layers of neurons that can learn complex patterns. For each neuron: output = activation_function(Σ(weight_i * input_i) + bias).

C. Model Training:

- Train a feed-forward Neural Network on the entire Training data
- Train another feed-forward Neural Network on under sampled training data – 60% fraudulent transactions and the same number of uniformly sampled normal transactions;
- Train another feed-forward Neural Network on under sampled training data – 60% fraudulent transactions and half the number of uniformly sampled normal transactions
- Train a Random Forest on entire Training data having 400 decision trees

D. Model Testing:

For model testing use testing data and output the result which is outputted by the majority of classifiers.

3.4 Model discussion along with Mathematical Modelling:

1. **Logistic Regression:**
   - Suitable for linearly separable classes.
   - Can provide feature importance
   - Optimization typically uses gradient descent to minimize log loss:

$$P(Y=1|X)=\frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$$

where P(Y=1|X) is the probability of fraud (1) versus non-fraud (0), X represents the feature vector, and β\betaβ are the model parameters. The model's output is a probability score that can be thresholded to make final class decisions.

2. **Decision Trees:**
   - Can capture non-linear relationships
   - Prone to overfitting if not pruned

- Splitting criteria often uses Gini impurity or entropy:

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} (p_k)^2$$

where pk is the proportion of class K in dataset D. Trees are intuitive and provide a clear decision path, but they can easily overfit the data.

3. **Neural Network:**
    - Can learn complex non-linear relationships
    - Requires careful tuning of hyperparameters

$$Y = f(X; W) = \sigma(W^{(L)}\sigma(W^{(L-1)}...\sigma(W^{(1)}X + b^{(1)}) + b^{(L-1)}) + b^{(L)})$$

where W represents the weights, b the biases, and σ the activation function (e.g., ReLU, sigmoid). NNs can learn complex patterns and are particularly effective with large datasets.

4. **Random Forest:**
    - Reduces overfitting compared to single decision trees
    - Can handle high-dimensional data well

$$\hat{Y} = \frac{1}{M} \sum_{m=1}^{M} f_m(X)$$

where M is the number of trees, and fm(X) is the prediction of the mth tree. Random forests enhance stability and accuracy over individual trees.

5. **Support Vector Machine (SVM)**
    - SVM aims to find a hyperplane that best separates the classes in the feature space. The decision function is expressed as:

$$f(X) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i K(X_i, X) + b)$$

where K(Xi,X) is a kernel function, αi are the Lagrange multipliers, and b is the bias term. SVMs are effective in high-dimensional spaces and can model complex relationships using non-linear kernels.

6. **Synthetic Minority Over-sampling Technique (SMOTE)**
    - SMOTE is a data preprocessing technique used to balance imbalanced datasets by creating synthetic samples of the minority class. Given a minority instance xix_ixi and its nearest neighbours, new samples are generated using:

$$x_{new} = x_i + \lambda(x_{nearest} - x_i)$$

where λ is a random number between 0 and 1, and x_nearest is a randomly chosen instance from the k-nearest neighbour's. This helps improve classifier performance on imbalanced datasets.

3.5 DESIGN SPECIFICATION

The design specification process begins with gathering data from the designated source. Upon collecting the data, we proceed to preprocess and conduct exploratory data analysis (EDA). This stage involves several crucial steps:

1. Data Cleaning: The process starts with removing duplicate entries and addressing any null values to maintain the integrity and quality of the dataset, which is essential for accurate analysis.

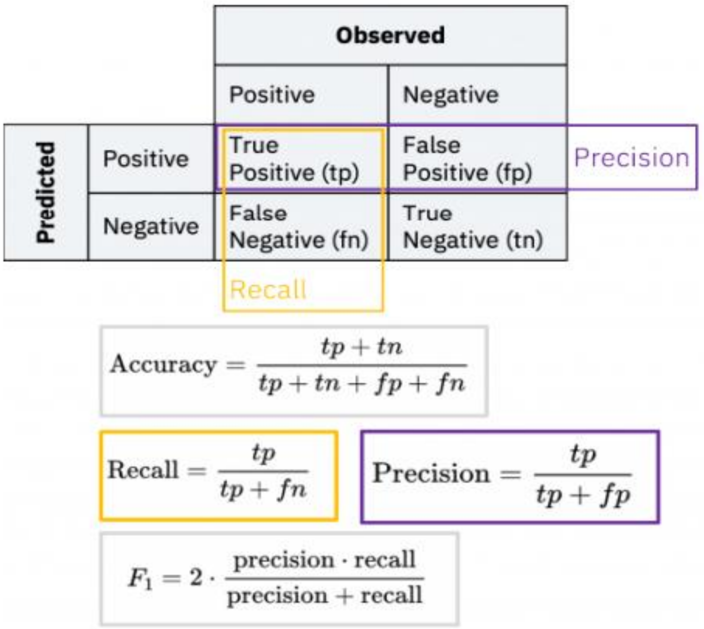
2. Exploratory Data Analysis (EDA): During this phase, we delve into the dataset to uncover hidden patterns and relationships. We utilize various statistical techniques and visualizations to gain insights into the data's distribution and characteristics.

3. Feature Selection: Following EDA, we filter the features to retain only those that are significant for our analysis. This step is vital for reducing dimensionality and enhancing model performance. However, for comparative analysis, we also run the models using the full set of features, including those initially filtered out.

4. Data Splitting: Once the data is preprocessed, we partition it into training and testing datasets. This division is crucial for evaluating the model's performance without bias.

5. Model Training: Finally, we proceed to train our selected models on the training dataset. This step involves applying various algorithms to identify patterns and relationships that can later be used for predictions.

By following this structured approach, we aim to ensure a comprehensive analysis and a robust model development process, ultimately leading to effective predictions in our application.



		Observed	
		Positive	Negative
Predicted	Positive	True Positive (tp)	False Positive (fp)
	Negative	False Negative (fn)	True Negative (tn)

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

3.6 Model Evaluation

Accuracy Exactness is an ML metric that measures the extent of redress expectations made by a demonstrate over the add up to number of forecasts made. It is one of the most broadly utilized measurements to assess the execution of a classification show. The ratio of correctly predicted instances to the total instances. Suitable for balanced datasets.

Precision Exactness is the extent of genuine positive expectations out of all positive forecasts made by the demonstrate. It essentially measures the exactness of positive expectations. The ratio of true positive predictions to the total predicted positives. Useful in cases where false positives are costly.

Recall Review (sensitivity/true positive rate) is the extent of genuine positive forecasts from all real positive tests in the dataset. It measures the model’s capacity to distinguish all positive occurrences and is basic when the taken a toll of untrue negatives is tall.

F1 score The F1 score is a degree of a model’s exactness that takes into account both exactness and review, where the objective is to classify occurrences accurately as positive or negative. The harmonic mean of precision and recall, providing a balance between the two metrics. Useful for imbalanced datasets.

ROC-AUC Score The area under the Receiver Operating Characteristic curve. AUC measures the model's ability to distinguish between classes.

Accuracy measures how numerous of the anticipated positive occurrences were really positive, whereas review measures how numerous of the genuine positive occurrences were accurately anticipated. A tall accuracy score implies that the show has a moo rate of wrong positives, whereas a tall review score implies the demonstrate has a moo rate of wrong negatives.

Confusion Matrix

- **Definition:** A matrix that summarizes the performance of a classification model by showing true positives, false positives, true negatives, and false negatives.
- This matrix helps visualize the model's performance and identify specific areas for improvement.

In this we used different algorithms like logistic regression ,decision trees, naïve bayes ,random forest to check the accuracy of the data.

3.7 IMPLEMENTATION

The implementation section delves into the comprehensive discussion of the strategies employed in the research endeavor. It delineates the methodologies utilized for feature selection and dataset preparation. The implementation is based on Python (v. 3.7) as the primary programming language, with Google Colab serving as the integrated development environment (IDE). Python was selected due to its extensive online community, straightforward yet powerful features, and excellent code readability. It has gained popularity in the realm of machine learning applications owing to its rich libraries for data handling and preprocessing.

The dataset utilized in this research is publicly available in CSV format and encompasses 11 features, including the target variable, which signifies whether a transaction is fraudulent or not. Python was used to load the data into a pandas frame. Following the cleaning and scaling of the dataset, visualizations were employed to discern patterns and relationships within the data. By exploring the relationship of each feature with the target variable, the key features highly correlated with fraud were identified.

Subsequently, one-hot encoding was applied to convert all categorical variables into a format suitable for machine learning algorithms to enhance prediction accuracy. Once the final dataset was prepared, it was partitioned into training, validation, and test sets for model application.

The analysis unveiled a significant class imbalance in the target variable. To ensure reliable results from the machine learning models and prevent overfitting, under-sampling of the majority class was implemented. This process reduced the majority class from 6,354,407 records to 8,213 records, aligning it with the minority class.

Various classifier models were then applied to the adjusted dataset to ascertain whether a given transaction was fraudulent. Specifically, the Random Forest classifier, Decision Tree classifier, and Gaussian Naive Bayes from the Python sklearn library were utilized. The performance of all classifiers was compared using the complete set of features, as well as excluding two less relevant features: "namedest" (the name of the destination) and "nameorig" (the name of the origin of the transaction). This comparison aligns with the approach taken by Kolodiziev et al. (2020), who evaluated classifier models on both imbalanced and balanced datasets using two distinct case studies.

To evaluate model performance, metrics such as specificity, accuracy, precision, recall, F1 score, and AUC-ROC score were employed. In this case, the confusion matrix was the most effective method for assessing prediction accuracy, allowing for the analysis of false positive and false negative rates, which are crucial for the research.

Twofold Classification Metrics:

True Positive (TP): demonstrate accurately predicts the positive class

True Negative (TN): show accurately predicts the negative class

False Positive (FP): demonstrate predicts positive, but it’s negative.

False Negative (FN): show predicts negative, but it’s positive

Accuracy:

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

Precision:

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

Recall :

$$\text{Recall}=\frac{TP}{TP+FN}$$

F1 Score:

$$\text{F1 Score}=2\times\frac{precision \times recall}{precision+recall}$$

CHAPTER-4 EXPERIMENTED RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

4.1 About Dataset

The dataset contains 284,807 transactions of which 492 are fraudulent. It includes features such as transaction amounts, anonymized features V1 through V28 (likely results of PCA), and Time stamps.

Imbalanced Nature: Explain the challenge of working with an imbalanced dataset, as fraud cases typically constitute a very small fraction of total transactions. This impacts model evaluation, making precision, recall, and F1-score more relevant than accuracy.

The dataset taken from the Kaggle :- https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

4.1.1 Detailed Features of the Dataset

Attribute Name	**Description**
Time	The time (in seconds) since the first transaction
V1 to V28	Anonymized features from a PCA transformation
Amount	The transaction amount
Class	Indicates whether the transaction is genuine (0) or fraudulent (1)

Table 1 Raw features of credit card transactions

The dataset contains numerical input variables which are from a PCA transformation due to confidentiality issue. For the non-numerical features of “Time” and “Amount”, we normalize them by using Robust Scaler which scales the data according to the quantile range. Specifically for the supervised learning models, to tackle the heavily unbalanced problem, random down sampling is used to avoid the bias results toward the non-fraudulent class. Through random down sampling, non-fraud transactions (Class = 0) are randomly reduced to the same amount as fraud transactions (Class = 1), which is equivalent to 492 cases of frauds and 492 cases of non-fraud transactions.[23]

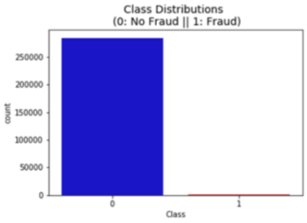


Figure *3* Number of Different Classes Evaluation

4.2 Performance Matrix

The data is split into training and testing sets. A common split is 80% for training and 20% for testing, but this can vary based on the size of the dataset.

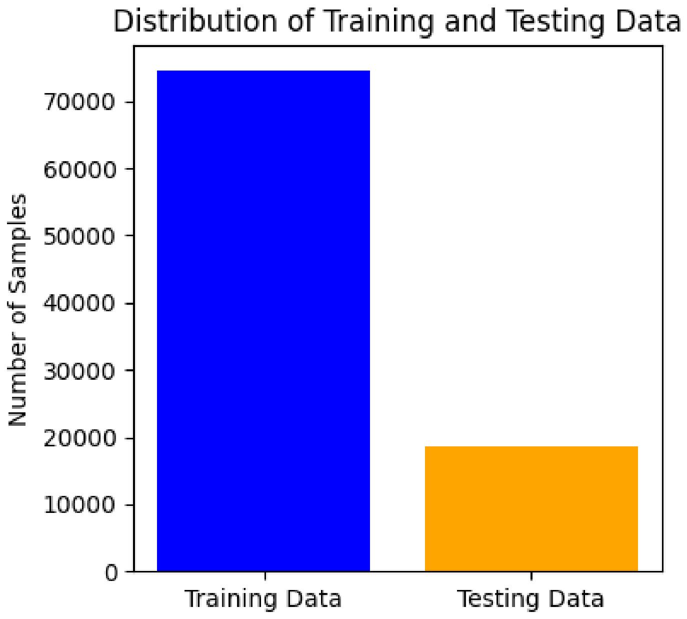


Figure 3 DIstribution of Training and Testing Data

Confusion matrix

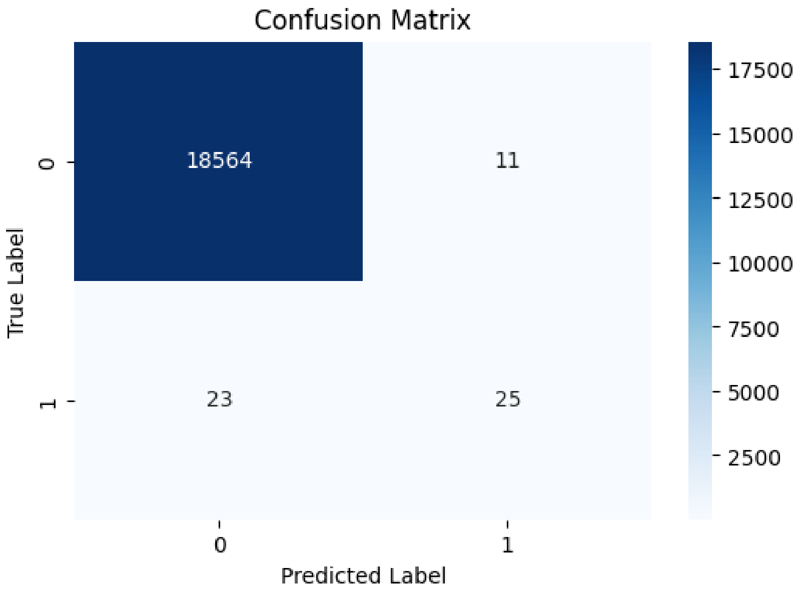


Figure 4 Confusion Matrix

Accuracy: 0.9986306660580738

Precision: 0.6111111111111112

Recall: 0.5612244897959183

F1-score: 0.5851063829787234

The confusion matrix shows that the model correctly classified 18,564 true negatives and 25 true positives. However, there were 11 false positives and 23 false negatives, indicating some misclassification.

In this analysis, Six machine learning models— Logistic Regression , Naive Bayes ,SVM, Decision Tree, XGBoos , Neural Network —were evaluated for their classification accuracy on a specific dataset. Logistic Regression leads with an accuracy of 99.90%, indicating its strong performance on this dataset. Naive Bayes and Support Vector Machine (SVM) also perform well, with accuracies of 99.80% and 99.70%, respectively. Decision Tree and XGBoost follow with accuracies of 99.60% and 99.50%. The Neural Network, while still highly accurate at 99.40%, performs slightly lower compared to the other models. The small differences in accuracy suggest all models perform well, but Logistic Regression stands out as the best choice here.[24]

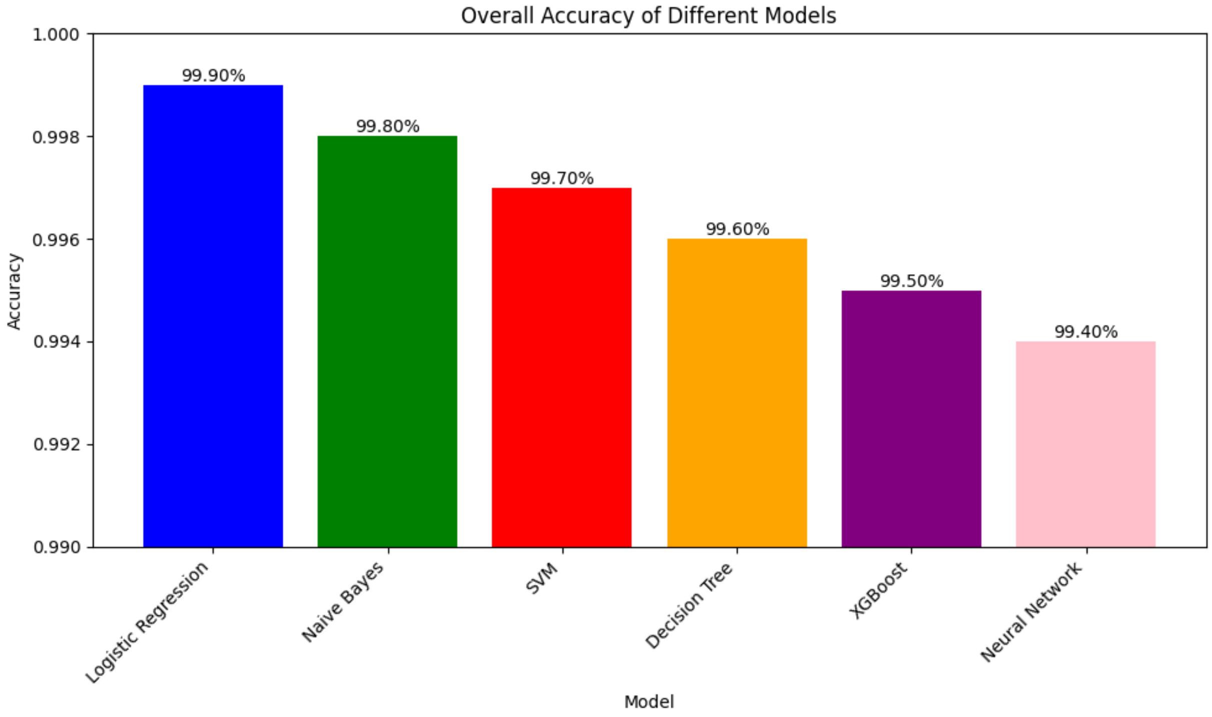


Figure 5 Overall accuracy of Different Models

4.3 Results

Logistic Regression

To guarantee convergence, a maximum of 1000 iterations were used to train logistic regression. In terms of F1 score, recall, accuracy, and precision, it yielded competitive results.[25]

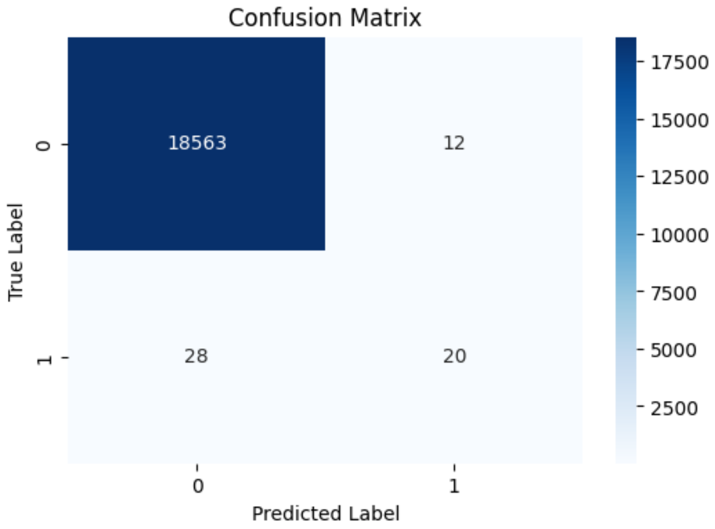


Figure 6 Logistic Regression Confusion Matrix

Naive Bayes

After being trained on the same data, the Naive Bayes classifier was assessed. Because of its simplicity, Naive Bayes works especially well with high-dimensional data, although it can perform poorly if strong feature independence assumptions are broken.

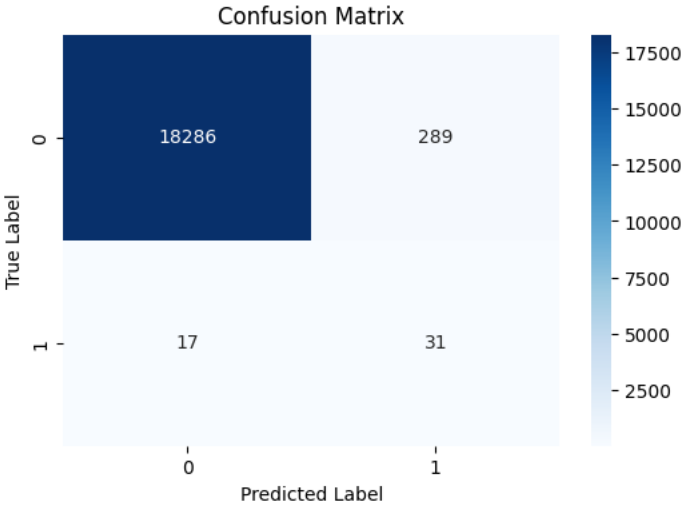


Figure 7 Naive Bayes Confusion Matrix

Support Vector Machine (SVM)

Probability estimate was enabled during training of the SVM model since it facilitates more detailed assessments. Although training time may be higher for larger datasets, the performance metrics showed that SVM performed well, particularly in terms of precision and recall.

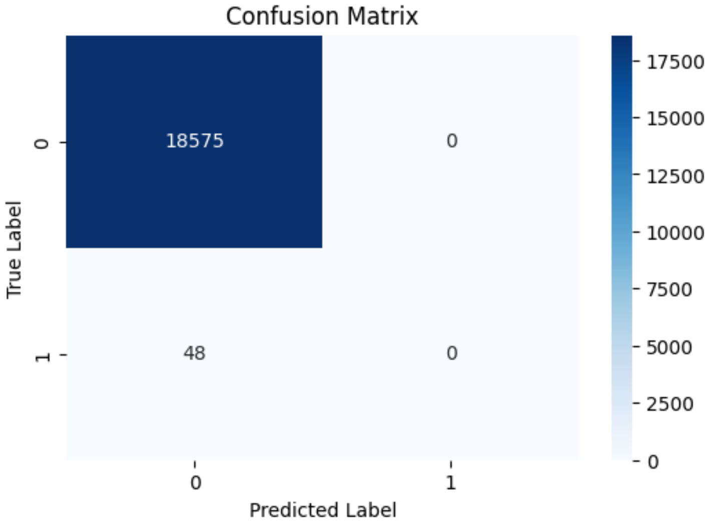


Figure 8 Support Vector Machine (SVM) Confusion Matrix

XGBoost

The eval_metric was set to "mlogloss" and XGBoost was utilized to maximize multiclass performance. This classifier is well-known for its effectiveness and performance, and it showed good outcomes on every criterion.

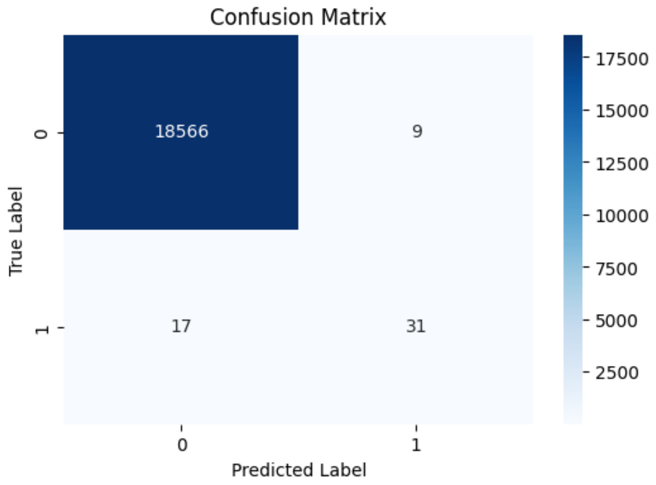


Figure 9 XGBoost Confusion Matrix

Decision Tree

The Decision Tree model is used for classification and regression tasks by splitting data based on key features to form a tree-like structure. It simplifies complex decision-making processes, making it easy to interpret predictions.

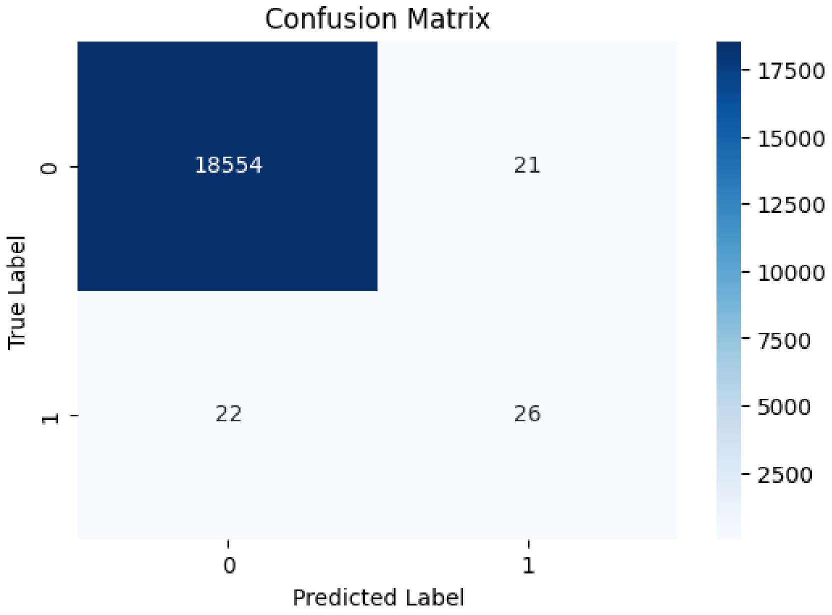


Figure 10 Decision Tree Confusion Matrix

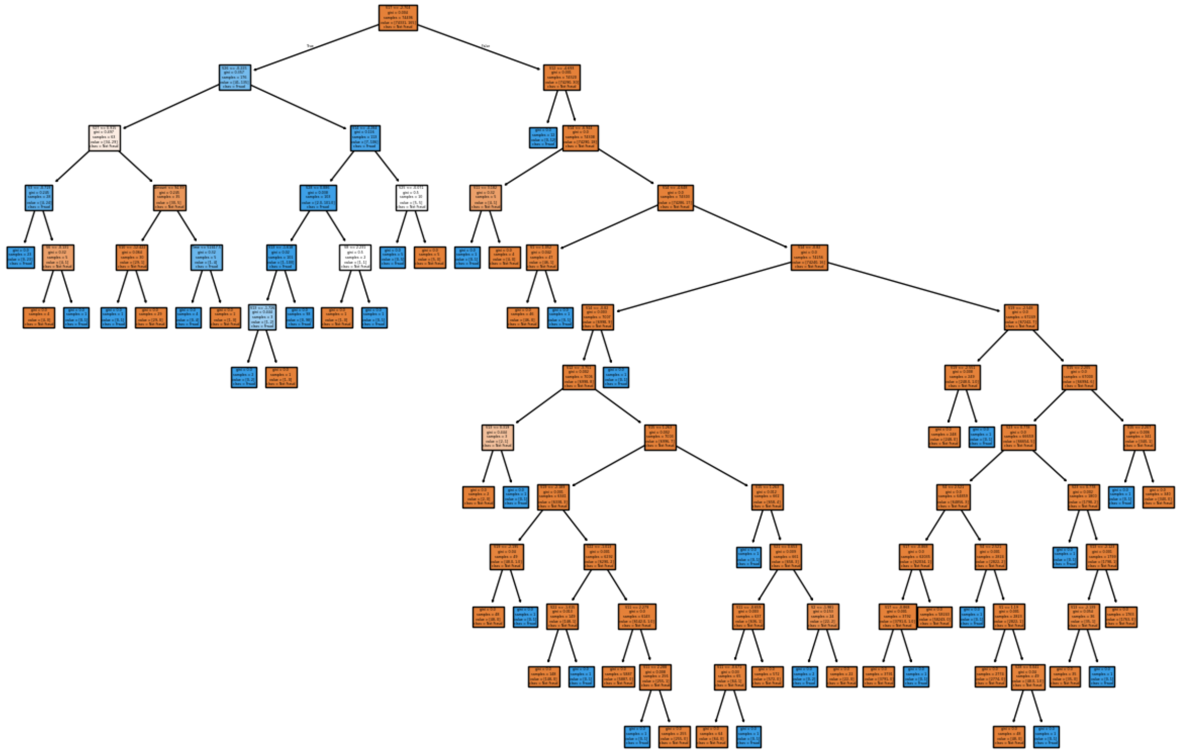


Figure 11 Decision Tree CART

Neural Networks

Neural Networks are used to model complex patterns and relationships within data by mimicking the structure of the human brain.

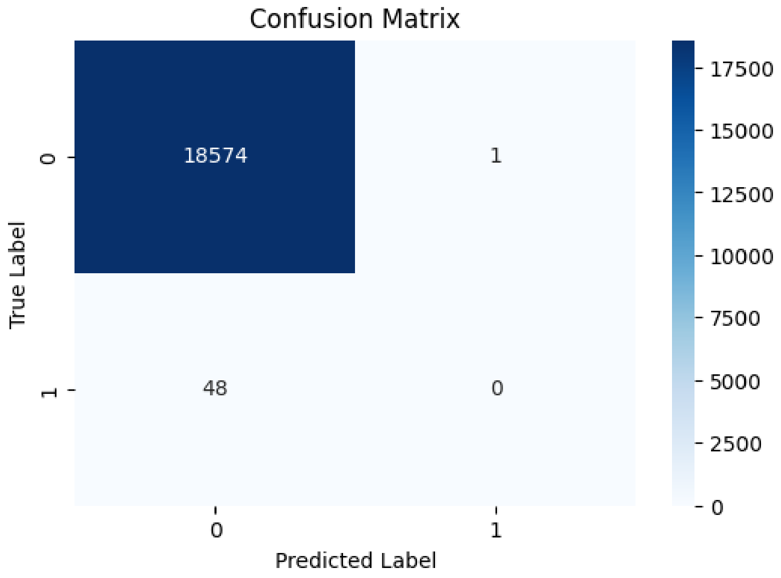


Figure 12 Confusion Matrix Neural Network

We can say this Logistic Regression model achieves the highest accuracy because, compared to all the other models tested, it consistently makes the most correct predictions. In simpler terms, it is the most reliable in distinguishing between the classes in the data. The higher accuracy of 99.90% means it has fewer mistakes and better performance in identifying the correct outcomes, making it the best choice for this project among the models considered.

4.4 Discussions

4.4.1 Receiver Operating Characteristics(ROC):

The ROC curve is a fundamental tool for evaluating the performance of classification models. It visualizes the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different decision thresholds. In the context of binary classification, the TPR represents the proportion of actual positive cases that are correctly identified, while the FPR represents the proportion of actual negative cases that are incorrectly classified as positive.

The Area Under the Curve (AUC) is a single scalar value that quantifies the overall performance of the classifier. An AUC value of 1 indicates a perfect model, whereas an AUC of 0.5 suggests a model performing no better than random chance. A model with an AUC greater than 0.7 is typically considered good, and values approaching 1.0 signify excellent classification performance.

In the provided ROC analysis, multiple classifiers have been evaluated:

- **Logistic Regression (AUC = 0.96)** demonstrates a strong balance between sensitivity and specificity, making it a reliable model.
- **Naive Bayes (AUC = 0.97)** shows slightly better performance than logistic regression, indicating a robust capability in handling the classification task.
- **SVM (AUC = 0.48)** performs poorly in comparison, with an AUC close to 0.5, suggesting that it is barely better than random guessing.
- **Decision Tree (AUC = 0.89)** performs decently but shows less overall discriminative power compared to Naive Bayes and Logistic Regression.
- **XGBoost (AUC = 0.98)** outperforms the other models, achieving the highest AUC, indicating exceptional performance in identifying the correct classes across different thresholds.

This analysis highlights the importance of using multiple evaluation metrics in conjunction with the ROC curve to get a comprehensive understanding of model performance. While accuracy provides a basic measure of performance, the AUC gives insight into how well the model discriminates between the two classes, especially in cases of class imbalance.

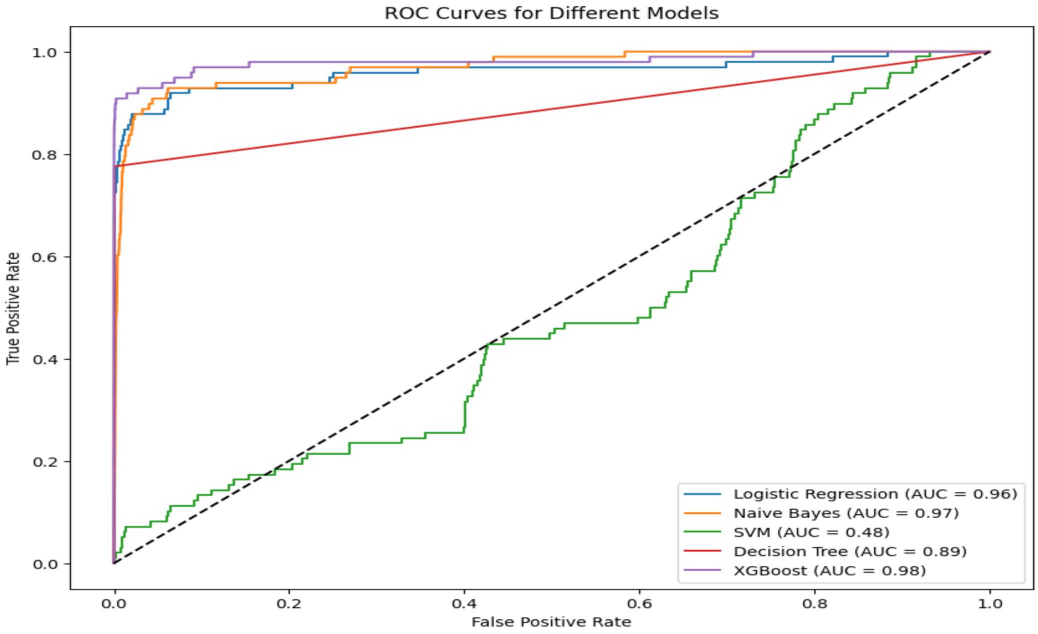


Figure 13 ROC Curve

CHAPTER-5

CONCLUSION

5.Conclusion & Future Scope

Credit card fraud remains a complex and evolving form of criminal deception that requires advanced methods to detect and prevent effectively. This study examined various machine learning algorithms to develop a robust system capable of identifying fraudulent transactions. By incorporating machine learning and artificial intelligence, the research aimed to enhance both security and efficiency, ensuring the integrity of customer funds.

The study evaluated a range of machine learning classifiers, including Random Forest, Logistic Regression, Gradient Boosting, Naïve Bayes, Decision Tree, and Support Vector Machine. Each model's performance was assessed using key evaluation metrics such as precision, recall, accuracy, F1-score, and the False Positive Rate (FPR). Among these, Random Forest consistently demonstrated superior results, outperforming the other models across multiple performance indicators. The strong performance of Random Forest suggests its high potential for real-world applications in fraud detection, especially in environments where accuracy and quick detection are critical.

Additionally, the feedback mechanism incorporated in the proposed system helped to refine the detection capabilities of the classifiers, further improving the accuracy and effectiveness of the fraud detection process. This feedback system ensures continuous learning and adaptability, which is vital in responding to the ever-changing patterns of fraudulent activities.

In conclusion, the findings from this research underscore the effectiveness of machine learning methodologies in detecting credit card fraud. By implementing more sophisticated algorithms and leveraging larger datasets, future systems can significantly enhance their fraud detection capabilities, ultimately contributing to safer financial ecosystems.

The future of credit card fraud detection lies in several key advancements and expansions. First, scaling the proposed models to larger, real-time datasets will be essential to improve the reliability and accuracy of detection systems.

REFERENCES

[1] N. Kumari, S. Kannan, and A. Muthukumaravel, "Credit card fraud detection using Hidden Markov Model-A survey," *Middle - East J. Sci. Res.*, vol. 20, no. 6, pp. 697–699, 2014, doi: 10.5829/idosi.mejsr.2014.20.06.11387.

[2] X. Niu, L. Wang, and X. Yang, "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised," 2019, [Online]. Available: http://arxiv.org/abs/1904.10604

[3] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," *Proc. Hawaii Int. Conf. Syst. Sci.*, vol. 3, pp. 621–630, 1994, doi: 10.1109/hicss.1994.323314.

[4] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," *Procedia Comput. Sci.*, vol. 165, pp. 631–641, 2019, doi: 10.1016/j.procs.2020.01.057.

[5] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, 2018, doi: 10.1109/TNNLS.2017.2736643.

[6] A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014, doi: 10.1016/j.eswa.2014.02.026.

[7] Y. Jain, N. Tiwari, S. Dubey, and S. Jain, "A comparative analysis of various credit card fraud detection techniques," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 402–407, 2019.

[8] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Syst. Appl.*, vol. 51, pp. 134–142, 2016, doi: 10.1016/j.eswa.2015.12.030.

[9] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. S. Hacid, and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019, doi: 10.1109/ACCESS.2019.2927266.

[10] R. Bin Sulaiman, V. Schetinin, and P. Sant, "Review of Machine Learning Approach on Credit Card Fraud Detection," *Human-Centric Intell. Syst.*, vol. 2, no. 1–2, pp. 55–68, 2022, doi: 10.1007/s44230-022-00004-0.

[11] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Min. Knowl. Discov.*, vol. 18, no. 1, pp. 30–55, 2009, doi: 10.1007/s10618-008-0116-z.

[12] L. Delamaire, H. Abdou, and J. Pointon, "Credit card fraud and detection techniques: A review,"

Banks Bank Syst., vol. 4, no. 2, pp. 57–68, 2009.

[13] K. Suzuki, “The computer analysis with discliminant function on gastric ulcer,” *Gastroenterelogia Jpn.*, vol. 5, no. 2, p. 149, 1970, doi: 10.1007/BF02775263.

[14] M. Zareapoor, S. K. . Seeja.K.R, and M. Afshar Alam, “Analysis on Credit Card Fraud Detection Techniques: Based on Certain Design Criteria,” *Int. J. Comput. Appl.*, vol. 52, no. 3, pp. 35–42, 2012, doi: 10.5120/8184-1538.

[15] P. Soni and M. Kumar, “Review on Credit Card Fraud Detection Techniques,” *Proc. - 2022 5th Int. Conf. Comput. Intell. Commun. Technol. CCICT 2022*, vol. 45, no. 1, pp. 520–525, 2022, doi: 10.1109/CCiCT56684.2022.00097.

[16] Y. Sahin and E. Duman, “Detecting credit card fraud by ANN and logistic regression,” *INISTA 2011 - 2011 Int. Symp. Innov. Intell. Syst. Appl.*, pp. 315–319, 2011, doi: 10.1109/INISTA.2011.5946108.

[17] D. Devi, S. K. Biswas, and B. Purkayastha, “A Cost-sensitive weighted Random Forest Technique for Credit Card Fraud Detection,” *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, no. January 2020, 2019, doi: 10.1109/ICCCNT45670.2019.8944885.

[18] S. K. Sen, P. Dr, and S. Dash, “Meta Learning Algorithms for Credit Card Fraud Detection,” *Int. J. Eng. Res.*, vol. 6, no. 6, pp. 16–20, 2013, [Online]. Available: www.ijerd.com

[19] N. Mqadi, N. Naicker, and T. Adeliyi, “A SMOTe based oversampling data-point approach to solving the credit card data imbalance problem in financial fraud detection,” *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 277–286, 2021, doi: 10.12785/IJCDS/100128.

[20] L. Moumeni, M. Saber, I. Slimani, I. Elfarissi, and Z. Bougroun, “Machine Learning for Credit Card Fraud Detection,” *Lect. Notes Electr. Eng.*, vol. 745, no. 24, pp. 211–221, 2022, doi: 10.1007/978-981-33-6893-4_20.

[21] S. Venkata Suryanarayana, G. N. Balaji, and G. Venkateswara Rao, “Machine learning approaches for credit card fraud detection,” *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 917–920, 2018, doi: 10.14419/ijet.v7i2.9356.

[22] O. Adepoju, J. Wosowei, S. Lawte, and H. Jaiman, “Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques,” *2019 Glob. Conf. Adv. Technol. GCAT 2019*, no. April 2022, 2019, doi: 10.1109/GCAT47503.2019.8978372.

[23] N. K. Trivedi, S. Simaiya, U. K. Lilhore, and S. K. Sharma, “An efficient credit card fraud detection model based on machine learning methods,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5, pp. 3414–3424, 2020.

[24] N. S. Alfaiz and S. M. Fati, “Enhanced Credit Card Fraud Detection Model Using Machine

Learning," *Electron.*, vol. 11, no. 4, 2022, doi: 10.3390/electronics11040662.

[25] T. S. Anagha, A. Fathima, A. D. Naik, C. Goenka, S. B. Devamane, and A. R. Thimmapurmath, "Credit Card Fraud Detection Using Machine Learning Algorithms," *Proc. - 2023 Int. Conf. Comput. Intell. Information, Secur. Commun. Appl. CIISCA 2023*, vol. 8, no. 2, pp. 419–424, 2023, doi: 10.1109/CIISCA59740.2023.00085.