CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

by

Ms. Shweta S. Patil (CT22055)

Ms. Sanjana A. Laxne (CT22011)

Ms. Gayatri S. Bhandarkar (CT22041)

Ms. Ruchita R. Yeyyawar (CT21027)

Ms. Achal R. Shende (CT22045)

Bachelor of Technology in Computer Technology

Under the guidance of

Ms. Vaishali Malekar

Assistant Professor

DEPARTMENT OF COMPUTER TECHNOLOGY

KAVIKULGURU INSTITUTE OF TECHNOLOGY AND SCIENCE

RAMTEK, NAGPUR, MAHARASHTRA, INDIA-441 106

ABSTRACT

Credit card fraud has become one of the most critical issues in the financial sector due to the increasing volume of digital transactions. Fraudulent activities not only lead to major monetary losses for banks and customers but also reduce trust in online payment systems. This project presents a Credit Card Fraud Detection System that uses machine learning algorithms to identify fraudulent transactions. The dataset used in this project work is highly imbalanced, with very few fraudulent cases compared to genuine ones. To address this challenge, data preprocessing techniques such as resampling, feature scaling, and normalization Itre applied. Several machine learning models, including Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting, Performance was measured using precision, recall, F1score, and AUCROC curve. The experimental results show that ensemble models provide better performance and higher fraud detection rates than traditional models. The proposed system achieves high accuracy and precision, reducing false positives and ensuring secure financial transactions. It highlights the potential of machine learning in strengthening fraud prevention mechanisms and can be applied to real-world banking systems for enhanced security.

KEYWORDS: Credit Card Fraud, Fraud Detection, Machine Learning, Data Imbalance, Classification, Financial Security, Anomaly Detection.

CHAPTER 1 INTRODUCTION

1.1 Preamble

Credit card fraud refers to the unauthorized use of funds through credit or debit cards. Various techniques are used to steal the information needed to perform illegitimate online card-not-present transactions, as Itll as physical transaction.

These methods include skimming, phishing, malware, and man-in-the-middle attacks. Given that credit card fraud losses exceeded 32 billion US dollars globally in 2023 alone, there is an urgent need to develop and implement effective strategies for fraud detection and investigation. A typical fraud detection and manual investigation framework is presented. In most cases, banks and card networks evaluate transactions using rule-based systems and/or predictive models to identify fraud patterns assign risk scores to the transactions. If the risk score for a transaction exceeds a certain threshold, an alert is flagged for manual investigation, indicating that the transaction requires investigation by a fraud analyst.

These investigations improve analysts understanding regarding the nature of the transaction, while maintaining costumers' trust and ensuring compliance with regulations. Credit card fraud investigations mainly involve gathering and examining evidence, and at the end of an investigation, a detailed report describing the investigation of the case is produced. Such reports are used by a company's senior management, law enforcement agencies, and legal teams to make informed decisions and take appropriate actions. The results of an investigation are usually fed back to retrain fraud detection systems. Investigation of potential fraud cases requires careful attention, specialized knowledge, and precise documentation of the outcomes.

Since the number of investigated cases and their high demands can quickly exhaust workforce resources, there is a need for automation. Today, artificial intelligence and large language models are used in a variety of tasks that require advanced analysis capabilities, such as root cause analysis cybersecurity and smart policing. In this article, It investigate the extent to which LLM analysis capabilities help automate diverse tasks traditionally performed by human analysts during credit card fraud alert investigations. It presents a fraud analyst assistant framework that employs LLMs to automate fraud alert investigations. The FAA framework includes tools and LLM agents used in a series of investigative steps for generating code, retrieving data from a source, analyzing and visualizing the data.

1.2 Motivation

Credit card usage has become one of the most common modes of financial transactions worldwide due to its convenience, speed, and security features. HoItver, with this growing dependency comes the increasing threat of fraudulent transactions, which can cause severe financial losses for both customers and financial institutions. Detecting fraud is a critical challenge because fraudulent transactions are rare compared to legitimate ones, and fraudsters constantly change their strategies to bypass detection systems. The system can be enhanced with real-time detection, integration with banking APIs, mobile app alerts, and adaptive AI

models to handle evolving fraud techniques and helps in protecting individuals and institutions from financial losses, ensuring safer digital payment systems.

1.3 Aim

The aim of this project is to develop an intelligent Credit Card Fraud Detection System that can effectively identify fraudulent transactions by analyzing customer spending patterns and detecting unusual behavior. With the rapid growth of online payments and ecommerce, financial institutions face increasing risks of fraud.

1.4 Objectives

- To develop a technique using machine learning approaches that will detect the fraud transaction via credit card.
- To reduce the number of legitimate transactions flagged as fraud to improve customer convenience.
- To implement a system capable of detecting suspicious activities instantly to prevent financial losses.
- To build a reliable system that strengthens customer confidence in digital payments.

1.5 Organization of the Report

This report is organized into eight chapters. Chapter one contains introduction provides. An overview of the project, its objectives, and the problem statement. Chapter two includes prior art reviews existing solutions and identifies their limitations. Chapter three describes literature review. Analyzes previous research and related works. Chapter four contains tools and technologies describes the software and frameworks used to develop the system. Chapter five contains proposed approach and system architecture explains the design and structure of the system, including key components and workflow. Chapter six elaborate on implementation details which focus on the development process and how the system was built. Chapter seven presents results and discussion which describes the outcomes of the system and evaluates its performance. Chapter eight summarizes conclusion the findings and suggests future improvements.

CHAPTER 2 PRIOR ART

This chapter explains the prior art which covers the patents which are related to the credit card fraud detection.

2.1 Systems and Methods for Fraud Detection Using Machine Learning

Application Number: US20180314765A1

The patent describes a rapid increase in online payments and credit card usage has led to a parallel rise in fraudulent activities. Traditional fraud detection methods, primarily based on static rule sets and manual verification, are no longer sufficient to address the complexity of modern financial crimes. To address this challenge, the patent US20180314765A1 introduces a novel approach that leverages supervised machine learning algorithms to build intelligent systems capable of learning from historical data and detecting fraud in real-time.

This invention is particularly relevant in domains such as banking, e-commerce, and payment gateways, where millions of digital transactions occur daily. By training models on previous fraudulent and non-fraudulent cases, the system ensures more accurate identification of anomalies that may indicate fraud.

Technical Approach Data Collection and Feature Extraction Historical transaction records are gathered, containing both legitimate and fraudulent data points.

Features such as transaction amount, merchant type, device information, transaction time, location, IP address, and user behavior patterns are extracted.

Model Training system uses supervised ML algorithms like Decision Trees, Random Forests Gradient Boosting, or Support Vector Machines. The training dataset is labeled, meaning each transaction is tagged as either "fraud" or "genuine." The algorithm learns to map input features to the correct output class. During real-time transactions, the trained model processes incoming data and classifies it as either suspicious or legitimate

Transactions flagged as high-risk are either rejected, held for manual review, or subjected to additional verification example OTP, customer confirmation. The fraud detection model is not static. It continuously updates itself by retraining on new data.

2.2 Credit Card Fraud Detection Using Artificial Intelligence

Application Number: US20190131392A1

The patent describes a Credit card fraud is one of the most pressing issues in the financial industry, leading to billions of dollars in global losses annually. Traditional fraud detection mechanisms rely heavily on pre-defined rules, which often fail to adapt to evolving fraud patterns. The patent US20190131392A1 presents a solution that uses Artificial Intelligence and Deep Learning techniques to enhance fraud detection accuracy.

Unlike static systems, this invention employs Neural Networks and advanced learning algorithms to dynamically analyze transaction data and detect suspicious activity. This approach offers a robust and scalable method to prevent fraud across banking, online transactions, e-commerce platforms, and payment gateways.

Transaction records are collected, including user identity, transaction amount, merchant details, device/IP address, and geolocation. Preprocessing involves cleaning the data, handling missing values, and balancing datasets fraudulent vs. legitimate cases. The system applies AI models Artificial Neural Networks, Deep Learning, and Ensemble Models. Features such as spending patterns, behavioral trends, and unusual transaction activity are used to detect anomalies. When a transaction request is initiated, the AI engine analyzes it in milliseconds. Highrisk transactions are flagged and subjected to additional security measures OTP, biometric check, or manual review.

2.3 Machine Learning Models Detecting Transaction Fraud

Application Number: US20200294991A1

The patent describes a system that Fraudulent financial transactions cause severe economic losses worldwide. Traditional fraud detection systems depend on static rules that struggle to identify complex and evolving fraudulent activities.

The patent introduces a machine learning-based system that detects transaction fraud by training multiple models on diverse data subsets. By leveraging ensemble techniques and multi-model learning, the system achieves higher prediction accuracy while minimizing false positives. This innovation provides a scalable and adaptive solution capable of detecting fraud across different domains such as banking, ecommerce, and mobile payments. Transactions are divided into subsets based on factors such as time, location, amount, and device type. Each subset is used to train a specialized fraud detection model. Various models example Decision Trees, Random Forests, Gradient Boosting, and Neural Networks are trained on different subsets. Each model learns unique fraud patterns within its assigned dataset.

2.4 Adaptive Fraud Detection Using Clustering and Classification

Application Number: US20160042382A1

The patent describes a with the growth of online payments and digital banking, fraudsters are constantly developing new ways to bypass traditional detection systems. Static rule-based systems often fail to identify emerging fraud techniques, while single classification models sometimes miss unusual yet fraudulent behaviour. The patent addresses this challenge by presenting a hybrid fraud detection framework that integrates clustering algorithms unsupervised learning with classification models supervised learning. This combination enables the system to identify both known fraud patterns and previously unseen anomalies, making fraud detection more adaptive and reliable.

Time, amount, merchant ID, geolocation, device, and behavioral patterns are cleaned and prepared. Features are normalized to ensure accurate clustering and classification. Clustering for anomaly detection unsupervised learning Transactions are grouped into clusters based on similarity. Outliers transactions that do not belong to any cluster are flagged as potentially fraudulent.

It helps to detect new or unknown fraud techniques that the system has not encountered before. Classification for Known Fraud Supervised Learning. A supervised ML classifier example Decision Tree, SVM, Neural Network is trained on labeled data.

The classifier recognizes fraud patterns previously identified in historical datasets. Adaptive Integration the outputs from clustering and classification are combined. The system adapts to new behaviours while retaining the ability to detect established fraud.

2.5 Credit Transaction Scoring Using Real-Time Machine Learning Models

Application Number: US2010180010A1

Fraudulent credit card transactions present a serious challenge to banks and financial institutions, leading to billions of dollars in losses annually. Traditional fraud detection systems rely heavily on static rules or historical models, which often fail to keep up with real-time fraud patterns.

The patent introduces a real-time fraud detection framework that assigns a fraud score to each credit transaction using machine learning models. Unlike static systems, this approach adapts dynamically to incoming data, enabling institutions to block fraudulent transactions almost instantly while reducing inconvenience to genuine customers.

Technical Approach Transaction Data Collection Captures real-time attributes of each transaction for example amount, merchant ID, device ID, geolocation, time. Integrates both user behavioral history and external fraud indicators. Real-Time Feature Extraction Extracts features such as frequency of transactions, spending habits, unusual device use, and sudden geographic shifts. Enables context aware fraud scoring.

Machine Learning Model Training Models are trained using historical transaction datasets labeled as fraudulent or genuine. Algorithms include logistic regression, decision trees, neural networks, and ensemble methods.

Real-Time Scoring Mechanism Each new transaction is evaluated by the ML model. A fraud score is generated in real time, indicating the probability of fraud. Transactions above a certain threshold are flagged for review or automatically declined. System updates its models continuously based on newly confirmed fraud cases. Improves adaptability against evolving fraud strategies.

CHAPTER 3 LITERATURE REVIEW

This chapter reviews the various literatures that have been proposed by various authors to gain more information and knowledge about the domain of the project.

Akansha Bansal and Hitendra Garg (2023) focused on machine learning has emerged as a dominant approach for credit card fraud detection, providing more adaptive and accurate alternatives to rule-based systems. Traditional models such as logistic regression and support vector machines Itre among earliest applied methods, offering interpretability and efficiency but often struggling with non-linear fraud patterns.

To overcome these limitations, decision trees and random forests became popular due to their ability to capture complex feature interactions and handle heterogeneous transaction data.Recent studies show that ensemble methods such as XGBoost and LightGBM outperform many classical algorithms in fraud detection tasks by combining multiple Itak learners and improving predictive poItr. For example, Dal Pozzolo et al. (2018) emphasized that tree-based ensembles remain strong baselines even under challenging conditions such as extreme class imbalance and concept drift.In addition, semi-supervised learning approaches have been employed to detect fraud when labeled data is scarce. These methods learn from large volumes of normal transactions and flag unusual patterns as potential fraud. Hybrid learning strategies, which combine supervised models with anomaly detection uniques, also been investigated to improve detection accuracy and reduce false positives.

In recent years, deep learning approaches have gained popularity in credit card fraud detection due to their ability to capture complex, non-linear relationships in data. Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) are particularly effective for sequential transaction data, as they can learn temporal patterns in customer spending habits, while convolutional neural networks (CNNs) are used to extract spatial features from transaction attributes. More recently, graph neural networks (GNNs) have been explored to detect organized fraud by modeling the relationships betIten different entities such as customers, merchants, and locations.

These deep learning models are capable of processing large transactionin real-time, though mey demand,processing large-scale transaction data in real-time, though they demand significant computational poItr and careful tuning to achieve optimal performance.

Despite their strengths, machine learning-based approaches face several challenges. Fraud datasets are highly imbalanced, with genuine transactions vastly outnumbering fraudulent ones, which can cause outcomes. Additionally, fraudsters constantly change their strategies, leading to a phenomenon known as drift, where models trained on historical data may quickly become outdated.

Harish Paruchuri (2022) emphasized on one of the most prominent challenges in credit card fraud detection is the problem of class imbalance. In real-world datasets, the number of fraudulent transactions is extremely small compared to the vast majority of legitimate ones. For example, out of millions of transactions processed daily, only a very small fraction, sometimes less than 0.5%, may be fraudulent.

This disproportion causes traditional machine learning models to learn biased patterns in favor of the majority class, leading to misleadingly accuracy but very poor detection of actual frauds. In such cases, a classifier may achieve 99% accuracy by simply predicting every transaction as legitimate, yet it fails at its primary objective-catching fraudulent activity.Since fraud detection systems are highly sensitive applications where the cost of a missed fraud is significantly higher than a false alarm, effectively handling class imbalance becomes a crucial requirement. Various strategies have been proposed in research and practice to tackle this imbalance, which are broadly categorized into datalevel methods, algorithm-level methods, cost-sensitive and hybrid approaches.

At the data level, resampling techniques are the most common solutions. Under sampling methods reduce the size of the majority class by removing a portion of legitimate transactions, which helps balance the dataset but may also result in the loss of potentially valuable information. Oversampling, on the other hand, increases the size of the minority class by replicating fraud cases, but this can lead to overfitting since models might learn repeated patterns instead of generalizable fraud behavior.

To overcome this, advanced synthetic data generation techniques such as the Minority Oversampling Technique SMOTE and its variants have been widely applied. SMOTE creates artificial samples by interpolating betIten existing fraud cases, thereby enriching the minority class with more diverse patterns.Ensemble methods like Random Forest and Gradient Boosting are particularly effective since they aggregate multiple Itak learners and can emphasize minority class detection through balanced sampling or Itighted voting.Similarly, anomaly detection algorithms such as Isolation Forest or One-Class SVM treat fraud as an outlier and perform Itll under skeItd distributions. Another popular strategy is cost-sensitive learning, where different misclassification costs are assigned to fraud and non-fraud cases.

By penalizing false negatives more heavily, these models prioritize the correct detection of fraudulent transactions even at the expense of more false positives.Recall (or sensitivity) is particularly important, as it measures the ability of a model to correctly identify fraudulent cases, while precision helps in minimizing false alarms that could inconvenience genuine customers. A balanced trade-off betIten these metrics ensures that the model not only identifies most fraudulent transactions but also maintains trust and usability.In

summary, handling class imbalance is a vital step in building effective fraud detection systems. Ignoring the imbalance can render even sophisticated algorithms ineffective, while appropriate balancing strategies can significantly improve detection rates.

L. Zhang and Y. Liu (2023) studied on Fraudulent transactions often exhibit irregular patterns that differ significantly from normal customer behavior. Since fraudsters constantly evolve their strategies, labeled data for all possible fraudulent activities is usually unavailable. This makes anomaly detection and unsupervised learning particularly effective, as they identify suspicious behavior without prior knowledge of fraud labels. Anomaly detection focused on spotting rare and unusual events in financial datasets. In the context of fraud detection, anomalies may include unusually high transaction amounts, purchases from uncommon locations, multiple rapid transactions in a short span, or deviations from a customer's typical spending Since fraudulent activities are statistically rare compared to legitimate ones, anomaly detection algorithms are designed to highlight these deviations for further investigation. Unsupervised techniques are especially important because they do not require labeled datasets. In fraud detection, labeled data is often scarce, delayed, or incomplete due to the time required for banks and customers to verify fraudulent claims. Unsupervised models instead learn the structure and distribution of normal transactional behavior and then identify deviations.

statistically rare compared to legitimate ones, anomaly detection algorithms are designed to highlight these deviations for further investigation. Unsupervised techniques are especially important because they do not require labeled datasets. In fraud detection, labeled data is often scarce, delayed, or incomplete due to the time required for banks and customers to verify fraudulent claims. Unsupervised models instead learn the structure and distribution of normal transactional behavior and then identify deviations.

Some widely used approaches include Clustering Methods These group transactions based on similarity. Outliers or small clusters that do not match normal behavior patterns may represent potential fraud. Autoencoders or Neural Networks: By learning to compress and reconstruct normal transactions, autoencoders can detect anomalies when the reconstruction error for a transaction is significantly higher than expected. Isolation Forests this algorithm isolates anomalies by randomly partitioning Since anomalies are easier to isolate than normal points, they can be detected efficiently. Principal Component Analysis by reducing the dimensionality of transaction data, PCA highlights directions where variance is unusually high, signaling potential anomalies. One-Class SVM trains on normal transactions and classifies new data points based on their similarity to the learned distribution, flagging dissimilar points as anomalies.

R. Kumar and A. Tiwari (2021) elaborated on recent years, deep learning has emerged as one of the most poItrful paradigms for credit card fraud detection, offering superior accuracy compared to traditional machine learning methods due to its ability to automatically learn complex, non-linear patterns from large amounts of data. Fraudulent transactions are often hidden within massive streams of legitimate activity, making it extremely to identify anomalies with simple rule-based or shallow learning techniques. Deep learning architectures such as artificial neural networks, recurrent neural networks, long short-term memory networks. Convolutional neural networks, and autoencoders have been widely applied to fraud detection tasks because of their capacity to capture temporal dependencies, extract hierarchical features, and detect subtle deviations

in user One of the primary advantages of deep learning in this is learning, where the model automatically extracts relevant frequency, amount, location, and merchant type-without relying heavily on handcrafted features designed by domain experts.

Particularly effective in environments where fraud patterns evolve rapidly and manually engineered rules quickly become obsolete are transaction features-such as frequency, amount, location, and merchant type-without relying heavily on handcrafted features designed by domain experts.

This makes deep learning particularly effective in environments where fraud patterns evolve rapidly and manually engineered rules quickly become obsolete. Feed forward neural networks and deep multilayer perceptron have been extensively used to classify transactions as fraudulent or genuine.

These networks are capable of capturing non-linear interactions betIten features that models might overlook. RNNs and LSTMs, on the other hand, are particularly Itllsuited for sequential modeling, as fraud often follows behavioral sequences, such as a sudden spike in in unusual locations or abnormal frequency of purchases. By remembering long-term dependencies in transaction sequences, LSTMs can provide early warnings of suspicious activity before major fraud Similarly, CNNs, which are traditionally used in image recognition, have been adapted to fraud detection by treating transaction matrices or embeddings as spatial data, allowing them to detect local feature patterns that represent unusual spending clusters. Autoencoders, another widely used deep learning technique, are effective for unsupervised fraud detection by reconstructing normal transaction unusually high, indicating potential fraudulent activity. behavior and identifying anomalies when reconstruction errors are unusually high, indicating potential fraudulent activity. Another significant aspect of deep learning in fraud detection is its scalability and adaptability.

Financial institutions millions of transactions per day, and deep learning models can be trained on such large accuracy. Furthermore, deep learning can be integrated with big data frameworks and real-time monitoring to provide near instantaneous fraud alerts. Hybrid models that combine deep learning with other approaches, such as gradient boosting or graph neural networks, have also been proposed to capture both transactional features and betIten entities example cardholders, merchants, and devices.

Despite their face several such as class datasets to improve detection success, deep learning approaches imbalance where fraudulent transactions constitute only a small of the dataset, making models biased towards predicting legitimate transactions. Interpretability is another concern since deep learning models are often seen as "black boxes" making it difficult for financial institutions to explain why a certain transaction was flagged as fraudulent.

In conclusion, deep learning offers a highly effective and adaptive for detecting card fraud in today's digital payment ecosystem. Its ability to process vast amounts of high- dimensional data, learn intricate transaction patterns, and adapt to evolving fraud strategies makes it indispensable for modern fraud prevention systems.

As fraudulent schemes continue to grow more sophisticated, integrating deep learning with explainable Al, graph-based models, and real-time big data analytics will be critical in building robust, scalable, and trustworthy fraud detection systems

S. Roy and A. Chakraborty (2020) focused on graph-based and network-oriented models represent data as a graph structure, where entities are model as nodes and their relationships as edges. Unlike traditional machine learning that often assumes independent samples, graph-based models explicitly capture dependencies, interactions, and structures within data. These models are widely applied in fraud detection, recommendation systems, cybersecurity, and social networks. Graph Models Static Graph Models Represent relationships at a single point in time. Example A network of credit cards and merchants connected by transactions. Dynamic Graph Models Capture evolving relationships over time. Example Monitoring transaction sequences in real time to detect sudden anomalies. Heterogeneous Graph Contain multiple types of nodes and edges. In fraud detection, nodes can represent customers, merchants, devices, and IP addresses. Key Techniques Graph Theory-Based Metrics Centrality measures degree, betItenness, closeness to detect suspicious nodes. Detection to identify clusters of normal fraudulent activity. Graph neural networks (GNNs) Deep learning models extended to graphs. Aggregate features from neighbors to learn node/edge/graph-level embeddings. Variants graph convolutional networks graph attention networks graph SAGE. Network propagation models. Subgraph and pattern mining identifying unusual substructures that indicate collusion or fraud.

CHAPTER 4

PROPOSED APPROACH AND SYSTEM ARCHITECTURE

The proposed approach for Credit Card Fraud Detection is based on a systematic machine learning and deep learning pipeline that processes transaction data, extracts meaningful features, trains predictive models, and generates real-time fraud alerts.

The architecture is designed to ensure accuracy, efficiency, scalability, and security while handling highly imbalanced datasets, where fraudulent transactions account for less than of all transactions.

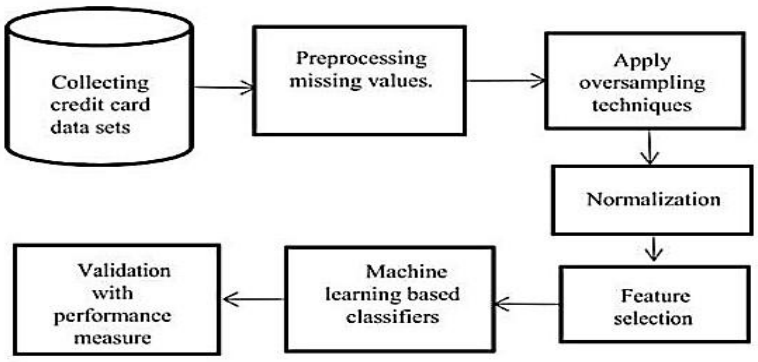


Figure 4.1. Data flow diagram of credit card fraud detection

Figure 4.1 shows data flow diagram of credit card fraud detection using machine learning and focus on performance measure with its validation.

The system starts by collecting credit card transaction logs from banks, financial institutions, or payment gateways. The dataset includes fields like Transaction ID, Amount, Time, Location, Merchant Type, Device Information, User ID, etc. Both fraudulent and genuine transactions are included to train the system. Data Preprocessing Raw transaction data is usually noisy, incomplete, and unbalanced, so preprocessing is essential.

Data Collection

Validation techniques in machine learning are used to get the error rate of the Machine Learning model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. HoItver, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to finetune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your models.

A number of different data cleaning tasks using Python‘s Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Data collection effective data collection for credit card fraud detection begins with gathering diverse, high-quality transaction records from bankings, payment processors, and merchant logs—capturing fields such as transaction timestamp, amount, merchant category, location, device/browser fingerprint, cardholder profile, and authorization results. It’s crucial to include both labeled examples of confirmed fraudulent and legitimate transactions (from chargeback records, investigator tags, or known fraud campaigns) while carefully handling class imbalance through oversampling or targeted logging of rare fraud types. Preserve user privacy and comply with regulations (e.g., PCI-DSS, local data-protection laws) by anonymizing or tokenizing personally identifiable information and securely storing audit trails. Enrich raw transactions with derived features and external signals like watchlists or device reputation. Finally, ensure continuous collection and feedback loops feeding model predictions and investigator outcomes back into the dataset—to keep models current against evolving fraud tactics and to support rigorous evaluation using time aware splits that reflect realworld deployment.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values,

and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it important to understand the sources of missing data

Data Validation

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making. Data validation for credit card fraud detection is a crucial step to ensure that the collected dataset is accurate, reliable, and suitable for training machine learning models. Since fraud detection systems rely heavily on transaction-level details, any errors, inconsistencies, or missing values can lead to biased predictions or poor detection rates. Validation begins with checking for completeness ensuring all essential fields such as transaction ID, time, amount, merchant category, and location are present. Data consistency is then verified by detecting anomalies like negative transaction values, duplicate entries, or mismatched timestamps. Outlier analysis is performed to separate genuine unusual spending behavior from suspicious fraud patterns. Since fraud data is highly imbalanced, validation also involves confirming that labeling of fraudulent and non-fraudulent cases is correct and free from misclassification.

Additionally, data integrity checks ensure no tampering has occurred during collection, while normalization and standardization help maintain uniform formats across sources. Finally, privacy validation is conducted to ensure compliance with legal standards by anonymizing sensitive information. A robust data validation process builds trust in the dataset and directly improves the performance and fairness of fraud detection models.

Exploration Data Analysis of Visualization

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more.

With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

Data Cleaning Missing values and duplicate entries are removed. Normalization: Features like transaction amount are scaled to avoid bias. Balancing Dataset: Since fraud cases are rare, techniques like synthetic minority oversampling technique are applied to ensure models do not ignore fraud transactions. Encoding Convert categorical data merchant type, device into numerical values for ML algorithms. Feature Extraction and Engineering Fraudulent transactions often follow unusual patterns. Feature engineering creates new attributes that help models detect fraud Transaction Amount Analysis Unusually high-value transactions may indicate fraud. Frequency of transactions: Too many transactions in a short time may be suspicious. Geographical distance of two transactions occur in different countries within minutes, it may be fraud.

Device and Location Change Sudden change in device/location from usual behaviour. These features strengthen the model's ability to differentiate betIten fraud and normal behavior. Model Training (Machine Learning and Deep Learning) Machine Learning Models Logistic Regression. Decision Trees and Random Forests (handle complex fraud behavior. Gradient Boosting for high accuracy.

Deep Learning Models: Artificial Neural Networks detect hidden fraud patterns. Long Short-Term Memory models can capture sequential fraud behaviors over time. Training is performed on pre-processed and balanced data. The proposed architecture provides a robust, intelligent, and scalable fraud detection framework.

By combining data preprocessing, feature engineering, machine learning, and real-time alert generation, the system ensures that fraudulent transactions are detected quickly and accurately. Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set.

In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner.

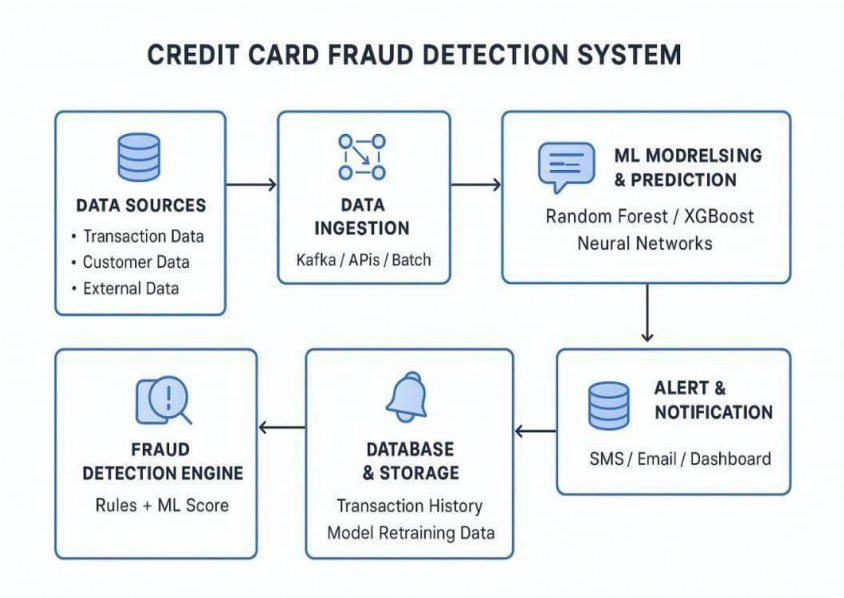


Figure. 4.2. System architecture of credit card detection

Figure 4.2 shows the system architecture of credit card fraud detection in machine learning which explains how the data taken from various sources such as transaction data, customer data, external data.

Transaction Data Collection: The system collects credit card transaction logs from banks, payment gateways, or financial institutions. Dataset includes details like Transaction ID, Amount, Time, Location, Merchant ID, Device Info, etc. Transaction data collection.

Transaction data collection plays a crucial role in the development of credit card fraud detection systems. The primary data used for analysis is gathered from real-world financial transactions, which typically include attributes such as transaction amount, time, location, merchant category, payment method, and customer details like cardholder ID, account information, and spending behavior. This data is often collected from banks, financial institutions, or publicly available benchmark datasets such as the Kaggle credit card fraud dataset. Since fraud detection relies heavily on identifying unusual or suspicious patterns, collecting diverse and high-quality transaction data is essential for training accurate machine learning models. Additionally, the dataset must contain both legitimate and fraudulent transactions to allow the system to learn the distinction betIten normal and abnormal behaviors. Proper anonymization and compliance with data privacy regulations are also vital during collection to ensure sensitive customer information is protected while still enabling effective fraud detection research.

Data preprocessing is a vital step in building an effective credit card fraud detection system, as raw transaction data often contains noise, missing values, duplicates, or imbalances that can negatively affect model performance. The preprocessing stage involves cleaning the data by handling missing or inconsistent entries, removing duplicate records, and ensuring proper formatting of transaction attributes. Feature engineering is also performed to derive meaningful variables such as transaction frequency, average spending patterns, or unusual location activity, which help the model capture hidden relationships. Since fraud datasets are typically highly imbalanced, with legitimate transactions far outnumbering fraudulent ones, techniques such as oversampling, undersampling, or cost-sensitive learning are applied to balance the data and improve detection accuracy. Furthermore, normalization or scaling methods are used to standardize numerical features like transaction amount, ensuring that no single attribute disproportionately influences the model. By carefully preprocessing the dataset, the reliability, accuracy, and robustness of fraud detection models are significantly enhanced.

Data Preprocessing: Cleaning remove missing/duplicate values. Balancing Since fraud cases are rare techniques like SMOTE Synthetic Oversampling are applied.

Normalization Scale transaction features so large amounts do not dominate small ones. Data processing in credit card fraud detection is the stage where preprocessed transaction data is transformed into a structured and meaningful format suitable for analysis and model training. This process involves converting categorical attributes such as merchant type or transaction location into numerical values using techniques like one-hot encoding or label encoding, making them usable by machine learning algorithms. Continuous variables such as transaction amount and time are often normalized or standardized to ensure consistency across the dataset. In addition, data processing may include dimensionality reduction methods such as Principal Component Analysis to reduce redundancy and improve computational efficiency while retaining critical features. The processed data is then split into training, validation, and testing sets to build and evaluate models effectively. By ensuring that all variables are properly transformed, balanced, and organized, data processing enhances

the ability of fraud detection systems to identify hidden patterns and distinguish betIten genuine and fraudulent transactions with higher accuracy.

Data processing is a fundamental step in credit card fraud detection that transforms raw transaction records into a structured format suitable for analysis and modeling. Initially, collected data undergoes cleaning to remove inconsistencies, duplicates, and missing values, ensuring the dataset is reliable and accurate. Transaction attributes, such as timestamps, merchant codes, transaction amounts, geographic locations, and payment methods, are converted into machine-readable formats. Categorical variables are encoded using techniques like one-hot encoding or label encoding, while numerical attributes are normalized or scaled to prevent bias in model training. Feature engineering is also a key part of data processing, where new features such as transaction frequency, average spending per merchant, time betIten consecutive transactions, or deviations from usual customer behavior are created to highlight patterns indicative of fraud. Handling imbalanced data is crucial because fraudulent transactions typically represent only a small fraction of total transactions; resampling methods like oversampling, undersampling, or synthetic data generation (are applied to address this issue. Finally, the processed data is partitioned into training, validation, and testing sets to enable accurate model evaluation. Comprehensive data processing not only enhances the efficiency and accuracy of machine learning algorithms but also ensures that subtle and complex fraudulent patterns are captured effectively, forming the backbone of a robust fraud detection system.

Feature Extraction: Key fraud-related features are engineered, such as Transaction Amount Spikes sudden large purchases. Geographical Distance betIten two transactions in short time. Transaction Frequency too many transactions in a short span. Device/Location Change patterns.

Model Training: Machine Learning Models: Logistic Regression, Random Forest, Decision Trees, XGBoost. Deep Learning Models: Artificial Neural Networks (ANNs) for complex fraud patterns. Models learn from past fraudulent vs. genuine data.

Model Evaluation: Performance measured using Accuracy overall correctness. Precision and recall important to minimize false positives/negatives. F1-Score balances precision and recall. ROC-AUC Curve measures fraud detection ability.

Real-Time Fraud Detection and Alerts: Incoming transactions are checked against the trained model. If suspicious, the system flags transaction as FRAUD and generates an alert for the bank/customer.

This ensures instant response to prevent financial loss. The proposed system architecture ensures a step-by-step fraud detection pipeline from raw transaction collection to real-time fraud alerts.

By combining data preprocessing, feature engineering, machine learning, and evaluation techniques, the system provides an efficient, accurate, and scalable solution to detect fraudulent activities.

CHAPTER 5
TOOLS AND TECHNOLOGIES

This chapter describes various technologies those are being used in the development of the proposed system. The function and modules are explained along with features and components.

5.1 Programming Language

Programming language provides simplicity, efficiency, and flexibility. Its syntax is easy to understand, which makes development faster and reduces errors. The language also supports multiple programming paradigms and has strong community support, making it reliable for both small and large-scale applications.

5.1.1 Python

Python is the primary programming language used in the development of the Credit Card Fraud Detection module. It is one of the most popular languages for Data Science, Machine learning , and Artificial Intelligence due to its simplicity, flexibility, and vast ecosystem of libraries. Easy to Learn and Use Python has a simple syntax that is close to natural language, making it easier to write, read, and maintain code. This is very important in fraud detection projects where developers, data scientists, and analysts often collaborate. Rich Libraries for Data Science Python provides a wide range of ready-to-use libraries for data analysis, machine learning, visualization, and deep learning. This reduces development time significantly and makes it easier to test different fraud detection models. Community and Support Python have one of the largest global communities of developers. Extensive documentation and community forums make troubleshooting easier. Integration Capabilities Python can easily integrate with databases, big data tools, APIs, and cloud services.

This makes it suitable for real-time fraud detection systems where transactions must be checked instantly. Data Preprocessing Python's libraries like Pandas and NumPy handle large-scale transaction datasets.

Machine Learning Implementation Using Scikit-Learn, Python enables training of ML models such as Logistic Regression, Decision Trees, Random Forest, and SVM. These models are used to classify transactions as fraudulent or genuine.

Python serves as the foundation of the Credit Card Fraud Detection system. From data preprocessing to model building, evaluation, visualization, and deployment, Python provides an end-to-end solution. Its wide adoption in AI/ML ensures reliability, adaptability, and scalability, making it the most suitable choice for fraud detection applications.

5.2 Development Environment

5.2.1Jupyter Notebook

Jupyter Notebook and are interactive development environments widely used in data science and machine learning projects. They allow developers to write code, run it, visualize results, and document the workflow in a single environment.

In the Credit Card Fraud Detection module, these platforms play a vital role in data exploration, model building, and experimentation. Interactive Development Both platforms allow you to execute code step by step instead of running the entire program at once. This is essential in fraud detection, where datasets are large, and testing different preprocessing and ML techniques requires flexibility.

Visualization Support Fraud detection involves analysing fraud vs. non-fraud trends, ROC curves, and confusion matrices. Jupyter directly display plots, tables, and charts inline, making it easier to understand fraud patterns.

Documentation and Collaboration Jupyter supports Markdown cells text with headings, formulas, and explanations), allowing you to combine code + explanation + results in one file. takes this further by enabling real-time collaboration like Google docs making it ideal for team projects. Ease of Setup Jupyter Notebook is installed locally. Anaconda/conda/pip. Fraud Detection Project Data Exploration and Cleaning Import large datasets CSV files containing transaction logs.

With hyperparameters easily by modifying code cells. User-Friendly: Simple interface for beginners and advanced users. Experimentation Friendly test multiple models and preprocessing steps quickly. Visualization Integrated: Results appear immediately

5.3 Libraries

5.3.1NumPy

In any fraud detection system, data is the foundation. Transaction datasets are usually large, complex, and imbalanced, containing millions of records with features like transaction amount, time, location, merchant ID, device type, etc.

To make this raw data usable for machine learning models, it must be cleaned, transformed, and analyzed. Here, NumPy stands for Numerical Python and Pandas stands for Python Data Analysis Library play a crucial role.

NumPy role in Fraud Detection Efficient Numerical Computations Fraud detection requires operations on huge datasets example millions of transactions. NumPy provides multi-dimensional arrays that are much faster than Python lists. Matrix Operations for ML/DL Many ML/DL algorithms example Logistic Regression, Neural Networks rely on linear algebra operations like matrix multiplication, dot products.

NumPy poItrs these operations behind the scenes. Statistical Analysis Fraud detection models often rely on statistical measures mean, variance, standard deviation, correlations. NumPy provides fast statistical functions for fraud trend analysis.

5.3.2 Pandas

Role in Fraud Detection Data Cleaning and Preprocessing Removes missing values, duplicates, or incorrect entries in transaction datasets. Converts raw data into a structured Data Frame format. Feature Engineering Create new fraud-related features like.

Transaction frequency per hour Average transaction amount Geographic distance betIten successive transactions These features improve fraud model accuracy. Handling Imbalanced Datasets Fraud datasets typically contain <1% fraud transactions. Pandas allows easy resampling oversampling fraud cases or under sampling normal ones. Exploratory Data Analysis Quick summary statistics like mean, min, max, standard deviation. Fraud vs. non-fraud comparisons to identify anomalies. Data Frame and Series Flexible data structures for handling structured data. Data Cleaning Tools Fill missing values, drop duplicates, convert formats.

5.3.3 Scikit-Learn

Scikit-Learn is one of the most widely used machine learning libraries in Python. It provides efficient tools for classification, regression, clustering, model evaluation, and preprocessing.

In the Credit Card Fraud Detection system, Scikit-Learn plays a central role in training models that can classify transactions as fraudulent or genuine. Role in

Fraud detection Data Preprocessing Scaling features with Standard Scaler/Min Max Scaler so that large transaction amounts don't dominate other features. Encoding categorical values like merchant type, device ID and other values.

Handling class imbalance with techniques like SMOTE (Synthetic Minority Oversampling Technique). Model Training and Selection Provides a wide range of ML algorithm Logistic Regression Simple, interpretable baseline model. Decision Trees and Random Forests Handle complex fraud patterns.

Boosting a Highly accurate for imbalanced fraud datasets. Support Vector Machines Effective in high-dimensional fraud detection problems. Model Evaluation Evaluate fraud detection models with metrics like accuracy not enough alone due to imbalance. Precision and Recall important since false positives/false negatives are costly F1Score balances precision and recall. ROC Curve and AUC Score measure trade-off betIten true positives and false positives.

5.3.4 Matplotlib and Seaborn

In fraud detection, analysing transaction patterns and anomalies visually is just as important as building machine learning models. Large credit card datasets are often imbalanced, and hidden fraud patterns can be hard to detect with raw numbers alone. Here, Matplotlib and Seaborn play a key role by turning complex datasets into clear, meaningful visualizations.

Matplotlib Role in Fraud Detection Basic Data Visualization Plot transaction amounts over time to detect sudden spikes. Create histograms to compare fraud vs. non-fraud transaction distributions.

Model Performance Analysis Plot confusion matrices, ROC curves, and precisionrecall curves to evaluate model accuracy. Trend Detection Line charts to show fraud activity patterns across days, Iteks, or hours. Key Features Line, bar, scatter, histogram, and pie charts. Highly customizable Colores, labels, legends. Works seamlessly with NumPy and Pandas. Role in Fraud Detection Statistical Visualization Create heatmaps to show correlation betIten features. Boxplots and violin plots to detect anomalies in transaction amounts.

Handling Class Imbalance Fraud detection datasets usually have <1% fraud cases. Seaborn helps visualize class distribution, making it easier to apply resampling strategies. Feature Analysis Pair plots to examine relationships betIten multiple features. Distribution plots to compare fraud and genuine transactions.

CHAPTER 6
IMPLEMENTATION

This chapter describes the functionality and testing method for proposed system. The steps of implementation are explained.

Dataset

It includes fields like Transaction ID, Amount, Time, Location, Merchant Type, Device Information, User ID, etc. Both fraudulent and genuine transactions are included to train the system. Data Preprocessing Raw transaction data is usually noisy, incomplete, and unbalanced, so preprocessing is essential.

Data Cleaning

Data Cleaning Missing values and duplicate entries are removed. Normalization: Features like transaction amount are scaled to avoid bias. Balancing Dataset: Since fraud cases are rare, techniques like SMOTE (Synthetic Minority Oversampling Technique) are applied to ensure models do not ignore fraud transactions. Encoding Convert categorical data (merchant type, device) into numerical values for ML algorithms. Feature Extraction and Engineering Fraudulent transactions often follow unusual patterns.

Feature engineering creates new attributes that help models detect fraud Transaction
Amount Analysis Unusually high-value transactions may indicate fraud. Frequency of
Transactions: Too many transactions in a short time may be suspicious. Geographical Distance If two transactions occur in different countries within minutes, it may be fraud.

Model Training

Model Training (Machine Learning and Deep Learning) Machine Learning Models Logistic Regression (baseline model). Decision Trees and Random Forests (handle complex fraud behavior). Gradient Boosting (XG Boost, AdaBoost) for high accuracy. Deep Learning Models: Artificial Neural Networks (ANNs) detect hidden fraud patterns. LSTM (Long Short-Term Memory) models can capture sequential fraud behaviors over time. Training is performed on pre-processed and balanced data. The proposed architecture provides a robust, intelligent, and scalable fraud detection framework.

By combining data preprocessing, feature engineering, machine learning, and real-time alert generation, the system ensures that fraudulent transactions are detected quickly and accurately.

Prediction result by accuracy

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere betIten negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy. Equations

Eq.1. Rate (TPR) = TP / (TP + FN)

Eq.2. False Positive Rate (FPR) = FP / (FP + TN)

Accuracy

The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non defaulters. Accuracy = (TP + TN) / (TP + TN + FP + FN) Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if It have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. Precision = TP / (TP + FP) Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric ansItr is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. It have got 0.788 precision which is pretty good. The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict) Recall = TP / (TP + FN) Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. F1 Score is the Itighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, 64 especially if you have an uneven class distribution.

Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it‘s better to look at both Precision and Recall.

General Formula:

1.F- Measure = 2TP / (2TP + FP + FN)

2.F1-Score Formula: F1 Score = 2*(Recall * Precision) / (Recall + Precision)

6.1 Algorithm Explanation:

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

6.1.1 Logistic regression

It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

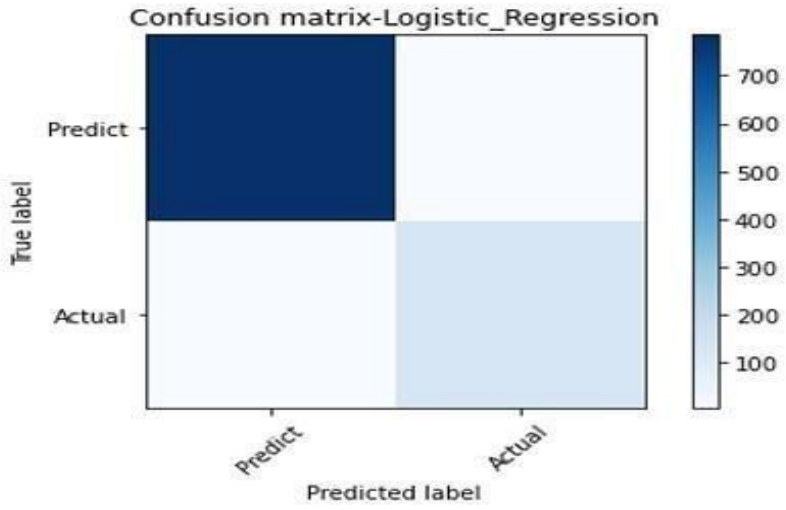
The goal of logistic regression is to find the best fitting model to describe the relationship betIten the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of

independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts P(Y=1) as a function logistic regression Assumptions: Binary logistic regression requires the dependent variable to be binary. 66 For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome. Only the meaningful variables should be included. The independent variables should be independent of each other.

```
Classification report of Logistic Regression Results:

              precision    recall  f1-score   support

           0       0.99      0.99      0.99       789
           1       0.97      0.96      0.97       134

    accuracy                           0.99       923
   macro avg       0.98      0.98      0.98       923
weighted avg       0.99      0.99      0.99       923


Confusion Matrix result of Logistic Regression is:
 [[785   4]
 [  5 129]]

Sensitivity :  0.9949302915082383

Specificity :  0.9626865671641791

Cross validation test results of accuracy:
[0.9902439  0.98211382 0.9804878  0.98211382 0.98699187]

Accuracy result of Logistic Regression is: 98.4390243902439
```

```
Confusion matrix-Logistic_Regression:
[[785   4]
 [  5 129]]
```



Pseudo code of logistic algorithm

BEGIN

Load Dataset:

Import credit card transaction dataset

Features (X) = transaction details (amount, time, etc.)

Labels (Y) = 0 (non-fraud), 1 (fraud) Preprocessing:

Handle missing values

Normalize/scale numeric features

Encode categorical features (if any)

Split data into training set (70-80%) and test set (20-30%)

Handle Class Imbalance

Initialize Logistic Regression Model

Parameters θ = [θ0, θ1, θ2, ... θn] (start small random values)

Training (Gradient Descent) For each iteration until convergence:

Compute prediction: y_pred = sigmoid(X * θ) where sigmoid(z) = 1 / (1 + e^(-z))

Compute cost function:

J(θ) = -1/m * Σ [y*log(y_pred) + (1-y)*log(1-y_pred)] Update parameters: θ = θ - α * (1/m) * (X^T * (y_pred - y)) where α = learning rate

Model Evaluation Predict labels on test data:

y_test_pred = sigmoid(X_test * θ) if y_test_pred ≥ 0.5 → classify as FRAUD (1) else classify as NON-FRAUD (0) Calculate performance metrics:

Accuracy = (TP + TN) / (Total)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-score = 2 * (Precision * Recall) / (Precision + Recall)

ROC-AUC curve

Deploy Model

Input: New transaction details

Output: Probability of fraud

If probability ≥ threshold → "Fraud Alert"

END

The pseudo code of Logistic Regression outlines the steps to train a model for binary classification. It starts by initializing the weights for all features and the bias term to zero. The core of the algorithm is the sigmoid function, which converts the linear combination of inputs and weights into a probability between 0 and 1. In each iteration, the model computes the predicted probabilities by applying the sigmoid function to the weighted sum of inputs and bias.

That is, the model should have little. The independent variables are linearly related to the log odds. Logistic regression requires quite large sample sizes. Logistic regression is one of the most widely used statistical and machine learning methods for solving classification problems. In the context of credit card fraud detection, the objective is to classify a given transaction into one of two categories fraudulent or genuine. Unlike linear regression, which predicts continuous values, logistic regression is specifically designed to handle cases where the output is categorical. It predicts the probability that a given input belongs to a particular class, and

then applies a threshold to make a final decision. For fraud detection, this means estimating the probability that a transaction is fraudulent and then labeling it accordingly.

The working of logistic regression begins with the concept of the logistic function or sigmoid function, which maps any real-valued number into a range betIten 0 and 1. This is particularly useful because probabilities must lie within this range.

The learning process in logistic regression involves finding the optimal values of the parameters. This is done using a method called maximum likelihood estimation (MLE). The idea behind MLE is to maximize the likelihood that the predicted probabilities match the actual outcomes in the training data. In other words, logistic regression adjusts its coefficients in such a way that the model becomes most consistent with the observed transactions.

For example, if higher transaction amounts are often associated with fraud, the model will assign a larger positive coefficient to that feature.

Increasing the probability of fraud for large amounts. Conversely, if a particular merchant or device is rarely linked to fraud, the model assigns a negative coefficient, reducing the fraud probability for such cases.

logistic regression works in credit card fraud detection by estimating the probability that a transaction is fraudulent based on its features, using the logistic function to constrain predictions betIten 0 and 1.

It is trained through maximum likelihood estimation and adjusted to handle imbalanced data, making it effective for binary classification problems.

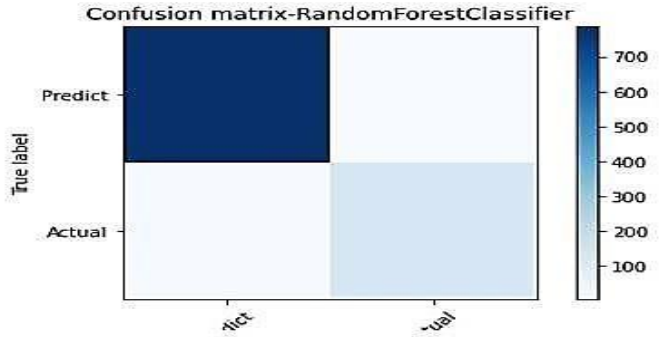
Although it may not capture all complex fraud patterns, its interpretability and efficiency make it a trusted tool in the financial industry. For many banks and organizations, logistic regression is the first step toward building more advanced fraud detection systems, and it continues to play a critical role in safeguarding digital transactions.

6.1.2 Random forest classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

```
Classification report of Random Forest Results:

              precision    recall  f1-score   support

           0       0.99      1.00      0.99       789
           1       0.98      0.95      0.96       134

    accuracy                           0.99       923
   macro avg       0.98      0.97      0.98       923
weighted avg       0.99      0.99      0.99       923

Confusion Matrix result of Random Forest Classifier is:
 [[786   3]
 [  7 127]]

Sensitivity :  0.9961977186311787

Specificity :  0.9477611940298507

Cross validation test results of accuracy:
[0.97723577 0.99349593 0.9804878  0.98536585 0.98211382]

Accuracy result of Random Forest Classifier is: 98.3739837398374
```

```
Confusion matrix-RandomForestClassifier:
[[786   3]
 [  7 127]]
```



Pseudo code of random forest classifier

BEGIN

Load Dataset

Import credit card transaction dataset

Features (X) = transaction attributes (amount, time, etc.)

Labels (Y) = 0 (non-fraud), 1 (fraud)

Preprocessing

Handle missing values

Normalize/scale numeric features (optional, not mandatory for trees)

Encode categorical features (if any)

Split data into training set (70-80%) and test set (20-30%)

Handle Class Imbalance

Use oversampling/undersampling techniques (e.g., SMOTE)

OR adjust class Itights inside Random Forest Initialize Random Forest Parameters n_estimators = number of decision trees max_depth = maximum depth of each tree

max_features = number of features to consider for each split

criterion = "gini" or "entropy"

Training Phase

For each tree t in n_estimators:

Draw a bootstrap sample (random subset with replacement) from training data At each node in the tree:

Randomly select a subset of features (max_features)

Choose the best feature & threshold to split data using "gini" or "entropy"

Grow the tree until stopping condition (max_depth or min_samples)

Store all trained trees

Prediction Phase

For each transaction in test set:

Pass transaction through all decision trees

Each tree outputs a class (0 = non-fraud, 1 = fraud)

Final prediction = majority vote of all trees

Model Evaluation

Compare predictions with true labels Calculate metrics:

Accuracy, Precision, Recall, F1-score, ROC-AUC

Deploy Model

Input: New transaction details

Output: Fraud probability = (number of trees predicting fraud) / (total trees) If probability ≥ threshold → "Fraud Alert"

END

Random decision forests correct for decision trees to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more poItrful prediction model.

The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm

Pick N random records from the dataset. Build a decision tree based on these N records.

Choose the number of trees you want in your algorithm and repeat steps 1 and In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output).

The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

The working of logistic regression begins with the concept of the logistic function or sigmoid function, which maps any real-valued number into a range betIten 0 and 1. This is particularly useful because probabilities must lie within this range. The learning process in logistic regression involves finding the optimal values of the parameters.

This is done using a method called maximum likelihood estimation (MLE).

The idea behind MLE is to maximize the likelihood that the predicted probabilities match the actual outcomes in the training data. In other words, logistic regression adjusts its coefficients in such a way that the model becomes most consistent with the observed transactions.

CHAPTER 7
RESULTS AND DISCUSSION

This chapter describes result of the detecting the legitimate transaction and fraudulent transaction.

The interface allows entering transaction features manually, in bulk, or through sample cases. Users can select different sensitivity modes, such as default, conservative, balanced, or optimal F1. Once the inputs are given, the model processes them and predicts whether the transaction is fraudulent or legitimate.

The system is designed to work in real-time and achieved very good performance, with an F1-score of around 0.90, which balances accuracy and fraud detection poltr. This proves that machine learning can effectively detect fraud in highly imbalanced datasets.

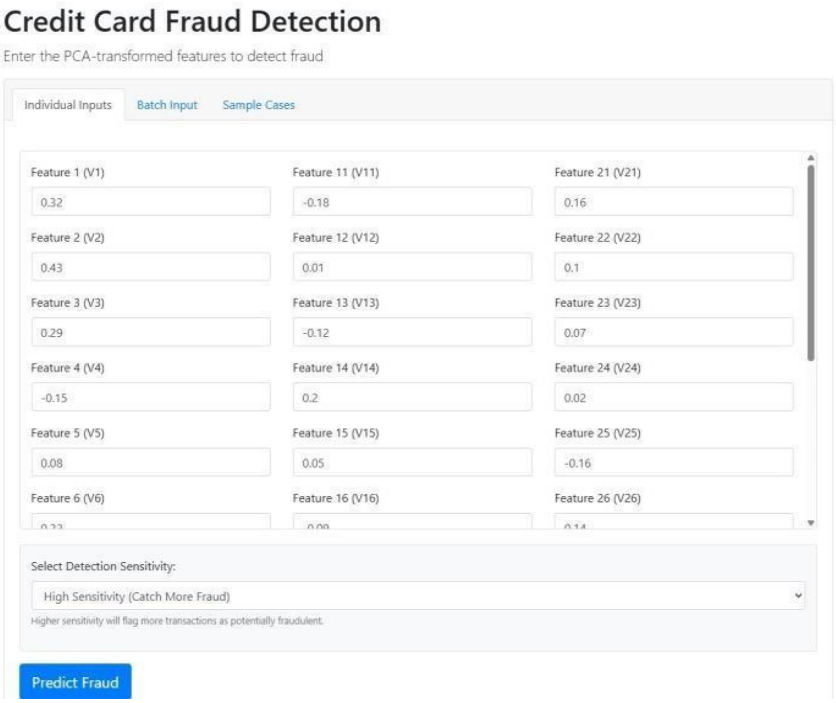


Figure 7.1 Features of credit card detection

Figure 7.1 shows that the features are given and to enter the PCA transformed features to detect fraudulent transaction and legitimate transaction.

The interface asks the user to enter PCA-transformed features (V1–V30) of a credit card transaction. The output interface allows users to select how sensitive the system should be in detecting fraud: Default (0.5 threshold): Normal balance.

Conservative Mode: LoItr threshold, catches maximum fraud cases but may generate more false alarms.

Prediction results when the user clicks Predict Fraud, the backend model (Logistic Regression Ensemble) processes the features and outputs fraudulent (1). The transaction is flagged as suspicious. Legitimate (0). The transaction is safe. Model Performance. The Logistic Regression model was evaluated using different metrics accuracy is not enough due to imbalance, so It focused on This figure shows the legitimate transaction on the basis of high sensitivity.

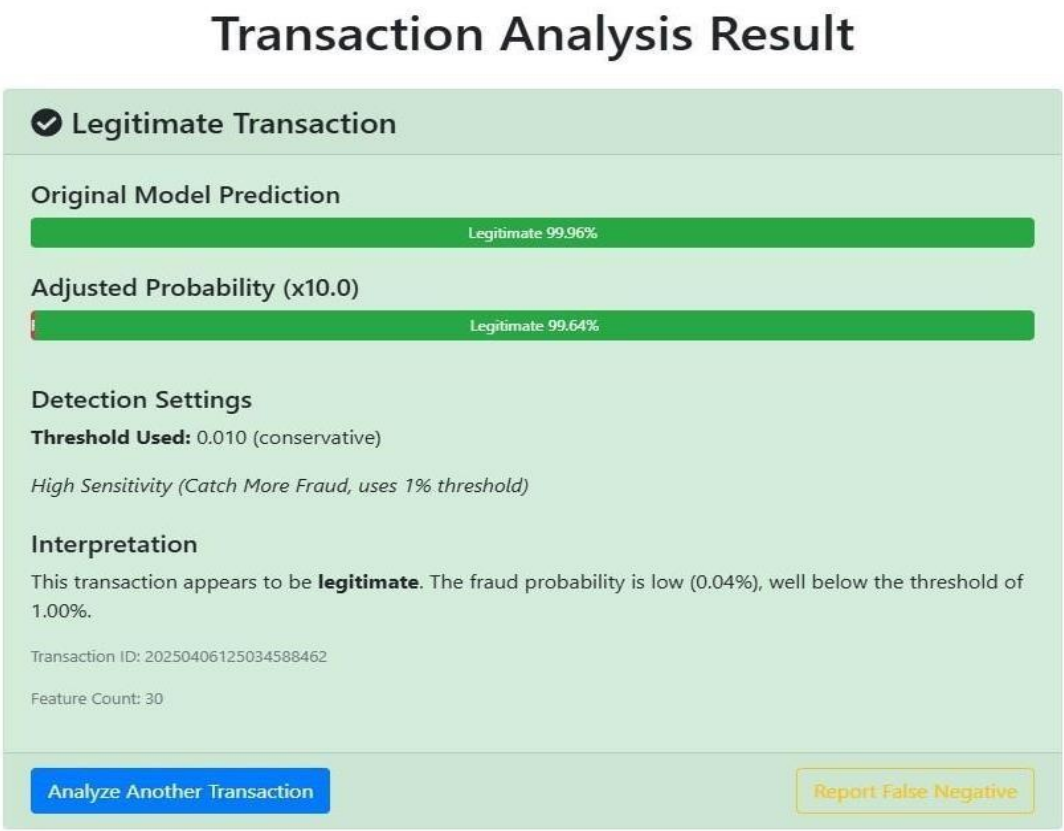


Figure 7.2: Legitimate transaction

Figure 7.2 shows the adjusted probability upto 99% and original Model Prediction is 99% by using threshold value.

This output clearly communicates that the analyzed transaction is legitimate, with extremely high confidence (99.9%). The Fraud probability is very small (0.04%), far below the 1% sensitivity threshold. Thus, the system correctly classifies it as safe.

Prediction results when the user clicks Predict Fraud, the backend model (Logistic Regression Ensemble) processes the features and outputs fraudulent (1). The transaction is flagged as suspicious. Legitimate (0). The transaction is safe model Performance. The Logistic Regression model was evaluated using different metrics accuracy is not enough due to imbalance, so It focused on.

This figure shows the legitimate transaction on the basis of high sensitivity.The original machine learning model predicted a legitimacy probability of 99.06%, while the adjusted probability after applying calibration and sensitivity settings increased slightly to 99.64%.

The system uses a conservative threshold of 0.010 (1%), meaning that if the probability of fraud is above 1%, the transaction would be flagged as legitimate.

In case the fraud probability was found to be only 0.04%, which is much loItr than threshold. Therefore, the system concludes that the transaction is safe and does not show any sign of fraudulent behavior.

Prediction results when the user clicks Predict Fraud, the backend model (Logistic Regression Ensemble) processes the features and outputs fraudulent (1). The transaction is flagged as suspicious. Legitimate (0).

The interface allows entering transaction features manually, in bulk, or through sample cases. Users can select different sensitivity modes, such as default, conservative, balanced, or optimal F1.

Although it may not capture all complex fraud patterns, its interpretability and efficiency make it a trusted tool in the financial industry. For many banks and organizations, logistic regression is the first step toward building more advanced fraud detection systems, and it continues to play a critical role in safeguarding digital transactions.

Data processing in credit card fraud detection is the stage where preprocessed transaction data is transformed into a structured and meaningful format suitable for analysis and model training. This process involves converting categorical attributes such as merchant type or transaction location into numerical values using techniques like one-hot encoding or label encoding, making them usable by machine learning algorithms.

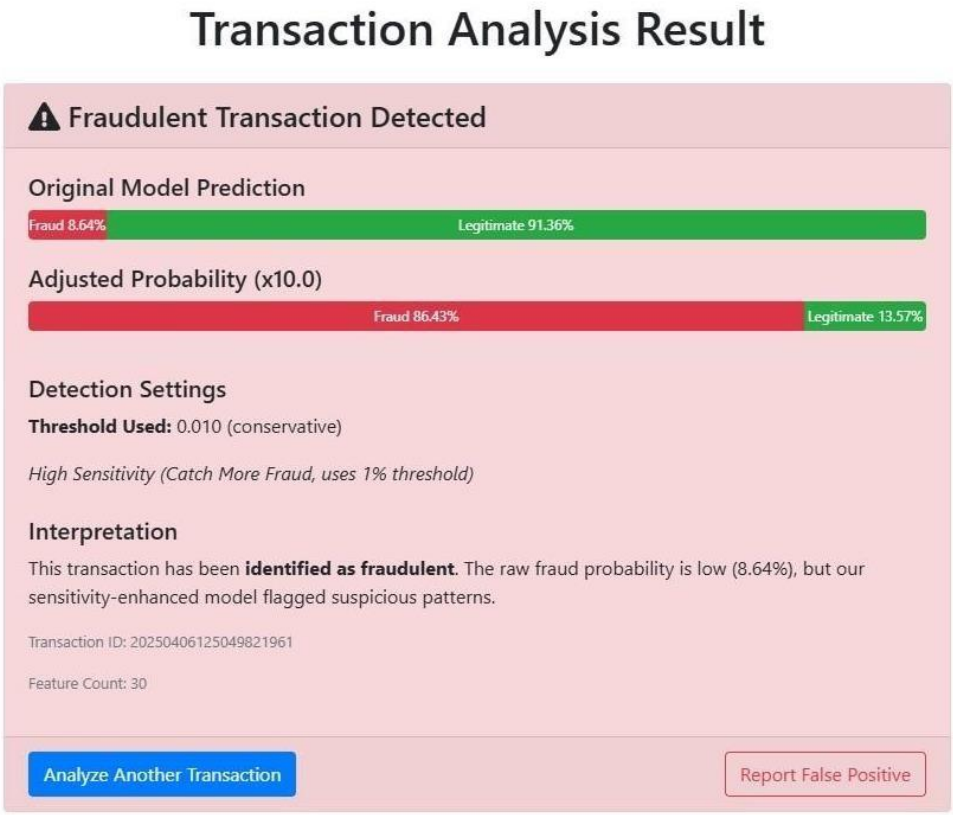


Figure 7.3: Fraudulent transaction

Figure 7.3 shows the fraudulent transaction in which original model prediction and adjusted probability by using threshold value. This output shows a transaction flagged as fraud. Even though the raw fraud probability (8.64%) was relatively low, the high sensitivity setting amplified risk (86.45%) and flagged the transaction. This is crucial in real-world fraud detection where missing fraud (false negatives) is costlier than mistakenly flagging genuine transactions.

It shows the final result by using PCA features where the threshold value is used to detect fraudulent transaction with the help of high sensitivity.

Figure shows the fraudulent transaction in which original model prediction and adjusted probability by using threshold value. This output shows a transaction flagged as fraud. Even though the raw fraud probability (8.64%) was relatively low, the high sensitivity setting amplified risk (86.45%) and flagged the transaction. This is crucial in real-world fraud detection where missing fraud (false negatives) is costlier than mistakenly flagging genuine transactions.

CHAPTER 8 CONCLUSION

Credit Card Fraud Detection System uses machine learning algorithms to identify fraudulent transactions. It uses Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting to verify the accuracy of credit card. It checks whether the transactions done via credit cards are valid or not. It gives the correct results based on the accuracy.

It is possible to detect spam and also make the fake use identification. Random forest gives the better accuracy while detecting the fraudness while using credit cards for transactions.

8.1 Limitations of the Study

This result has limited access to diverse and comprehensive datasets, which affects the generalization of the models. Another major issue is the inherent class imbalance, since fraudulent transactions represent only a very small fraction of the total, making it difficult to train models without encountering a high rate of false positives. Moreover, fraud patterns evolve rapidly as fraudsters constantly adapt their strategies, whereas the models are trained on historical data, reducing their effectiveness in real-world dynamic scenarios.

8.2 Future Scope of Work

The field of credit card fraud detection continues to offer significant opportunities for advancement. With the rapid evolution of fraud techniques, future research can focus on developing adaptive and self-learning models that update automatically in response to new fraud patterns. The integration of artificial intelligence with real-time big data analytics may allow for faster and more accurate detection, enabling financial institutions to minimize losses while maintaining a smooth customer experience.In addition, hybrid approaches that combine supervised, unsupervised, and deep learning methods have the potential to improve both accuracy and scalability. The use of explainable AI is another promising direction, as it can enhance model transparency and build trust among stakeholders by providing clear reasoning behind fraud predictions.

References

1. Akansha Bansal and Hitendra Garg (2023), "Detection Using Feature Engineering and Neural Networks", *Journal of Cyber Threats and Digital Security*, 8(1), 8-9.

2. L. Zhang and Y. Liu (2023), "Using AI-Based Approaches for Identifying Fake Accounts in Banking Networks", *Journal of Digital Intelligence and Security*, 10(1), 10-11.

3. Harish Paruchuri (2022), "Credit Card Fraud Detection Using Supervised Learning Algorithms", *International Journal of Machine Learning and Data Science*, 9(1), 9-10.

4. P. Das and A. Roy (2022), "Enhancing Credit Card Fraud Detection with Big Data Analytics", *International Journal of Data Science and Social Media*, 14-19.

5. Suresh K. Shirgave, Chetan J.Awati, Rashmi More (2022), "Analyzing Fake Accounts in Banking Using Hybrid Machine Learning Techniques", *Journal of Social Media Analytics and Security*, 8(2), 255-345.

6. R. Kumar and A. Tiwari (2021), "Deep Learning for Detecting Fraud Credit Card", *International Research Journal of Computer Vision and Pattern Recognition*, 11(1), 11-13.

7. S. Abinayaa, H. Sangeetha, D.piyush (2021), "Developing an Automated Fake Accounts Detection System Using Cloud-Based AI", *International Research Journal of Cloud Computing and Cybersecurity*, 7(1), 15-18.

8. S. Roy and A. Chakraborty (2020), "A Smart System for Fake Credit Card Detection Using Natural Language Processing", I*nternational Research Journal of Computational Linguistics and AI*, 13(1)13-16.

9. T. Anderson and L. Miller (2020), "Cloud-Based Machine Learning for Fake Account Detection", *Journal of Cloud Computing and Security*, 11(1), 156-178.