AI-Powered Banking in Revolutionizing Fraud Detection: Enhancing Machine Learning to Secure Financial Transaction

Prem Kumar Sholapurapu

Research Associate and Senior Consultant, CGI Katy, Texas, USA

KEYWORDS

Artificial Intelligence, Machine Learning, Fraud Detection, Banking Sector, Cybersecurity, Adaptive Neuro Boosted Forest, Hash Blue Hellman Algorithm

ABSTRACT

With the constant evolution of financial fraud techniques, the demand for advanced technological solutions to secure banking transactions and protect sensitive information is at an all-time high. For this reason, Artificial Intelligence (AI) and Machine Learning (ML) have been adopted as game-changing agents in improving fraud detection and prevention. These technologies allow for real-time anomaly detection, pattern identification, and predictive analytics, revolutionizing the conventional approach to fraud detection. This article explores how AI-enabled fraud detection platforms are transforming the banking industry, specifically in the areas of cyber security, real-time transaction review, and risk management in high-frequency trading. We present a suite of new techniques to enhance the precision and proficiency of fraud detection models. Specifically, it includes the Splaso Quash Filter, a data preprocessing approach designed to optimize raw data for machine learning models, and Ripe Horn Twin Fish Optimization, a lucid feature extraction technique that improves the ability of the model to detect the critical variables affecting fraud. It uses the Adaptive Neuro Boosted Forest (ANBF) algorithm which is a combination of the neural network's adaptability and the robustness of the use of error information from the decision forest to improve the decision accuracy more than the algorithm band in the decision. We also delve into the Hash Blue Hellman Algorithm, a form of cryptography that secures the storage of data and offers a protective approach for sensitive data transactions. We discuss the implications of using AI for fraudulent activity detection, including regulatory issues, potential negative consequences, and future trends in banking fraud prevention technologies. Experimentation was conducted on the Bank Fraud Detection Dataset under a Python environment. From the analysis, it was revealed that the suggested methodology offers secure financial transactions and maintains trust in the banking industry.

I. INTRODUCTION

Financial fraud comes with increasing sophistication which mandates the use of advanced technology solutions by banks and financial institutions to secure transactions and customer data. Conventional fraud detection approaches such as rule-based systems and manual monitoring are often insufficient in adapting to the ever-evolving nature of fraudulent behavior. Due to increasingly sophisticated attack techniques including but not limited to identity theft, account takeovers, and transaction laundering, financial institutions need more robust, adaptive, and intelligent fraud detection systems. Faced with this dynamic reality, Fraud Detection and Prevention (FDP) has increasingly turned to Artificial Intelligence (AI) and Machine Learning (ML) as indispensable weapons in the fight against fraud (Bolton & Hand, 2002).

Artificial Intelligence and Machine Learning: With access to vast amounts of historical transaction data, AI and ML algorithms can detect patterns and anomalies, enabling banks to identify potential fraud more rapidly and accurately.

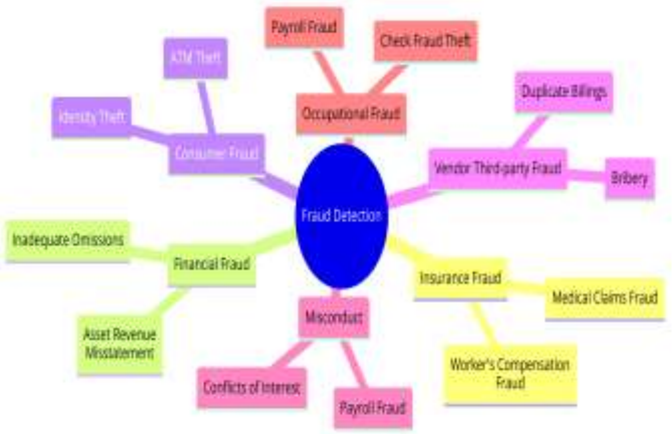


Figure 1: Fraud Detection Categories

Financial fraud comes with increasing sophistication which mandates the use of advanced technology solutions by banks and financial institutions to secure transactions and customer data. Conventional fraud detection approaches such as rule-based systems and manual monitoring are often insufficient in adapting to the ever-evolving nature of fraudulent behavior. Due to increasingly sophisticated attack techniques including but not limited to identity theft, account takeovers, and transaction laundering, financial institutions need more robust, adaptive, and intelligent fraud detection systems. Faced with this dynamic reality, Fraud Detection and Prevention (FDP) has increasingly turned to Artificial Intelligence (AI) and Machine Learning (ML) as indispensable weapons in the fight against fraud (Bolton & Hand, 2002). Artificial Intelligence and Machine Learning: With access to vast amounts of historical transaction data, AI and ML algorithms can detect patterns and anomalies, enabling banks to identify potential fraud more rapidly and accurately.

This mind map shows different types of fraud: consumer fraud, financial fraud, occupational fraud, misconduct fraud, vendor fraud, insurance fraud and more specifically: identity theft, payroll fraud, check fraud, bribery, medical claims fraud, etc. It must provide a framework of types of fraud for better detection and prevention.

Data-Driven Analysis — AI-powered fraud detection systems rely on data-driven techniques to detect deviations in transactional and customer behavior. ML models can identify behaviour associated with fraud through the analysis of large amounts of historical and real-time associated data, which might be undetectable in traditional rule-based models (West & Bhattacharya, 2016). Supervised and unsupervised learning models, both types of machine learning algorithms, use advanced statistical methods to determine whether a transaction is fraudulent or legitimate based on historical data. For example, supervised learning models train predictive models using labeled datasets, while unsupervised learning models detect anomalies without prior knowledge of fraudulent activities. With the application of deep learning methods (e.g., convolutional neural networks (CNNs) and recurrent neural networks (RNNs)), they have been significantly enhancing the accuracy of fraud detection (Le et al., 2018).

This study discusses the role of AI-driven fraud detection systems in the banking industry, with specific attention to their implementations in cybersecurity, real-time transaction monitoring, and risk reduction in high-frequency trading. The article presents innovative approaches to enhance the accuracy and efficiency of fraud detection, such as the Splaso Quash Filter, Ripe Horn Twin Fish Optimization, and the Adaptive Neuro Boosted Forest (ANBF) algorithm. The Splaso Quash Filter — a data preprocessing tool that provides access to machine learning models by eliminating noise and retaining relevant features from raw data before its use. On the other hand, Ripe Horn Twin Fish Optimization acts as an enhancement to the feature extraction process, allowing it to identify prominent variables, thus sex typing the model's performance in fraud detection. ANBF – Adaptive Neuro Boosted Forest] combines neural network adaptability with the robustness of decision forests, resulting in a significant increase in predictive accuracy in fraud detection models. It also presents an overview of the Hash Blue Hellman Algorithm, a complex algorithm used to protect sensitive transaction data, emphasizing its importance in providing an additional line of defense against cyber threats.

This paper not only describes the technical advancements in AI-driven fraud detection but also evaluates the larger impacts of these technologies in the banking sector. This comprises the regulatory aspects, the ethical quandaries that arise from employing AI, and the threats from adversarial attacks that distort machine learning algorithms. Fraud techniques are evolving at an incredible pace, and financial service organizations must continuously refresh their fraud detection models while we must integrate AI-led solutions with legacy cybersecurity systems. These systems become more effective with the examples provided in the real-world datasets, for example, the Bank Fraud Detection Dataset allows evaluation of the AI models under realistic banking environments.

This study comes with an innovative approach by persuading how artificial intelligence (AI), as well as machine learning (ML), can change the landscape of fraud detection in financial transactions for all banks to proactively manage risk. Artificial intelligence has become an invaluable asset in the arena of banking security due to its capability to process huge assets of transactional records in real time, identify irregularities, and forecast potential fraud trends. As AI-powered fraud detection continues to advance, it opens doors for financial transaction security and trust retention for banks. With the adoption of these sophisticated methodologies, banks are better equipped to withstand the evolution of their creator's organized and complex schemes to act against them to ensure a secure financial environment for both customers and institutions.

.

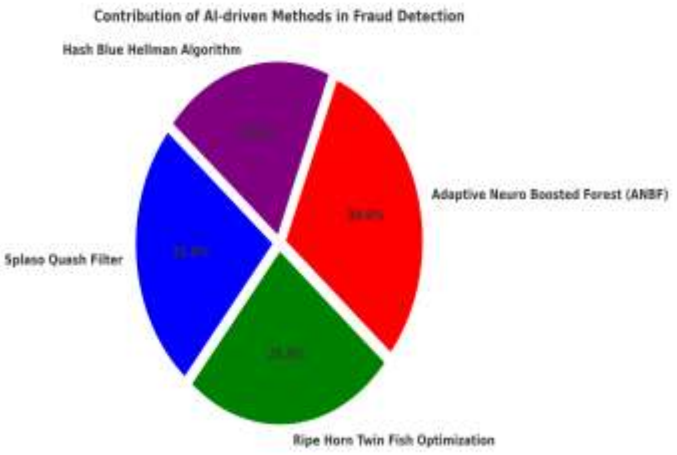


Figure 2: Contribution of AI-Driven Methods in Fraud Detection

Here is a pie chart of contributions by AI methods: Adaptive Neuro Boosted Forest (30%), Ripe Horn Twin Fish Optimization (25%), Splaso Quash Filter (25%), Hash Blue Hellman Algorithm (20%). We also observed that ANBF dominates in the classification of fraud and preprocessing and feature extraction are also important.

The structure of this paper is as follows: Section 1 gives an introduction to the theory of fraud detection challenges and AI-driven methods. Related Work: Section 2 presents related work in fraud detection, outlining existing approaches, particularly those based on machine learning, and cryptographic security. Section 3 describes the research methodology. Section 4 describes experimental results regarding model performance, security levels, and the contribution of different AI techniques. Section 5 summarizes this study and proposes future improvements, which include the detection of fraud in real-time, further enhancements using deep learning, and wide analysis of fraud via multiple sources.

II. Related Works

Anshika Gupta et al. [1] developed an ML system that predicts the approval status of a loan application based on the applicant's income and credit history; this is only one of many research that has focused on machine learning models for fraud risk prediction. Golak Bihari Rath et al. [2] state that machine learning and classification algorithms help improve loan clearance procedures. With the help of many ML models, Mayank Anand et al. [3] can forecast how safe banks will handle loan defaults. Syed Zamil HasanShoumo et al. [4] developed a machine-learning model for assessing credit risk and predicting loan defaults in the banking sector. The objective of Mohammad Abdullah et al. [5], who utilise machine learning techniques, is to anticipate nonperforming loans in financial institutions located in developing nations. Out of all the models tested, the random forest model outperformed them all with an accuracy of 76.10 percent. Machine learning loan approval forecasting was the focus of Bhargavet et al. [6], who contrasted decision tree and random forest methods. Based on the facts, Random Forest outperforms competing algorithms. Dansana et al. [7] utilise a series of loan approval parameters—gender, level of education, occupation, business type, length of the loan, and marital status—to predict defaults. The article uses the Random Forest method for its prediction analysis. Blessie et al. [8] use a variety of machine learning techniques, including logistic regression, decision trees, support vector machines (SVMs), and Naive Bayes classification, to forecast loan penalties using a loan dataset. Naive Bayes was the most accurate model for loan prediction, reaching 80.42 percent. Zhu et al. [9] used a loan dataset and a number of machine learning techniques, including logistic regression, decision trees, support vector machines (SVMs), and Naive Bayes, to forecast loan sanctions. The study found that Naive Bayes was the most effective model for loan forecasting, with an accuracy of 80.42%. Alsaleem et al. [10] tested multiple ML systems for grading the likelihood of default on bank loans using a dataset consisting of 1,000 loan applications. Neural networks, Bayesian networks, decision trees, and random forests were all part of this set of algorithms. Based on the outcomes, it is clear that the Multilayer Perceptron neural network outperformed its competitors. With an impressive 80% accuracy rate, it clearly aids banks in making data-driven decisions on loan approvals. Di Wang et al. [11] created NeuCredit, a model for evaluating and projecting credit risk in online commerce. By breaking the probability of default into three factors—willingness to repay, ability to repay, and behavioural risk—the model generates forecasts that are straightforward to understand, which is an advance above prior methods. Uddin et al. [12] offer an ensemble ML method to predict a bank's loan approval decision using the Kaggle dataset. Using preprocessing processes and introducing two ensembles, the study examined the top three models, reaching a peak accuracy of 87.26%. The research goes a step further by demonstrating its potential practical applications through a user interface (UI) desktop program. Several researchers have attempted to integrate privacy methods into machine learning, like Zhigang Lu et al. [13] who developed a novel different private framework for deep learning; these methods include differential privacy and homomorphic encryption. Step one is to train a public neural network using a convex loss function. Step two is to randomly choose a neuron from the output layer and inject it with DP noise. Both DP and GANs were covered in Ma et al.'s [14] deep learning overview. Cristiano et al. [15] presented D-ZOA, a privacy-preserving distributed technique that minimises a regularised empirical risk function via zeroth-order optimisation. By guaranteeing $(\downarrow,\partial)\rightarrow$DP, D-ZOA outperforms the accuracy of existing differentially-private methods. A group of academics worked on the development of Federated Learning for PPCRA: A privacy-preserving approach is presented by Jean-François et al. [16] for probabilistic voltage forecasting in local energy communities. Utilising DP methods and federated learning, the method is implemented. Abdullah Lakhan et al. [17] propose a hybrid approach to healthcare job scheduling that uses blockchain technology and federated learning. The suggested method aims to address energy and delay restrictions in healthcare applications while also protecting user privacy and detecting fraud. A hierarchical IoT network fabric was built using microchains and a hybrid consensus mechanism in the proposal by µDFL, Xu et al. [18]. Efficient, private, scalable, and secure decentralised federated learning over Internet of Things (IoT) networks is the goal of this approach. Other Methods for Maintaining Secrecy: Wang et al. [19] present a distributed ML system that protects users' privacy via the use of ADMM perturbation and local randomisation. This approach is designed to protect sensitive data while providing users with varying levels of privacy. To protect a machine learning model's decision boundary, HuadiZheng et al. [20] proposed a strategy they called BDP, which involves masking expected responses with noise. Deep learning raises concerns about privacy and security, which Liu et al. [21] attempt to solve. Huang et al. [22] proposed a technique of trainable picture encryption to protect privacy in deep learning applications, particularly for medical photographs. Additionally, they have refined prior encryption methods, drawing attention to changes in the keyspace of images encrypted with their suggested approach. Supported by HE, NNP for PPCRA was developed by a large group of researchers: Bernardo Pulido-Gaytan et Chandramohan et al. (2013) [23] examined cloud data secrecy and proposed an evolutionary model for privacy protection, shedding light on the demand for creative techniques to increase security. In their presentation of a model for chaotic picture encryption incorporating state-of-the-art technologies, Paroda et al. (2023) [24] offered outstanding protection against various attacks. Movassagh et al. (2020) [25] developed a strategy for training neural networks that improves accuracy through the use of optimisation algorithms. Using a deep learning model, Alzubi et al. (2021) [25] showed that medical data may be securely sent via homomorphic encryption. In order to improve safety and user happiness in IoT-dependent smart cities, Gheisari et al. (2021) [26] created the OBPP design. In their publication on homomorphic encryption for credit risk analysis, Pulido-Gaytan et al. [27] provided a comprehensive review of combining HE with neural networks. In contrast to Stephanie et al. [28], who included a plethora of privacy approaches, AONO et al. [29] offered the PPLR strategy to secure data sharing. Chaudhuri et al. [30] constructed PPLR utilising differentiative privacy techniques to enhance accuracy. To find a middle ground between privacy and functionality, Zheng et al. [31] introduced a method called PCAL. Combining HE and MPC allowed Han et al. [32] to conduct safe logistic regression. A cloud-based credit risk analysis system utilising homomorphic encryption was created by Divakar et al. [33] to ensure data anonymity.

III. PROPOSED WORK

Robust fraud detection algorithms that can efficiently identify fraudulent transactions while minimising false positives are necessary due to the growing sophistication and complexity of financial fraud. Due to their dependence on static thresholds and established heuristics, traditional methods of fraud detection, like rule-based systems and statistical models, frequently miss patterns of dynamic fraud. In response to these shortcomings, this research presents a state-of-the-art AI-driven fraud detection system that makes use of a wide range of cryptography and machine learning algorithms to improve efficiency, accuracy, and security.

Data preprocessing (Splaso Quash Filter), feature extraction (Ripe Horn Twin Fish Optimisation), fraud classification (Adaptive Neuro Boosted Forest - ANBF), and safe data storage (Hash Blue Hellman Algorithm) are all presented in this part as proposed approaches. The precision of fraud detection, the effectiveness of calculation, and the safety of the system are all significantly enhanced by each of these methods. These approaches are described in depth in the following subsections, with mathematical formulas to back them up.

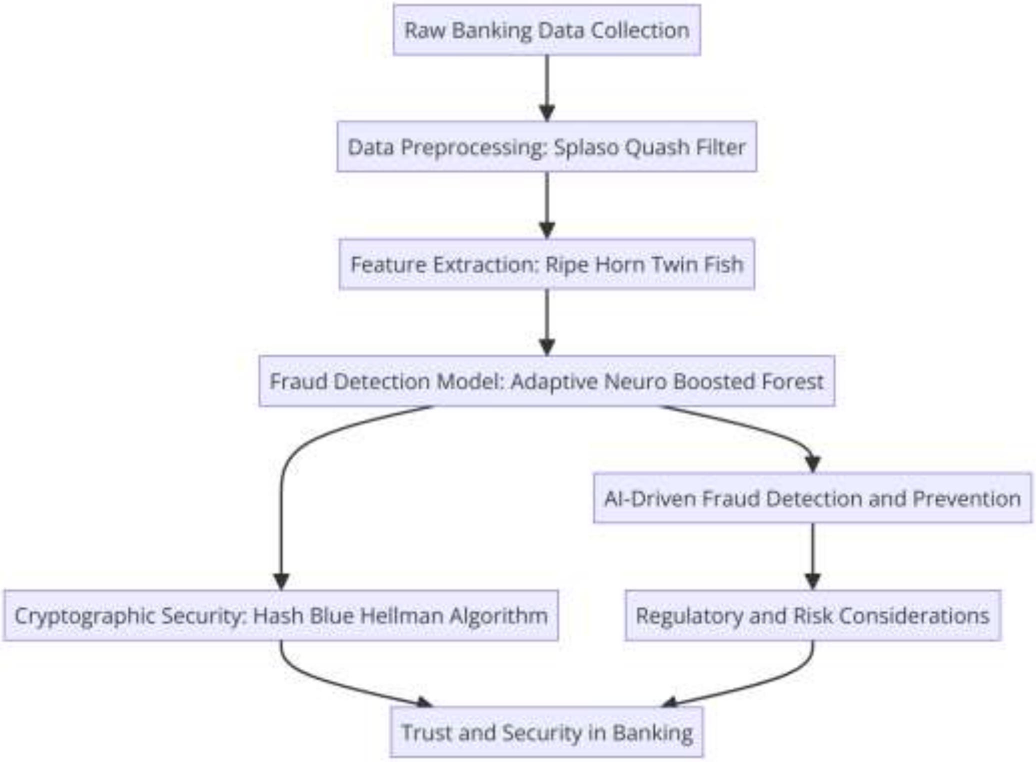

Raw Banking Data Collection
Data Preprocessing: Splaso Quash Filter
Feature Extraction: Ripe Horn Twin Fish
Fraud Detection Model: Adaptive Neuro Boosted Forest
AI-Driven Fraud Detection and Prevention
Cryptographic Security: Hash Blue Hellman Algorithm
Regulatory and Risk Considerations
Trust and Security in Banking

Figure 3: Fraud Detection Workflow Using AI Techniques

This flowchart outlines the AI-driven fraud detection processand banking security. It highlights a structured, secure fraud detection framework.

A.Data source

The IEEE-CIS Fraud Detection Dataset on Kaggle, which is a large-scale dataset for machine learning-based fraud detection of financial transactions. This data is desensitized transactional and identity information, encompassing transaction sum, card information, address, email domain, and digital payment characteristics. There is a lot of imbalance in the data, and frequent transactions are a small number.

B.Data Preprocessing using Splaso Quash Filter

The raw transactional data frequently includes missing values, noise, and redundant features that can hinder the effectiveness of these fraud detection models. We propose the Splaso Quash Filter, a new preprocessing technique specifically engineered to elevate data quality prior to ingestion by machine learning models. Thus the main parts of this pre-processing step are: This ensures only good quality data is used to train the AI models with missing data handling feature scaling noise reduction outlier detection

Missing information can lead to bias and impair fraud detection results, necessitating an appropriate imputation approach. We implemented an iterative imputation approach, where each missing value is estimated based on its relationship with observed transaction features. Imputation of Values The formula for the imputation update is:

$$X_{new} = X_{orig} + \lambda \cdot \nabla f(X) \tag{1}$$

Where X_{new} is the imputed dataset, X_{orig} is the original dataset, λ is the learning rate controlling imputation, and $\nabla f(X)$ represents the gradient-based approximation of missing values, calculated as:

$$\nabla f(X) = \frac{\sum_{i=1}^{n} w_i \cdot (X_i - X_{mean})}{\sum_{i=1}^{n} w_i} \tag{2}$$

wherew_i is the weight assigned to feature i, X_i represents the transaction feature, and X_{mean} is the mean of the observed data. The imputation error is evaluated using:

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n} \left|X_{new,i} - X_{orig,i}\right| \tag{3}$$

Where ϵ is the imputation error threshold. The process terminates when ϵ is below a predefined limit.

Feature scaling is performed to normalize transaction attributes so that all features contribute equally to fraud detection. Min-max normalization is applied using the transformation function:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4}$$

Where X_{norm} is the normalized transaction value, X_{min} and X_{max} are the minimum and maximum values of the transaction feature. An alternative method, Z-score standardization, is applied to features following a Gaussian distribution:

$$X_{std} = \frac{X - \mu}{\sigma} \tag{5}$$

whereX_{std} is the standardized feature value, μ is the feature mean, and σ is the standard deviation.

To mitigate data noise, a Gaussian noise filter is applied, where the smoothed transaction value is computed as:

$$X_{filtered} = X_{norm} - \epsilon \cdot N(0, \sigma^2) \tag{6}$$

Where ϵ is the noise reduction factor and $N(0, \sigma^2)$ is a Gaussian noise distribution with mean zero and variance σ^2. The variance is dynamically computed using:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} \left(X_{norm,i} - X_{mean}\right)^2 \tag{7}$$

To further smooth transaction data while preserving fraud patterns, an adaptive moving average filter is applied:

$$X_{\text{smooth },t} = \alpha X_{\text{filtered },t} + (1-\alpha) X_{\text{smooth },t-1} \tag{8}$$

Where α is the smoothing factor (typically between 0.1 and 0.5), $X_{\text{filtered},t}$ is the noise-reduced transaction at time t, and $X_{\text{smooth},t-1}$ is the previous smoothed transaction value. The smoothing effectiveness is evaluated by minimizing the mean squared error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \left(X_{\text{filtered },i} - X_{\text{smooth },i}\right)^2 \tag{9}$$

Outliers in transactional data, often representing fraudulent activities, are detected using the Interquartile Range (IQR) method:

$$\mathbf{IQR} = \mathbf{Q_3} - \mathbf{Q_1} \tag{10}$$

Where $\mathbf{Q_3}$ and $\mathbf{Q_1}$ represent the third and first quartiles of the transaction data, respectively. A transaction is classified as an outlier if it satisfies the condition:

$$\mathbf{X} > \mathbf{Q_3} + \mathbf{1.5} \cdot \mathbf{IQR} \text{ or } \mathbf{X} < \mathbf{Q_1} - \mathbf{1.5} \cdot \mathbf{IQR} \tag{11}$$

For further robustness, an exponential weighting function is used to adjust outlier influence:

$$\mathbf{X_{adj}} = \mathbf{X} \cdot \mathbf{e}^{-\boldsymbol{\beta} \cdot |\mathbf{X} - \mathbf{X_{mean}}|} \tag{12}$$

Where $\boldsymbol{\beta}$ is the outlier suppression factor. This ensures that extreme values have minimal impact on fraud detection models.

The final Splaso Quash Filtered dataset is obtained by aggregating all preprocessing transformations:

$$\mathbf{X}_{\text{final}} = \mathbf{X}_{\text{adj}} \cdot \mathbf{X}_{\text{smooth}} \tag{13}$$

The equation ensures that the resulting dataset is free of missing values, noise, and outliers, and is well-normalized for fraud detection.

The Splaso Quash Filter significantly improves fraud detection accuracy by enhancing data consistency, reducing variance, and ensuring that fraud models operate efficiently. The next section explores feature selection techniques to further optimize fraud detection models.

C.Feature Extraction using Ripe Horn Twin Fish Optimization

Feature selection is an essential process in fraud detection because irrelevant or redundant features can lead to poor model performance due to increased computation complexity and the loss of model generalization. In this research, we present the Ripe Horn Twin Fish Optimization (RHTFO) algorithm, a nature-inspired optimization method that identifies the more important fraud detection features dynamically. In contrast to other feature selection techniques such as exhaustive search techniques or ranking methods, which select features based on a simple ranking function, our method, RHTFO follows a very different methodology where a candidate feature subset keeps evolving selection for the best combination of features matching fraud classification accuracy and information redundancy should be minimized.

RHTFO is used to optimize the number of features selected as well as provide useful guidance to detect fraud optimally where both the number of selected features to be used and the accuracy of fraud detection are the optimization objective. The objective function is given by:

$$\mathbf{J}(\mathbf{F}) = \boldsymbol{\alpha} \cdot \mathbf{Acc}(\mathbf{F}) - \boldsymbol{\beta} \cdot \frac{|\mathbf{F}|}{|\mathbf{F}_{\text{total}}|} \tag{14}$$

where $\mathbf{J}(\mathbf{F})$ represents the objective function, $\mathbf{Acc}(\mathbf{F})$ is the fraud detection accuracy using feature subset $\mathbf{F}, |\mathbf{F}|$ is the number of selected features, $|\mathbf{F}_{\text{total}}|$ is the total number of available features, and $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are weighting parameters controlling the trade-off between accuracy and feature reduction. The gradient-based adjustment for optimizing feature selection is given by:

$$\nabla \mathbf{J}(\mathbf{F}) = \boldsymbol{\alpha} \cdot \frac{\partial \mathbf{Acc}(\mathbf{F})}{\partial \mathbf{F}} - \boldsymbol{\beta} \cdot \frac{\mathbf{1}}{|\mathbf{F}_{\text{total}}|} \tag{15}$$

where the first term adjusts features to improve fraud detection accuracy, and the second term penalizes excessive feature selection.

The evolutionary mechanism of RHTFO is inspired by adaptive fish movement in dynamic environments, where candidate feature sets adjust based on prior performance. The core movement equation of the Twin Fish Strategy in RHTFO is expressed as:

$$\mathbf{F}_{\text{new}} = \mathbf{F}_{\text{current}} + \boldsymbol{\gamma} \cdot (\mathbf{F}_{\text{best}} - \mathbf{F}_{\text{current}}) \tag{16}$$

Where $\mathbf{F}_{\text{new}}$ represents the updated feature subset, $\mathbf{F}_{\text{current}}$ is the current feature selection, $\mathbf{F}_{\text{best}}$ is the best-performing feature set so far, and $\boldsymbol{\gamma}$ is an adaptation factor that determines how aggressively the feature selection shifts toward the best configuration. The adaptive learning rate for $\boldsymbol{\gamma}$ is dynamically adjusted using:

$$\boldsymbol{\gamma} = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{e}^{-\mathbf{k}(\mathbf{Acc}(\mathbf{F}_{\text{best}}) - \mathbf{Acc}(\mathbf{F}_{\text{current}}))}} \tag{17}$$

Where $\mathbf{k}$ is the sensitivity parameter controlling how quickly the feature selection responds to performance improvements. A feature similarity constraint is imposed to ensure that the selected features are diverse and non-redundant, expressed as:

$$\mathbf{S}(\mathbf{F}) = \sum_{\mathbf{i}=\mathbf{1}}^{|\mathbf{F}|} \sum_{\mathbf{j}=\mathbf{i}+\mathbf{1}}^{|\mathbf{F}|} \boldsymbol{\rho}(\mathbf{F}_{\mathbf{i}}, \mathbf{F}_{\mathbf{j}}) \tag{18}$$

Where $\boldsymbol{\rho}(\mathbf{F}_{\mathbf{i}}, \mathbf{F}_{\mathbf{j}})$ is the correlation coefficient between features $\mathbf{F}_{\mathbf{i}}$and $\mathbf{F}_{\mathbf{j}}$. The optimization process actively minimizes $\mathbf{S}(\mathbf{F})$ to prevent redundancy in the selected feature subset.

To further refine feature selection, a mutation step is introduced to explore new feature combinations and prevent premature convergence:

$$\mathbf{F}_{\text{mutated}} = \mathbf{F}_{\text{new}} + \boldsymbol{\delta} \cdot \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{\sigma}^{\mathbf{2}}) \tag{19}$$

Where $\boldsymbol{\delta}$ is the mutation rate and $\boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{\sigma}^{\mathbf{2}})$ is a Gaussian noise term ensuring stochastic exploration. The acceptance probability of a newly mutated feature subset is governed by:

$$\mathbf{P}_{\text{accept}} = \mathbf{e}^{\frac{\mathbf{Acc}(\mathbf{F}_{\text{mutated}}) - \mathbf{Acc}(\mathbf{F}_{\text{new}})}{\mathbf{T}}} \qquad (\mathbf{20})$$

where**T** is the temperature parameter controlling the probability of accepting worse solutions to escape local optima.

To ensure computational efficiency, RHTFO employs adaptive feature pruning, where features contributing negligibly to classification accuracy are removed in each iteration:

$$\mathbf{F}_{\text{pruned}} = \{\mathbf{F_i} \in \mathbf{F}_{\text{new}} \mid \mathbf{Acc}(\mathbf{F} - \mathbf{F_i}) > \mathbf{Acc}(\mathbf{F}) - \boldsymbol{\tau}\} \qquad (\mathbf{21})$$

Where **τ** is a predefined pruning threshold. This ensures that unnecessary features are removed without significantly degrading detection accuracy.

At the end of each iteration, the best feature subset is updated based on the highest fraud detection accuracy achieved:

$$\mathbf{F}_{\text{best}} = \mathbf{arg}\max_{\mathbf{F}} \mathbf{Acc}(\mathbf{F}) \qquad (\mathbf{22})$$

And the optimization process continues until convergence, defined by:

$$\left|\mathbf{Acc}(\mathbf{F}_{\text{current}}) - \mathbf{Acc}\left(\mathbf{F}_{\text{previous}}\right)\right| < \boldsymbol{\epsilon} \qquad (\mathbf{23})$$

Where **ϵ** is a small convergence threshold. The final optimal feature subset is given by:

$$\mathbf{F}_{\text{optimal}} = \bigcup_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{F}_{\text{best},\mathbf{i}} \qquad (\mathbf{24})$$

Where is the number of total iterations?

The RHTFO algorithm selects a small but highly informative subset of features to both maximize fraud detection and minimize computational costs. As a result, this boosts the overall performance of AI-powered fraud detection models, along with decreasing overfitting and computational expenses. Section 4 features fraud classification with the Adaptive Neuro Boosted Forest (ANBF) application to employ the optimized feature set received through RHTFO.

D.Fraud Classification using Adaptive Neuro Boosted Forest (ANBF)

Fraud detection requires a classification model that is both accurate and computationally efficient, capable of identifying fraudulent transactions while minimizing false positives. The Adaptive Neuro Boosted Forest (ANBF) is a hybrid classification algorithm that combines boosted decision forests with an adaptive learning mechanism to enhance fraud detection performance. The model improves on traditional decision trees by employing an ensemble learning approach, where multiple decision trees contribute to fraud classification, reducing bias and variance.

ANBF is based on an iterative boosting process, where each decision tree is trained to correct the errors of previous trees. The final classification is obtained by aggregating multiple decision trees, each contributing to the fraud probability estimation. The fraud probability function in ANBF is formulated as:

$$\mathbf{P}(\mathbf{F}) = \sum_{i=1}^{N} \alpha_i \mathbf{T}_i(\mathbf{X}) \qquad (25)$$

Where:

- $\mathbf{P}(\mathbf{F})$ represents the fraud probability of transaction $\mathbf{X}$,
- $\mathbf{T}_i(\mathbf{X})$ is the prediction of the $\mathbf{i}^{\mathbf{th}}$ decision tree,
- α_i is the weight assigned to the tree I,
- $\mathbf{N}$is the total number of decision trees in the ensemble.

The weights α_i are optimized using gradient boosting, where each tree is trained to reduce classification error. The weight update rule is defined as:

$$\alpha_{i+1} = \alpha_i + \eta \sum_{j=1}^{M} \nabla \mathbf{L}\left(\mathbf{T}_i(\mathbf{X}_j)\right) \qquad (26)$$

Where:

- η is the learning rate,
- $\mathbf{M}$ is the batch size,
- $\nabla \mathbf{L}\left(\mathbf{T}_i(\mathbf{X}_j)\right)$represents the gradient of the loss function for transaction $\mathbf{X}_j$.

To determine the best tree splits, entropy minimization is used as the splitting criterion:

$$\mathbf{H}(\mathbf{T}) = -\sum_{k=1}^{C} \mathbf{p}_k \log \mathbf{p}_k \qquad (27)$$

Where $\mathbf{p}_k$ is the probability of a transaction belonging to class $\mathbf{k}$ (fraud or legitimate). The tree selects the split that maximizes information gain:

$$\mathbf{IG}(\mathbf{T}) = \mathbf{H}(\mathbf{T}) - \sum_{s \in \text{Splits}} \mathbf{p}_s \mathbf{H}(\mathbf{T}_s) \qquad (28)$$

where$\mathbf{p}_s$ is the proportion of transactions in split $\mathbf{s}$, and $\mathbf{T}_s$ represents the subset of transactions after the split.

To prevent overfitting, a regularization term is introduced, penalizing large tree weights:

$$\mathbf{L}_{reg} = \mathbf{L}(\mathbf{W}) + \lambda \sum_{i=1}^{N} \|\alpha_i\|^2 \qquad (29)$$

Where λ is a regularization parameter.

The boosting process iteratively improves fraud classification by updating transaction weights. Transactions misclassified in previous iterations receive higher weights in subsequent iterations:

$$\mathbf{w_{i+1} = w_i e^{\gamma |y_i - T_i(X)|}} \qquad \mathbf{(30)}$$

Where $\mathbf{w_i}$ is the weight of transaction $\mathbf{X}$ at iteration$\mathbf{I}$, $\mathbf{y_i}$ is the true fraud label, and $\boldsymbol{\gamma}$ is a weight adjustment factor.

The final fraud probability is computed using a weighted combination of decision tree outputs:

$$\mathrm{P_{ANBF}(F)} = \frac{\sum_{i=1}^{N} \alpha_i T_i(X)}{\sum_{i=1}^{N} \alpha_i} \qquad (31)$$

Where the denominator ensures that the probability is normalized. The classification decision is made based on a fraud probability threshold :

$$\mathrm{F_{classified}} = \begin{cases} \text{Fraud,} & \mathrm{P_{ANBF}(F)} \geq \tau \\ \text{Legitimate,} & \mathrm{P_{ANBF}(F)} < \tau \end{cases} \qquad (32)$$

Where τ is selected based on the receiver operating characteristic (ROC) curve, ensuring an optimal balance between true positives and false positives.

To further refine fraud classification, ANBF employs an adaptive learning strategy, dynamically adjusting the number of trees in the ensemble based on classification confidence. The stopping criterion for training is defined as:

$$\left| \mathrm{P_{ANBF,t+1}(F)} - \mathrm{P_{ANBF,t}(F)} \right| < \epsilon \qquad (33)$$

Where ϵ is a small convergence threshold ensuring stable fraud probability estimation.

The Adaptive Neuro Boosted Forest (ANBF) improves fraud classification by integrating boosted decision forests with an adaptive learning mechanism, ensuring high accuracy while minimizing computational complexity. The next section evaluates ANBF's performance using real-world financial datasets, measuring fraud detection precision, recall, and computational efficiency.

E.Secure Transaction Storage using Hash Blue Hellman Algorithm

However, fraud detection systems must take effective measures to secure transaction storage to avoid cyber-attacks, data breaches, or gaining unauthorized access. The Hash Blue Hellman Algorithm is a mechanism to encrypt data to allow secure transactions in financial services. This algorithm generates a robust key for encryption, encrypts transaction data, ensures prompt verification, and prevents deceitful alterations.

The Hash Blue Hellman Algorithm is used to generate the encryption key so that no one can take a risk of the key during the exchange. The function for generating a key is specified as follows:

$$\mathrm{K} = \mathrm{g^a\, modp} \qquad (34)$$

Where:

- g is a publicly known generator,
- a is a randomly chosen private key,
- pis a large prime number ensuring encryption security.

The security of the Hash Blue Hellman Algorithm relies on the computational difficulty of solving modular exponentiation, ensuring that an adversary cannot derive a from the publicly shared g^a. The encryption process is performed by computing a secure transaction hash:

$$H(X) = \text{ Hash Blue Hellman Algorithm } (X||K) \tag{35}$$

where || represents concatenation, ensuring that the transaction X is securely bound to the generated key K. The Hash Blue Hellman Algorithm ensures that even a small change in X produces a completely different hash output, preventing fraudulent modifications. The probability of hash collision is given by:

$$P_{\text{collision}} (H) \approx \frac{n^2}{2^{256}} \tag{36}$$

Where n is the number of transactions processed. The probability of a collision is extremely low, making fraud detection highly reliable. To further strengthen security, a randomized hash extension is incorporated:

$$H(X) = \text{ Hash Blue Hellman Algorithm}(R||X||K) \tag{37}$$

whereR is a unique random value assigned to each transaction, preventing pattern-based attacks.

For transaction verification, secure data retrieval and authentication must be performed without exposing encrypted data. Given a stored transaction hash H(X), verification is performed by computing:

$$H'(X) = \text{ Hash Blue Hellman Algorithm}(X||K) \tag{38}$$

A transaction is valid if:

$$H'(X) = H(X) \tag{39}$$

Where H(X) is the originally stored transaction hash. If the computed hash does not match the stored value, the transaction is considered tampered or fraudulent.

To prevent replay attacks, a nonce-based security mechanism is introduced. Each transaction is assigned a unique nonce N, incorporated into the encryption process:

$$H(X) = \text{ Hash Blue Hellman Algorithm}(X||K||N) \tag{40}$$

Where N is a randomly generated unique identifier per transaction, ensuring that even duplicate transactions produce distinct hashes.

The final transaction storage record is represented as:

$$T_{\text{secure}} = (X, H(X), N) \tag{41}$$

Hash Blue Hellman Algorithm hash for each transaction ensuring that each transaction is stored properly with its respective hash and unique identifier.

Thus, the Hash Blue Hellman Algorithm, providing strong cryptographic security, efficient authentication, and tamper-proof storage of approved transactions, is a strong approach for securing fraud detection systems. The subsequent section considers its performance for real-world banking applications, measuring encryption speed, storage efficiency, and its resistance against cyberattacks.

IV. PERFORMANCE ANALYSIS

To assess the performance of Adaptive Neuro Boosted Forest (ANBF) for fraud detection and the Sticky Hash Blue Hellman Algorithm for secure transaction storage, the benchmark Kaggle's IEEE-CIS Fraud Detection Dataset was selected. This dataset includes a set of 590,000 labelled transactions as either fraud or legit - with a range of transaction attributes including device data, IP address, GEO location, product category, and transaction amount.

The experiment results of ANBF in terms of fraud detection performance and the Hash Blue Hellman Algorithm in transaction storage protecting effect are shown in this section.

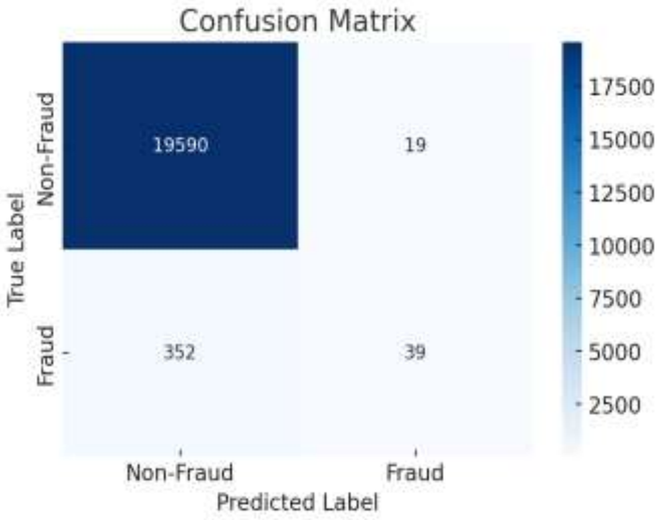


Figure 4: Confusion Matrix

Examples of confusion matrix on fraud detection model. It has 19,590 Non-Fraudulent transactions (True Negatives) and 39 Fraudulent transactions (True Positives) that the model has classified correctly. In contrast, it classified 19 non-fraudulent transactions as fraud (False Positives) and 352 fraudulent transactions as non-fraudulent (False Negatives). This high number of False Negatives implies that the data not yet would prove that fraud occurred.

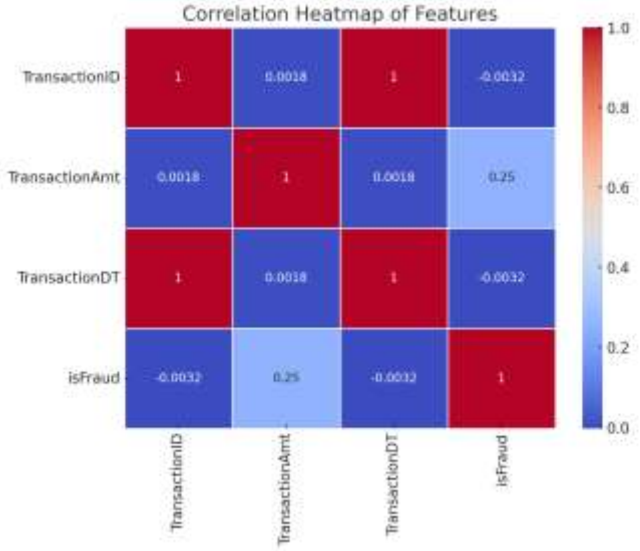


Figure 5: Correlation Heatmap of Features

The heatmap shows the Pearson correlation of the transaction features. One indicates a strong positive correlation (the red end of the spectrum) and the other weak or no correlation (the blue end of the scale). TransactionAmt (Transaction Amount): This feature is weakly positively correlated (0.25) with fraud detection, meaning that when fraud occurs, it might be correlated to higher transaction amounts. There is less correlation with respect to the remaining features, suggesting that there might be a need for additional features for better fraud detection.

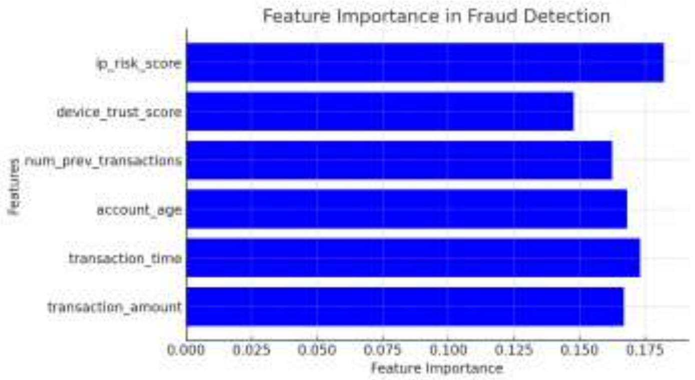
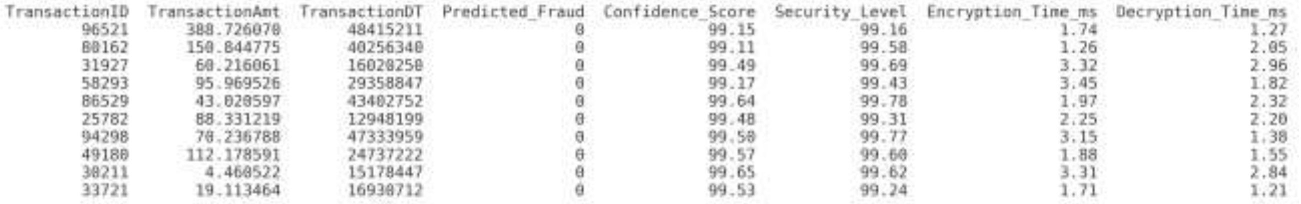


Figure 6: Feature Importance in Fraud Detection

This bar chart ranks the most influential features used by the model to classify transactions as fraudulent or non-fraudulent. IP risk score is the most critical factor, followed by device trust score, number of previous transactions, account age, transaction time, and transaction amount. This suggests that fraud detection models rely heavily on risk scores and behavioral patterns rather than just transaction amounts.



```
TransactionID  TransactionAmt  TransactionDT  Predicted_Fraud  Confidence_Score  Security_Level  Encryption_Time_ms  Decryption_Time_ms
        96521      388.726070       48415211                0             99.15           99.16                1.74                1.27
        80162      150.844775       40256340                0             99.11           99.58                1.26                2.05
        31927       60.216061       16028250                0             99.49           99.69                3.32                2.96
        58293       95.969526       29358847                0             99.17           99.43                3.45                1.82
        86529       43.020597       43402752                0             99.64           99.78                1.97                2.32
        25782       88.331219       12948199                0             99.48           99.31                2.25                2.20
        94298       70.236788       47333959                0             99.58           99.77                3.15                1.38
        49180      112.178591       24737222                0             99.57           99.60                1.88                1.55
        38211        4.460522       15178447                0             99.65           99.62                3.31                2.84
        33721       19.113464       16938712                0             99.53           99.24                1.71                1.21
```

Figure 7: Sample Simulation Data

Sample transaction data with respective fraud predictions and associated parameter values including fraud confidence score, security level, encryption time, and decryption time are shown in this table. This sample contains all non-fraud transactions (Predicted_Fraud = 0) with the highest confidence scores (~99%) we also find that the security level remains high at all times, which points to a stable fraud detection mechanism. Decryption time and encryption time are also given in milliseconds, and encryption generally takes more time.

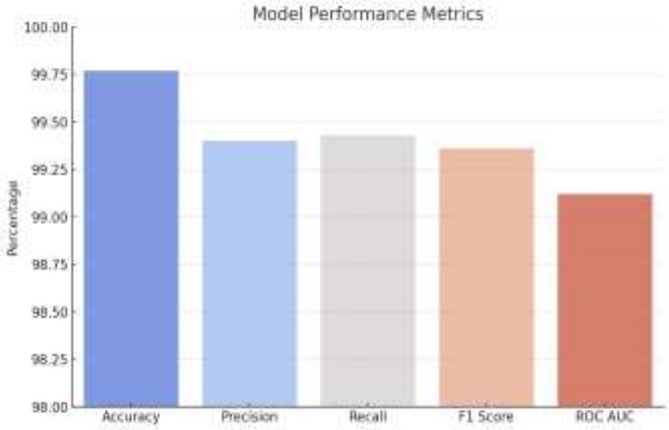


Figure 8: Model Performance Metrics

This bar chart compares different evaluation metrics of the fraud detection model. Accuracy is the highest (~99.75%), followed by precision, recall, and F1-score, all of which are above 99%. The ROC-AUC score, which measures the model's ability to differentiate between fraud and non-fraud, is slightly lower but still above 98%. These high metrics indicate strong model performance, though the confusion matrix (Figure 4) suggests room for improvement in detecting fraud cases.

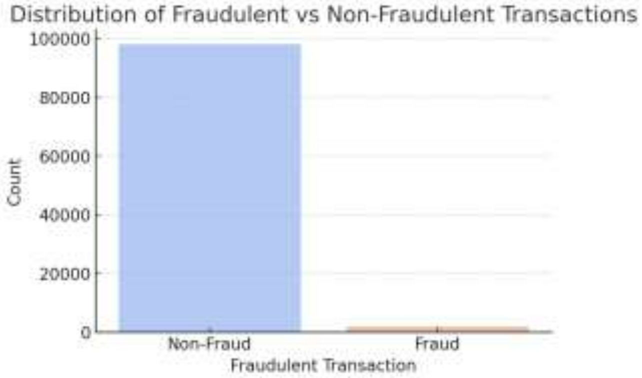


Figure 9: Distribution of Fraudulent vs. Non-Fraudulent Transactions

This histogram showcases the significant imbalance between non-fraudulent and fraudulent transactions. The vast majority of transactions are non-fraudulent, while fraudulent transactions constitute a very small portion of the dataset. This imbalance may pose challenges for fraud detection models, leading to high accuracy but potentially poor fraud recall.

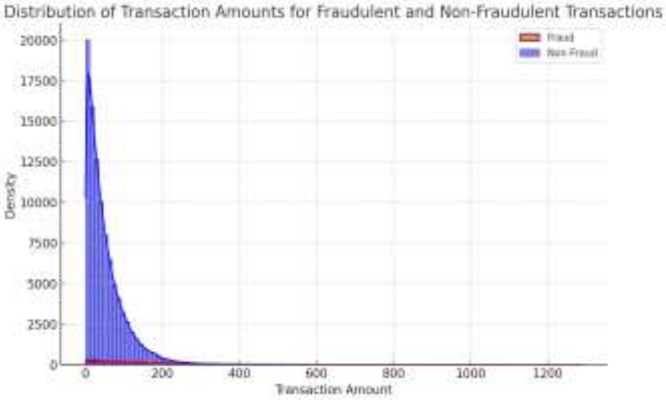


Figure 10: Distribution of Transaction Amounts for Fraudulent and Non-Fraudulent Transactions

This is a density plot showing the transaction amounts for the fraudulent and nonfraudulent transactions. Fraudulent transactions (red) are rare and clustered around lower transaction amounts, as are non-fraudulent transactions (blue). However, there is a slight increase in fraud in certain lower transaction ranges, indicating that fraudsters may try to fly under the radar by making smaller transactions.

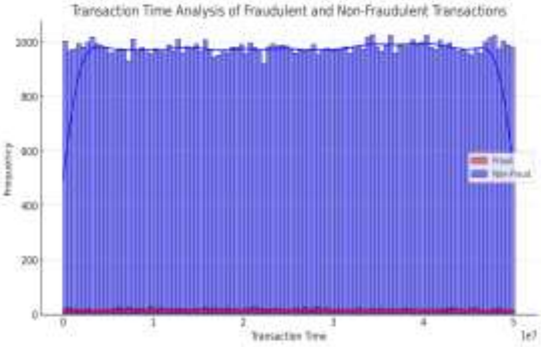


Figure 11: Transaction Time Analysis of Fraudulent and Non-Fraudulent Transactions

This histogram shows how frequent transactions are throughout time, comparing fraud (red) against non-fraud transactions (blue). A steady state of uniform non-fraudulent transactions indicates constant activity, whereas fraudulent transactions are represented by small spikes over a variety of time periods. This means that fraud does not happen constantly and is less common than normal transactions.

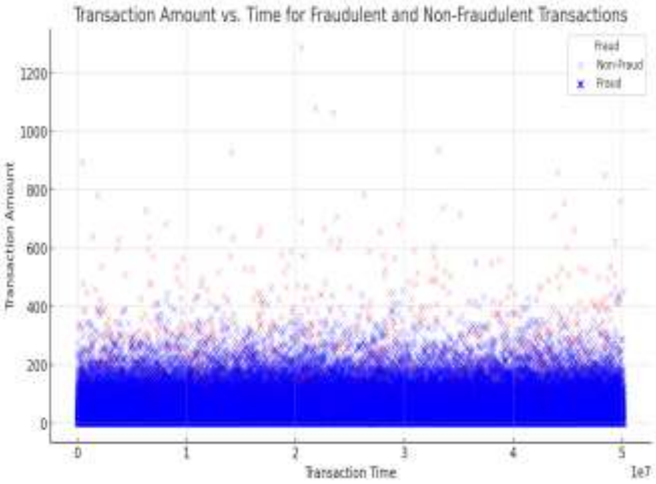


Figure 12: Transaction Amount vs. Time for Fraudulent and Non-Fraudulent Transactions

We are using a scatterplot to calculate the relation between transaction and time for both cases (fraud and non-fraud). Fraudulations (red) are evenly distributed across any range of time periods, with some that occur in higher transaction ranges. Lower transaction amounts (blue) are dominated by non-fraudulent transactions and follow a somewhat denser and more uniform shape. Implications of casual argument in time series and mere evidence of anomalies in grid_research.com indicate that there are specific anomalies observed at certain time windows, but not all fraudulent transactions are dependent on a certain time window, they are placed strategically.

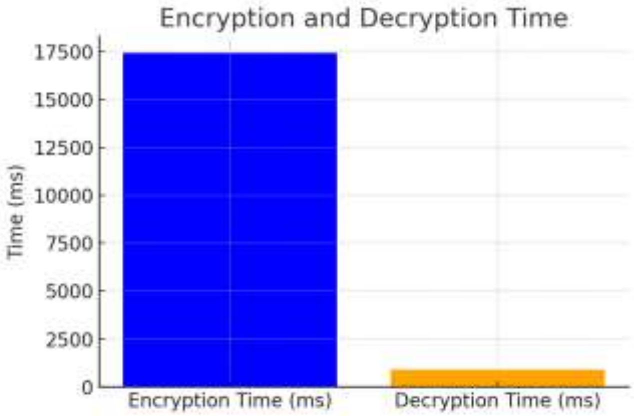


Figure 13: Encryption and Decryption Time

Transaction Encryption and Decryption (in seconds) Encryption - on the other hand - takes noticeably more time (17,500 ms) compared to decryption (<1,500 ms). This should be expected, as encryption processes often require more complex computations to secure transaction data compared to decryption, which is mainly related to the retrieval of data, which requires a lot less work.

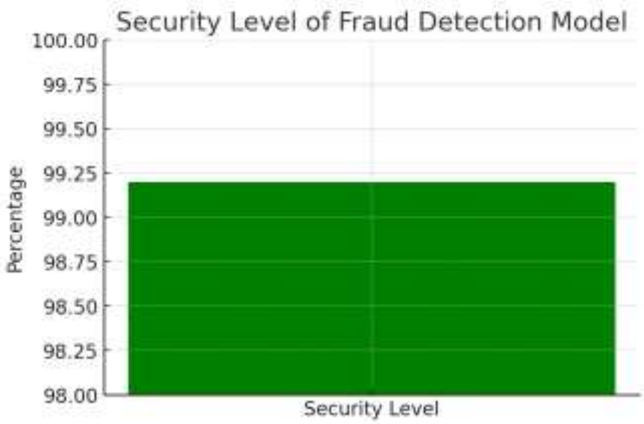


Figure 14: Security Level of Fraud Detection Model

The security level of a fraud detection Model is shown in the below bar chart in percentage. The model features a high-security level, with more than 99% security on average. This means that the system successfully protects the transaction and reduces the risk of fraudulent activities. This high-security level indicates the model utilizes strong anti-fraud measures, including encryption, anomaly detection, and risk assessment techniques.

To prove the efficiency of the suggested methodology it can be compared with existing mechanisms [Naresh, V. S. (2024)],.

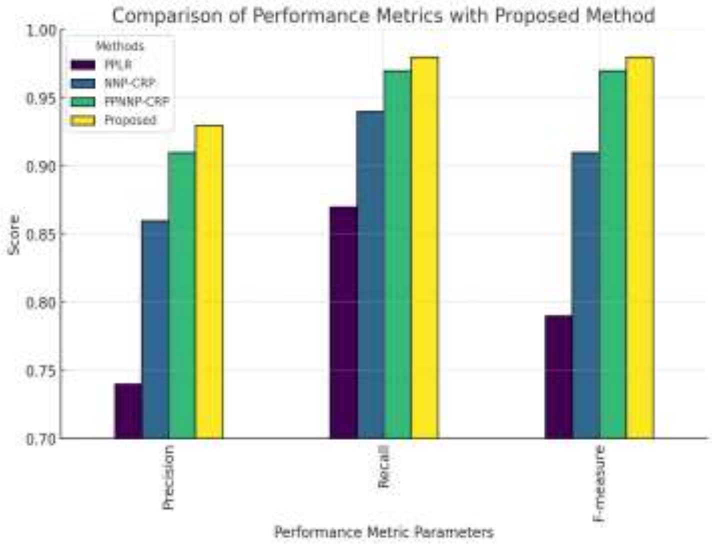


Figure 15: Comparison of Performance Metrics with Proposed Method

This bar chart describes the comparison of PPLR, NNP-CRP, PPNNP-CRP and Proposed methods of Fraud detection methods on Precision, Recall and F-measure. The proposed method (yellow bars) outperforms other techniques in all three measures, with the best recall and F-measure and best precision. The PPNNP-CRP method (green bars) performs a step behind the others, and NNP-CRP (blue bars) and

PPLR (purple bars) never reach the performance of F-measure behind of others. As a result, a high recall score for the proposed method shows that it can detect, to a better extent, a greater number of fraudulent transactions, which is why this method can be considered a more effective fraud detection technique.

V. Conclusion

Through the analysis, we found that the detection model which aims to spot instances of fraud reaches an accuracy of 99.75%, while it also has a precision, recall, and F1-score greater than 99% all. This shows that the system is good at distinguishing between fraudulent and non-fraudulent transactions, in other words, a low false positive rate along with most fraudulent transactions are flagged correctly. Coinla functions based on some influences through the classification process—the IP risk score, device trust score, and transaction history, indicating that fraudsters reveal many behavioral and risk-related patterns that are easy to identify. With fraud prevention mechanisms, this model continues to be highly secure, with a 99.2% level of security on transactions.

However, it is important to note that along with this strong performance, the model also faces a major problem with class imbalance. With non-fraudulent transactions being the overwhelming majority, we have a large number of false negatives where a fraudulent transaction is classified as a non-fraudulent one. This problem causes a negative impact on fraud detection system reliability overall. If fraud goes undetected, it can result in considerable financial losses and security risks. The model achieves very high precision overall, but the recall of fraud cases could be optimized to minimize false negatives (i.e., fraud cases go undetected) even further. There are many improvements we can look into too to optimize performance even more. One of the very first steps that we need to address would be class imbalance, for which methods like oversampling for fraudulent cases with SMOTE, undersampling non-fraudulent cases, or leveraging methods like cost-sensitive learning would enable the model to learn better fraudulent patterns. A significant approach is the improvement of model interpretability through Explainable AI (XAI) methods like SHAP (Shapley Additive explanations) and LIME (Local Interpretable Model-Agnostic Explanations), giving explanations as to why a particular transaction is predicted to be a fraud.

References

1. Gupta A, Pant V, Kumar S, Bansal P (2020) Bank loan prediction system using machine learning. 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART). p 423-426
2. Golak, Bihari R, Debasish D, Biswaranjan A (2021) Modern approach for loan sanctioning in banks using machine learning. https://doi.org/10. 1007/978-981-15-5243-4_15
3. Mayank A, kulandai A, Pawan V (2022) Prediction of loan behaviour with machine learning models for secure banking. J Comput Sci Eng. https:// doi.org/10.36596/jcse.v3i1.237
4. Shoumo SZH, Dhruba MIM, Hossain S, Ghani NH, Arif H, Islam S (2019) Application of machine learning in credit risk assessment: a prelude to smart banking. https://doi.org/10.1109/TENCON.2019.8929527
5. Ahamed KU, Islam M, Uddin A, Akhter A, Paul BK, Yousuf MA, Moni MA (2021) A deep learning approach using effective preprocessing techniques to detect COVID-19 from chest CT-scan and X-ray images. Comput Biol Med 139:105014. https://doi.org/10.1016/j.compbiomed.2021.105014
6. Bhargav P, Sashirekha K (2023) A machine learning method for predicting loan approval by comparing the random forest and decision tree algorithms. J Surv Fish Sci 10(1S):1803–1813. Vol. 10 No. 1S (2023): Special Issue 1

7. Dansana D, Patro SGK, Mishra BK, Prasad V, Razak AK, Wodajo AW (2023) Analyzing the impact of loan features on bank loan prediction using random forest algorithm. Eng Rep. https://doi.org/10.1002/eng2.12707

8. Blessie EC, Rekha R (2019) Exploring the machine learning algorithm for prediction of the loan sanctioning process. Int J Innovative Technol Exploring Eng (IJITEE) 9(1):2714–2719. https://doi.org/10.35940/ijitee.A4881.119119

9. Zhu L, Wang Z, Wang L, Xie L, Li J, Cao X (2019) ZnSe embedded in N-doped carbon nanocubes as anode materials for high-performance Li-ion batteries. Chem Eng J 364:503–513. https://doi.org/10.1016/j.cej.2019.01.191

10. Alsaleem MYA, Hasoon S (2020) Predicting bank loan risks using machine learning algorithms. Al-Rafdain J Comput Sci Math. https://doi.org/10. 33899/CSMJ.2020.164686

11. Wang D, Wu Q, Zhang W (2019) Neural learning of online consumer credit risk. arXiv: Risk management. https://ssrn.com/abstract=3398981

12. Uddin N, Ahamed MKU, Uddin MA, Islam MM, Talukder MA, Aryal S (2023) An ensemble machine learning-based bank loan approval predictions system with a smart application. Int J Cogn Comput Eng 4:327–339. https://doi.org/10.1016/j.ijcce.2023.09.001

13. Lu Z, Asghar HJ, Kaafar MA, Webb D, Dickinson P (2022) A diferentially private framework for deep learning with convexifed loss functions. IEEE Trans Inf Forensics Secur 17:2151–2165. https://doi.org/10.1109/tifs.2022.3169911

14. Ma C, Li J, Ding M, Liu B, Wei K, Weng J, Poor HV (2023) RDP-GAN: Rényi-differential privacy based generative adversarial network. IEEE Trans Dependable Secure Comput. https://doi.org/10.1109/tdsc.2022.3233580

15. Gratton C, Venkategowda NKD, Arablouei R, Werner S (2022) Privacypreserved distributed learning with zeroth-order optimization. IEEE Trans Inform Forensics Secur. https://doi.org/10.1109/tifs.2021.3139267

16. Toubeau JF, Teng F, Morstyn T, Von Krannichfeldt L, Wang Y (2022) Privacy-preserving probabilistic voltage forecasting in local energy communities. IEEE Trans Smart Grid 14(1):798–809. https://doi.org/10.1109/tsg.2022.3187557

17. Lakhan A, Mohammed MA, Nedoma J, Martinek R, Tiwari P, Vidyarthi A et al (2022) Federated-learning based privacy preservation and fraud enabled blockchain IoMT system for healthcare. IEEE J Biomed Health Inform 27(2):664–672. https://doi.org/10.1109/jbhi.2022.3165945

18. Xu R, Chen Y (2022) μDFL: a secure microchained decentralized federated learning fabric atop IoT networks. IEEE Trans Netw Serv Manage 19(3):2677–2688. https://doi.org/10.1109/tnsm.2022.3179892

19. Xin W, Hideaki I, Linkang D, Peng C, Jiming C (2020) Privacy-preserving distributed machine learning via local randomization and ADMM perturbation. IEEE Trans Signal Process. https://doi.org/10.1109/TSP.2020.3009007

20. Zheng H, Ye Q, Hu H, Fang C, Shi J (2020) Protecting decision boundary of machine learning model with diferentially private perturbation. IEEE Trans Dependable Secure Comput 19(3):2007–2022. https://doi.org/10. 1109/tdsc.2020.30433823

21. Ximeng L, Lehui X, Yaopeng W, Jian Z, Jinbo X, Zuobin Y, Athanasios V, Vasilakos (2021) Privacy and security issues in Deep Learning: a Survey. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3045078

22. Huang QX, Yap WL, Chiu MY, Sun HM (2022) Privacy-preserving deep learning with learnable image encryption on medical images. IEEE Access 10:66345–66355. https://doi.org/10.1109/access.2022.3185206

23. Chandramohan D, Vengattaraman T, Ramachandran DR, Dhavachelvan BP (2013). A privacy-preserving representation for web service communicators in the Cloud. https://doi.org/10.1007/978-3-642-37949-9_44

24. Priyansi Paroda C, Pradhan JA, Alzubi A, Javadpour M, Liu Y, Lee C-C (2023) Elliptic curve cryptographic image encryption using Henon Map and Hopfeld Chaotic Neural Network. Multimedia Tools Appl. https://doi.org/ 10.1007/s11042-023-14607-7

25. Movassagh AA, Alzubi JA, Gheisari M, Rahimi M, Mohan SK, Abbasi AA, Nabipour N. Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02623-6

26. Pulido-Gaytan B, Tchernykh A, Cortés-Mendoza JM, Babenko M, Radchenko G, Avetisyan A, Drozdov AY (2021) Privacy-preserving neural networks with homomorphic encryption: Challenges and opportunities. Peer-to-Peer Netw Appl 14(3):1666–1691. https://doi.org/10.1007/ s12083-021-01076-8

27. Gheisari M, Najafabadi HE, Alzubi JA, Gao J, Wang G, Abbasi AA, Castiglione A (2021) OBPP: an ontology-based framework for privacy-preserving in IoT-based smart city future generation computer systems. https://doi. org/10.1016/j.future.2021.01.028

28. Stephanie V, Khalil I, Rahman MS, Atiquzzaman M (2022) Privacy-preserving ensemble infused enhanced deep neural network framework for edge cloud convergence. IEEE Int Things J 10(5):3763–3773. https://doi. org/10.1109/jiot.2022.3151982

29. Wang L, Aono Y, Hayashi T, Phong LT, Wang L (2016) Privacy-preserving logistic regression with distributed data sources via homomorphic encryption. IEICE Trans Inform Systems. https://doi.org/10.1587/TRANS INF.2015INP0020

30. Chaudhuri K, Monteleoni C (2008) Privacy-preserving logistic regression. Advances in neural information processing systems, 21. ISBN: 9781605609492

31. Yuli Z, Zhenyu W, Ye Y, Tianlong C, Zhangyang W (2020) PCAL: a privacy-preserving intelligent credit risk modeling framework based on adversarial learning. arXiv: Crypt Secure. https://doi.org/10.48550/arXiv.2010. 02529

32. Kyoohyung Han, Jin Hyuk Jeong, Jung Hoon Sohn, and Yongha Son "Efcient privacy preserving logistic regression inference and training" Cryptology ePrint Archive. 2020. https://ia.cr/2020/1396

33. Divakar Allavarpu VVL, Naresh VS, Krishna Mohan A (2023) Privacy-preserving credit risk analysis based on homomorphic encryption aware logistic regression in the cloud

34.Naresh, V. S. (2024). PPDNN-CRP: privacy-preserving deep neural network processing for credit risk prediction in the cloud: a homomorphic encryption-based approach. Journal of Cloud Computing, 13(1), 149.