Journal of Global Research in Electronics and Communication
Volume 1, No. 7, July 2025
Available Online at: www.jgrec.info
RESEARCH PAPER

Optimizing Fraud Prevention in Financial Transactions using Scalable Machine Learning Models based on Credit Card Data

Amit Asthana
Postgraduate Student
Department of Computer Science and Engineering,
LNCT Group of Colleges
Bhopal, Madhya Pradesh, India
amit.asthana.me@gmail.com

Raj Kumar Sharma
Assistant Professor
Department of Computer Science and Engineering,
LNCT Group of Colleges
Bhopal, Madhya Pradesh, India
Rajkumar.s@Inct.ac.in

***Abstract*—Due to the increasing number of cyberattacks and frauds, especially in credit card transactions, fraud prevention in financial transactions has been even more important. The inherent difficulties of preventing fraud in financial transactions necessitate complex machine-learning in order to detect the frauds effectively and efficiently. This paper employs deep learning architectures, Fully Connected Neural Networks (FNN) and Convolutional Neural Networks (CNN), in order to classify fraudulent transactions based on European Customers Credit Card Transactions dataset. Various classification methods will be compared and contrasted in this study. With an accuracy of 99.87% and 99.61 over 30 trials, respectively, the suggested FNN and CNN considerably outperformed conventional models. Contrarily, during 10 experiments, the FNN achieved a high accuracy of 99.82 percent, and the CNN attained that of 99.81, indicating the stability/sturdiness of CNN. Although traditional models could be effective, their low recall and precision raised the chances of false negative results. Further evidence of the deep learning-based approach's dependability in real-world fraud detection situations was its enhanced precision, recall, F1-score, and AUC-ROC values. These results indicate an effectiveness of DL methods in the reduction of financial risks and increasing cybersecurity systems.**

Keywords*—Financial Fraud detection, credit card transactions, Fraud Prevention, Banking Fraud, Fraudulent Transactions, Financial Risk Management, Machine Learning, European customers Credit card transactions data.

I. INTRODUCTION

Technological growth has changed the banking and the financial sectors in a significant way. Electronic transactions, online banking, and card-less payment methods have transformed the system of financial servicing of the population, providing consumers with the opportunities of comfort and effectiveness[1][2][3]. Nonetheless, in addition to the above technologic development, the number of cases of financial fraud is also increasing and is a serious threat to individuals, companies, and financial organizations [4]. Losses in capital as well as consumer confidence and the stability of monetary systems are consequences of financial transaction fraud [5]. Fraud is defined as unauthorized taking or receiving money, goods, or services by a deceptive mode [6]. In the finance sector, fraud could be in different types, such as numeric fraud, money laundering, phishing, and cyber-attacks [7][8]. Financial fraud has become a serious topic worldwide, as hackers keep finding and exploiting the security defects to hack banking systems and payment networks [9]. The recent PricewaterhouseCoopers survey of 2022 revealed that half of all organizations in the world had experienced some kind of fraud, proving that fraud is a commonplace issue that plagues businesses and economies around the world [10].

Credit card fraud is one of the most common and fast growing forms of financial fraud. Fraud transactions involving stolen or fake credit card information have been on the increase as online shopping and using online payment services has become the norm [11]. The more complicated methods of accessing unauthorized credit card information include phishing, data breaches, and skimming by fraudsters[12]. Banks and customers both lose money as a consequence of these fraudulent operations, and financial institutions also lose trust [13]. To combat financial fraud, financial organizations have applied rule-based fraud detection systems which detect anomalous transactions according to preset patterns[14]. These techniques, however, often have trouble identifying novel fraud strategies, which outcomes in a high FPR and the loss of fraudulent transactions. Due to the changing nature of fraud, more sophisticated solutions are needed that can respond in real time to new threats[15].

AI[16], ML[17], and DL have revolutionized fraud detection and prevention by enabling systems to analyze large-scale financial transaction data and identify complex fraud patterns with high accuracy [18]. Machine learning models would be able to learn past fraud data and identify anomalies and classify the new transactions as fraud or legitimate in near real time. With complex patterns that conventional methods may fail to detect, DL methods like neural networks and autoencoders help to boost fraud detection level [19] [20][21] [22]. This work is about fraud prevention in financial transactions optimization using scalable machine learning models on credit card data [23]. Financial institutions can also better recognize fraudulent transactions, reduce false alarms, and increase the security of transactions, through the application of AI-driven methods. The study examines several ML and DL strategies, indicating their strengths in fighting financial fraud and meeting scalability requirements in the area of real-life banking.

A. Motivation and Contribution of Study

This research is propelled by the rising cases of financial fraud, particularly in credit card dealings which pose a grave financial and reputational risk to businesses and financial

services world over. The identification and proper classification of frauds is essential in reducing losses and financial security. Although traditional fraud detection techniques can be highly effective, they are limited to a scale and precision extent, and thus more advanced data-driven approaches must be adopted. This paper aims to offer a more straightforward, reliable and extensible solution to such issues by enhancing fraudulent transaction classification and detection through AI-powered ML and other approaches. The following are the most important findings from this study:

- To exploit the Kaggle dataset to develop a large-scale ML system capable for fraud credit card transactions detection.
- Use SMOTE and Nearmiss in correcting the imbalance in the classes, thus enhancing the performance of the model.
- Apply feature selection using Fisher Score to select the features important to the classification.
- To set up deep learning systems, such as FNN, CNN with optimizers tuned using Keras Tuner within Hyperband.
- To perform the comparison of architectures of different models and trial conditions (10 and 30 trials) in order to evaluate the effect of hyperparameter optimization.
- Determine the performance of the model through Geometric Mean, F1 score, recall, precision, and accuracy.

B. Justification and Novelty

This research takes a look at the rising problem of financial fraud and how DL models may help with effective and scalable detection. Machine learning approaches are critical to overcoming the ever-changing trends of fraud. The study proposes a new type of hybrid model that uses the combination of FNN and CNN architecture and is optimized by Keras Tuner with Hyperband using automatic hyperparameter optimization. It also incorporates SMOTE and NearMiss for class balancing and Fisher Score for feature selection, ensuring fair training and enhanced interpretability. A key novelty lies in evaluating multiple hyperparameter tuning trials (10 vs. 30) to determine the optimal fraud detection strategy. Comprehensive performance analysis using accuracy, precision, recall, F1-score, ROC curves, and precision-recall curves further strengthens the study's contribution to developing robust, real-world fraud detection systems.

C. Structure of Paper

Section II provides context for the work by discussing previous research on financial transaction fraud prevention. Section III explains the methodology, which includes how to prepare the data and choose the model. In Section IV, it gave the experimental data and performance analysis. In Section V, it finished the research and spoke about what the future holds.

II. Literature Review

In this section, the study reviews the existing literature on the classification and detection of financial fraud in credit card transactions. Most of the reviewed works summarized in Table I, focus on classification techniques and their effectiveness in fraud detection. Some of the notable reviews are:

Chaitanya et al. (2024) HNB and BBN use probabilistic reasoning and statistical analysis to classify transactions as fraudulent or not based on these features. The models are assessed using performance measures. Results show HNB achieving 86.87% accuracy, and BBN reaching 89.59%. Additionally, a comparison with other fraud detection approaches highlights HNB and BBN's competitive performance. Overall, this paper showcases their potential in accurately detecting credit card fraud, offering valuable tools for this endeavor [24].

Nti and Somanathan (2024) provide a framework for ensemble ML that detects financial crimes utilizing XGBoost and RF. Using Kaggle's IEEE-CIS fraud detection benchmark dataset, they put their approach through its paces. Accuracy = 0.9999, recall = 1.0, precision = 0.9965, and F1-Score = 0.9982 were the performance measures. A thorough evaluation, in contrast to cutting-edge methods, reveals the framework's scalability and resilience; these methods include DT, LR, GB, and PSO models [25].

Hashemi, Mirtaheri and Greco (2023) investigate the potential for modifying the weight assigned to legitimate and fraudulent transactions by use of hyperparameters controlling weight classes. With ROCAUC = 0.95, acc 0.79, rec 0.80, F1score 0.79, and MCC 0.79, the findings demonstrate that LightGBM and XGBoost successfully meet the best level requirement. Additionally, they use DL and the Bayesian optimisation approach to fine-tune the hyperparameters, obtaining the ROCAUC = 0.94, prec = 0.80, rec = 0.82, F1score = 0.81, and MCC = 0.81 [26].

Arram et al. (2023) investigate systems that predict credit card default using ML algorithms. The main goal is to determine which ML model works best with the proposed new credit card score dataset. MLP outperforms LR, DT, RF, LightGBM, and XGBoost according to TPR predictive performance, according to the experimental data, with an impressive AUC of 86.7%, an Acc rate of 91.6%, and a Rec rate above 80% [27].

Geetha et al. (2023), used ML techniques to create an API that can identify fraudulent credit card transactions and harmful URLs. This API can then host these files online, completely doing away with the requirement for users to download software packages locally. Their results are in line with those of previous models, which demonstrated an accuracy range of 70% to 90% [28].

Alsufyani et al. (2022) created an ML system that can identify credit card fraud using software. The first step in applying the Pearson correlation coefficient to determine which characteristics were worth include in Their model was data pre-processing. By adding more fraud data points, the SMOTE was able to correct the dataset's unbalanced data. To determine how well the model worked, they looked at its recall, accuracy, precision, and F1 score. The best recall score (88.55%) was achieved by SVM, which had a feature correlation of 0.1 [29].

Ahmed and Shamsuddin (2021) used Machine Learning (ML) methods in extensive experiments. It has integrated six ML approaches, i.e., To find the optimal mix of various classification methods, they use five performance metrics: accuracy, recall, AUC, precision, and fl-score. Together, the 99.99 percent AUC, fl-score, and 100% recall rate demonstrated that it was very accurate and precise [30].

TABLE I. SUMMARY OF BACKGROUND STUDY ON FRAUD PREVENTION IN FINANCIAL TRANSACTIONS USING MACHINE LEARNING

Author	Methodology	Dataset	Performance	Limitation/Future Work
Chaitanya et al. (2024)	HNB and BBN with probabilistic reasoning and statistical analysis	Financial Banking data	HNB Accuracy = 86.87%, BBN Accuracy = 89.59%	Limited comparison to other fraud detection methods Future work includes hybrid models for enhanced accuracy.
Nti, Somanathan (2024)	XGBoost and Random Forest, ADASYN oversampling, grid search algorithm with out-of-bag scoring	IEEE-CIS Fraud Detection (Kaggle)	F1-Score = 0.9982, AUC = 0.9999, Recall = 1.0, Precision = 0.9965, Accuracy = 0.9999	Limited to specific input features Expand feature set and test scalability across other datasets and domains.
Hashemi, Mirtaheri, Greco (2023)	Class weight-tuning hyperparameters, Bayesian optimization, majority voting ensemble learning	Bank data	ROC-AUC = 0.95, Precision = 0.79, Recall = 0.80, F1 Score = 0.79, MCC = 0.79	Not detailed; dataset information missing Extend ensemble techniques and explore additional hyperparameter tuning methods.
Arram et al. (2023)	ML models (LR, DT, RF, MLP, XGBoost, LightGBM), data preprocessing	New Credit Card Scoring Dataset	AUC = 86.7%, Accuracy = 91.6%, Recall > 80%	Dataset specifics and feature limitations not detailed Explore feature engineering and integration of additional ML algorithms.
Geetha et al. (2023)	ML-based APIs for malicious URL and fraud detection, hosted on the web	Fraud transactions data	General accuracy range: 70%–90%	Broad focus on multiple applications; limited details on specific fraud detection Optimize APIs for domain-specific use cases and improve interpretability of fraud detection results.
Alsufyani et al. (2022)	Pearson correlation for feature selection, k-fold cross-validation, SMOTE for data balancing	Credit card data	SVM Recall = 88.55%	Focused only on SVM for high recall, feature correlation threshold limited Test different correlation thresholds and integrate advanced ensemble methods.
Ahmed, Shamsuddin (2021)	Six ML techniques (LR, SVM, NB, RF, DT, KNN), oversampling technique	Fraud transaction data	RF with OS: Accuracy = 99.99%, AUC = 99.99%, F1-Score = 99.99%, Recall = 100%	Limited exploration of feature engineering Extend evaluation to larger datasets and incorporate deep learning models for comparison.

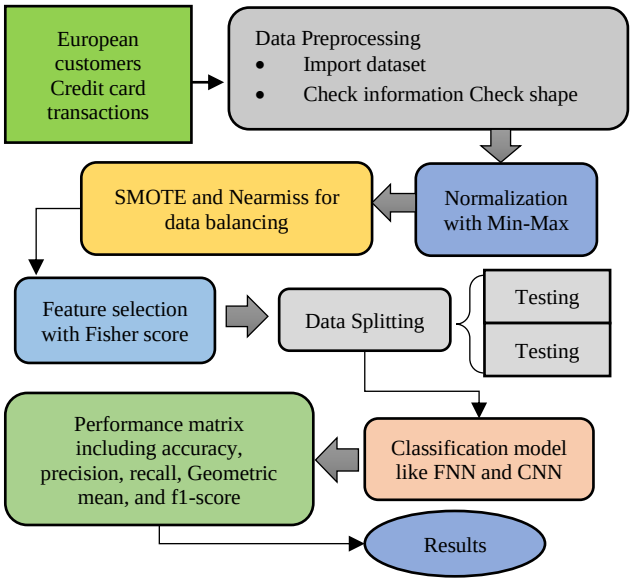

Fig. 1. Flowchart for Financial Fraud Detection

III. METHODOLOGY

The methodology for fraud prevention in financial transactions using machine learning models involves several key steps. Firstly collect the dataset from Kaggle named European customers Credit card transactions. The dataset is first preprocessed by handling missing values, applying Min-Max Scaling, and addressing class imbalance using SMOTE and NearMiss. The Fisher Score is used for feature selection in order to keep the most important qualities. Subsequently, the dataset was split into training (70%) and testing (30%) subsets to ensure robust evaluation. Scalable deep learning models, including FNN and CNN, are optimized using Keras Tuner with hyperparameter tuning strategies such as Hyperband. Multiple trials (10 and 30) are conducted to assess model performance based on Evaluation metrics, including confusion matrix, ROC curve, accuracy, precision, loss, geometric mean, recall, and f1-score are analyzed to compare model effectiveness. Finally, an accuracy comparison between different trials and models is performed to determine the most efficient fraud detection approach. The following systematic approach is illustrated in Figure 1.

Below are the main phases of the credit card fraud flow diagram:

A. *Data Collection*

For fraud detection classification and prediction, data collection is a very initial step. In this study, the dataset is sourced from Kaggle, which contains records of European customers' credit card transactions. The dataset includes multiple features and is used as the foundation for model training and testing. Figure 2 shows the data distribution.

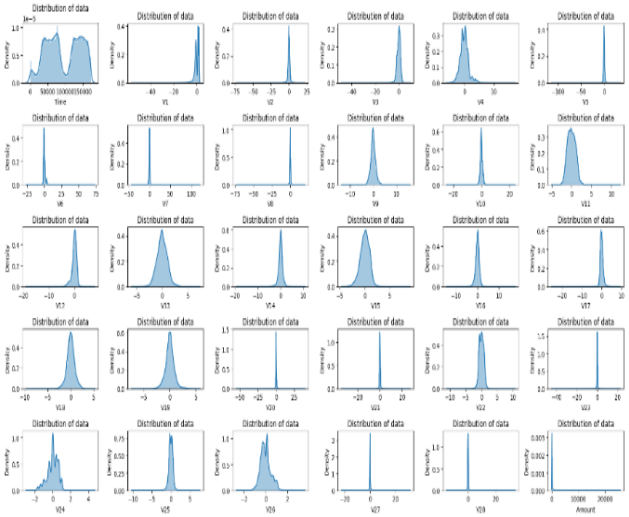

Fig. 2. Density plot for Description of the data

Figure 2 presents a series of density plots, likely representing the distributions of various features or variables within a dataset. Each subplot shows the distribution of a different feature using a kernel density estimation (KDE) plot.

The plots reveal a variety of distribution shapes, providing insights into the characteristics and potential relationships between the different features in the dataset.

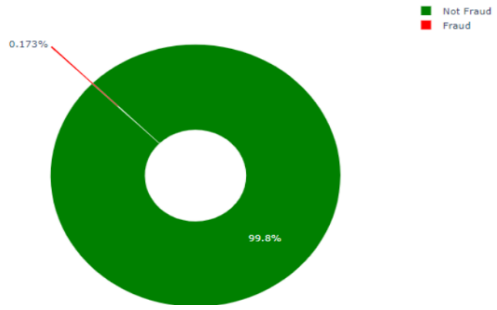

Fig. 3. Distribution of target column classes before balancing

Figure 3 is a pie chart that visually represents the class distribution of a dataset, likely related to fraud detection. The chart shows that the vast majority of cases (99.8%) are classified as "Not Fraud," while only a small fraction (0.173%) are labeled as "Fraud." ML models built on this data may be biased towards the dominant class due to the considerable class imbalance.

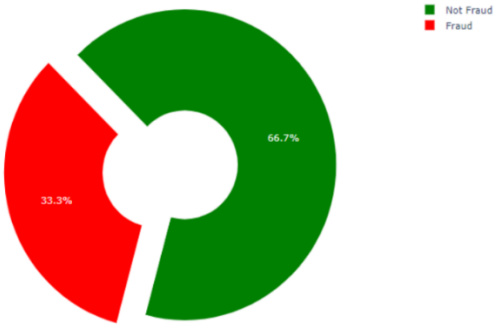

Fig. 4. Distribution of target column classes after data balancing

Figure 4 is a pie chart illustrating the class distribution of a dataset, likely related to fraud detection. The chart reveals that 33.3% of the cases are classified as "Fraud," while 66.7% are categorized as "Not Fraud." This suggests a less severe class imbalance compared to a dataset where the majority class dominates significantly. However, it still highlights the presence of an imbalanced distribution, which can pose challenges for machine learning models during training and evaluation.

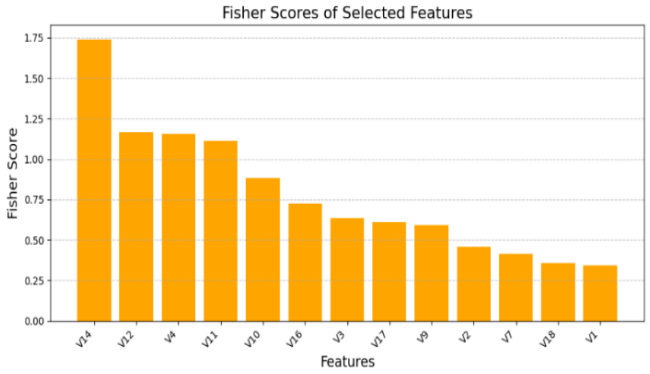

Fig. 5. Fisher Scores of Selected Features

A bar chart showing the Fisher Scores of certain traits is shown in Figure 5. A statistic called the Fisher Score is used to assess how well a feature can discriminate between several groups. In this chart, the features are ranked based on their Fisher Scores, with higher scores indicating greater discriminatory ability. The feature "V14" has the highest Fisher Score, suggesting it is the most effective feature for separating the classes in the dataset.

B. Data Preprocessing

Preprocessing data involves converting it into a format that data science tools like machine learning and data mining can work with more efficiently. The dataset was pre-processed by handling missing values (none found), applying Min-Max Scaling to the "Amount" column, and dropping the "Time" column. Class imbalance was addressed using SMOTE and NearMiss, reshaping the data to (341178, 29). Feature selection with the Fisher score. In this step, the data were processed in the following ways.

- **Import dataset:** Loaded the dataset using Pandas and performed an initial exploration (info(), describe()).
- **Check information:** Inspect the dataset's structure, including the number of rows, columns, and data types, to understand its contents.
- **Check shape:** Verify the number of rows (data instances) and columns (features) to understand the dataset's scale and complexity.

C. Normalization with min-max

The characteristics were rescaled from 0 to 1 using the MinMax Scaler approach in this study. This method's strength lies in its resilience to outliers; it employs statistical procedures that have no effect on the data's variance (Equation (1)).

$$x' = (x - min(x))/max(x) - min(x) \qquad (1)$$

The scaled value is denoted by x', the original value is represented by x, the maximum and minimum values of the feature are provided in Equation (1).

D. SMOTE and Nearmiss for Balancing

SMOTE and NearMiss are two effective techniques for handling imbalanced datasets for Fraud Prevention in Financial Transactions. For the minority class, SMOTE creates synthetic samples by extending existing instances, which improves model generalization and prevents overfitting. NearMiss [31] comprises a series of under-sampling methods that address unbalanced datasets by eliminating observations from the majority class that are geographically near to those from the minority class. As a first step, the NearMiss method determines the total distance between all observations that include both the bulk and minority groups.

E. Feature Selection with Fisher Score

Feature selection is a crucial process for improving model performance and decreasing computational complexity, especially in high-dimensional datasets [32]. A popular supervised feature selection method, the Fisher Score measures the discriminating strength of features according to their class separability. The chosen characteristics should improve classification performance, hence it evaluates the ratio of inter-class variation to intra-class variance. Mathematically, the Fisher Score for a given feature is computed as Equation (2):

$$F_i = \frac{\sum_{c=1}^{C} N_c(\mu_{c,i}-\mu_i)^2}{\sum_{c=1}^{C} N_c\sigma_{c,i}^2} \qquad (2)$$

where: C

- C is the number of classes,
- N_c is the number of samples in class c, i
- $c\mu_{c,i}$ and $\sigma_{c,i}^2$i are the mean and standard deviation of feature x_i within class
- μ_i is the overall mean of feature x_i .

F. Train-Test Split

There are two types of data: training and testing. The model is trained on the training set and its performance is assessed on the testing set. The split ratio of (70:30).

G. Models Selection

The proposed method includes deep learning models. Each models like FNN, and CNN is described in below:

1) Fully Connected Neural Network (FNN)

A kind of ANN called a FNN is composed of input, hidden, and output layers. The quantity of input and output parameters determines how many neurones are in a FNN's input and output layers, respectively [33]. The typical three-layer neural network's structure is depicted in Figure 6. If the data is very complicated, the number of hidden layers in a FNN model could go above ten, although in general, at least one is required [34]. Equation (3) shows the computation method for the three-layer neural network displayed in Figure 6. Each buried layer neuron's computation mostly entails activation and linear combination calculations. It is not common practice for the output layer to activate itself.

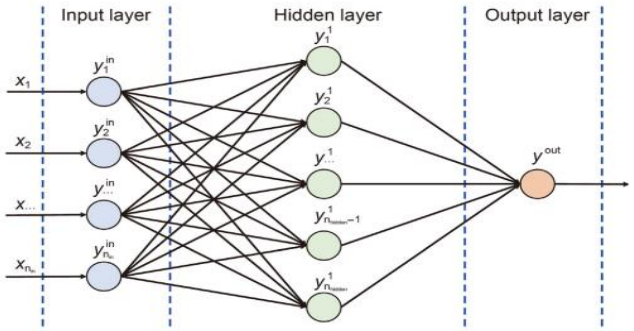

Fig. 6. Schematic diagram of the FNN model

$$\begin{cases} y_i^{in} = x_i, i \in [1, n_{in}] \\ y_i^1 = \varphi_{active}\left(\left(\sum_{k=1}^{n_{in}} w_{k,i}^1 \times y_k^{jn}\right) + b_i^1\right) \imath\epsilon[1, n_{hidden}] \\ y^{out} = \left(\sum_{k=1}^{5} w_k^{out} \times y_k^1\right) + b_i^{out} \end{cases} \quad (3)$$

In Equation (3), n_{in} indicates how many parameters are being entered, n_{hidden} is the total amount of neurones in the hidden layer. $w_{k,i}^1$ is the weight of the input layer's neurone k's output [35] when the hidden layer's neurone I executes linear combination, and b_i^1 is the bias. 4active is the activation function. The FNN model was implemented using Keras Tuner with the Hyperband algorithm for hyperparameter optimization over 30 trials. The tuning process explored dense layer sizes ranging from 32 to 256 in steps of 32 and dropout rates between 0.2 and 0.5 in increments of 0.1. Additionally, the learning rate was selected from predefined values of 0.1, 0.01, and 0.001. The maximum number of epochs was set to 10, with a reduction factor of 3 to balance exploration and exploitation in the search process.

2) Convolutional Neural Network (CNN)

A CNN is a DL model for image processing that attempts to imitate the anatomy of the visual brain in animals. Its primary function is to facilitate auto- and adaptive-feature-hierarchy learning moving from basic to advanced patterns. CNNs primarily function to grasp the important aspects of the incoming data. Layer one of this approach consists of learnable filters applied to a collection of convolutional feature extractors. The CNN layers are built using convolutional kernels that produce different feature maps. Regional connections to neurones are shown in the feature map of the layer below the present one [36]. A feature map can't be made without sharing the kernel across all input locations. A fully connected layer (or layers) is used to complete the classification process once the convolutional and pooling layers have been established. Equation (4) depicts the procedure performed on CNNs using input feature maps:

$$h_j^{(n)} = \sum_{k=1}^{k} h_k^{(n-1)} \otimes \imath w_{kj}^{(n)} + b_{kj}^{(n)} \quad (4)$$

where $\otimes$ a 2D convolution and $h_j^{(n)}$ represents the result of the jth feature map in the nth hidden layer. Meanwhile, $h_k^{(n-1)}$ denotes symbolizesannel of the $(n-1)^{th}$ hidden layer, $\imath w_{kj}^{(n)}$ symbolises the values of the kth channel inside the jth filter of the nth layer, and $b_{kj}^{(n)}$ denotes the relevant bias term. An iterative technique that switches between feed-forward and backpropagation data movements is used to finish CNN training [37]. Backpropagation is a continuous process of tweaking the convolutional filters and fully connected layers. Equation (5) shows that the key aim is to decrease the average loss EE for all true class labels and network outputs.

$$E = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{c} \hat{y}_i^{(k)}\, log(y_i^{(k)}) \quad (5)$$

where $\hat{y}_{\text{i}}^{(k)}$denotes a true label, and $\hat{y}_{\text{i}}^{(k)}$ denotes a network output of the i^{th} input in the k^{th} class. Furthermore, mm denotes the training input and cc denotes the neurones in the output layer [38]. The CNN model was optimized using Keras Tuner's Hyperband algorithm over 30 trials, exploring various hyperparameters. The first convolutional layer (Conv_1) had filter sizes ranging from 32 to 128 with kernel sizes of (3,5), while the second convolutional layer (Conv_2) had filter sizes between 32 and 64 with the same kernel options. The dense layer size varied between 32 and 128 in step 16, and dropout rates were tuned from 0.2 to 0.5 in increments of 0.1. The LearningRate was chosen from 1e-2, 1e-3, and 1e-4. A reduction factor of three was used to optimise the tuning process while training the model across a maximum of ten epochs.

H. Performance Matrix

A common technique for evaluating a model's performance in classification tasks is a confusion matrix. The two possible outcomes of a binary classification problem are the "positive" and "negative" categories. Here are the parts that make up a confusion matrix:

- **TP (True Positives):** accurately detected positive instances),
- **TN (True Negatives):** accurately recognised negative situations),
- **FP (False Positives):** situations of negativity that are mistakenly categorised as positive),
- **FP (False Negatives):** positive situations that were mistakenly labelled as negative. To assess a model's

performance, that may get the following matrices from these elements:

1) Accuracy

As a statistic for measuring performance, accuracy is defined as the proportion of accurate fraud predictions to total model predictions (including non-fraud forecasts). It is calculated with the following Equation (6):

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \quad (6)$$

2) Recall

A statistic called recall/sensitivity counts how many out of a total of fraud transactions were properly categorized (TP). It is calculated as Equation (7):

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

3) Precision

The precision ratio is defined as the percentage of predicted transactions that are really fraudulent divided by the total number of transactions (TP + FP). It is represented as Equation (8):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

4) F1 Score

An F1 score is a weighted average of two metrics: recall and precision. A value near to one indicates the greatest value, and it ranges from zero to one. It used the phrase to calculate. The F1 score may be defined in Equation (9):

$$F1 = \frac{2*(\text{precision}*\text{recall})}{\text{precision}+\text{recall}} \quad (9)$$

5) Geometric mean

To calculate the geometric mean, first take all of the numbers and multiply them by themselves. Then, multiply that result by the square of the number of samples used to create the product. To determine the geometric mean formally, one uses Equation (10):

$$Geometric\ Mean = \sqrt[\frac{1}{n}]{\prod_{i=1}^{n} x_i} \quad (10)$$

where x_i is the datapoint n is the number of datapoints in the set.

ROC AUC score: A simple metric that shows how well a model performs across multiple probability thresholds is the ROC AUC score, which measures the area under the ROC curve. The formula for the AUC score is shown in Equation (11):

$$AUC = \frac{1+TP-FP}{2} \quad (11)$$

It demonstrates that increasing the number of TPs does not necessitate increasing the number of FPs in any model.

IV. Results And Discussion

To perform the research work used Python programming language and Jupyter Notebook development environment. The pre-processing requires a system that meets the following requirements. All of these components, including an Intel (R) Core (TM) i3-6100U CPU operating at 2.30GHz and 2304 MHZ, 256 GB of solid-state drive capacity, 8 GB of RAM, and 4 logical processors, operate together with the processor. Results from testing the proposed models on a dataset consisting of European consumers' credit card transactions are presented in this section. Table II displays the results of the models' evaluation using a performance matrix that includes f1-score, G-mean, recall, accuracy, and precision.

TABLE II. Performance Metrics of CNN and FNN Models on European Customers Credit Card Transactions Data

Performance Metric	30 Trials		10 trials	
	CNN	*FNN*	*FNN*	*CNN*
Accuracy	99.61	99.87	99.82	99.81
Precision	99.61	99.87	99.82	99.81
Recall	99.61	99.87	99.82	99.81
F1 score	99.61	99.87	99.82	99.81
Geometric mean	1.0	1.0	1.0	1.0

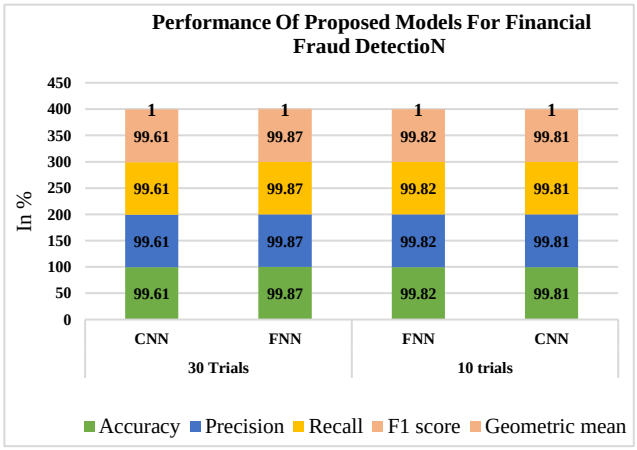

Fig. 7. Bar Graph for Proposed Models Performance

Table II and Figure 7 presents the performance of CNN and FNN models, demonstrating exceptional classification accuracy. With 30 trials, CNN achieves 99.61% across accuracy, precision, recall, and F1-score, while FNN outperforms slightly with 99.87% in each metric. In the 10-trial evaluation, FNN maintains a high performance of 99.82%, closely followed by CNN at 99.81% for all metrics. Both models achieve a perfect geometric mean of 1.0, indicating their robustness in detecting fraudulent transactions. The findings show that CNN and FNN work well to identify credit card fraud with high accuracy.

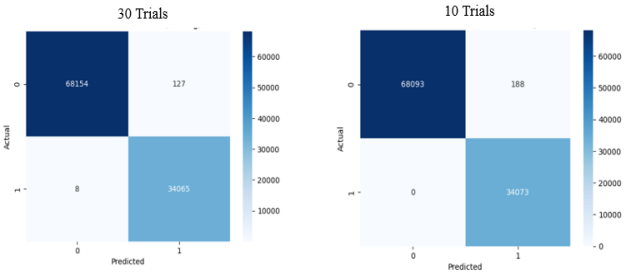

Fig. 8. Confusion matrix for FNN Models

Figure 8 shows two confusion matrices comparing the performance of FNN Models under different numbers of trials - 30 trials (left) and 10 trials (right). Both matrices use a color-coded format where darker blue represents higher values and lighter blue represents lower values. In the 30 trials matrix, can observe values of 68154 and 127 along the diagonal (TP and TN), while the 10 trials matrix shows values of 68093 and 188. The matrices are labeled with "Predicted" on the x-axis and appear to use a similar scale, with values ranging up to around 60000. The similar blue shading and numerical distributions indicate comparable performance between the two trial conditions, with minor differences.

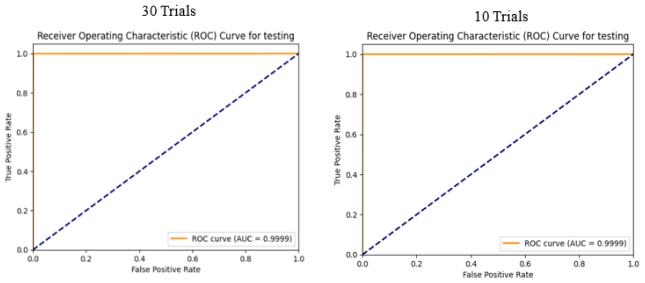

Fig. 9. ROC Curve for FNN Models

Figure 9 displays the ROC curves for FNN Models under two different conditions - 30 trials (left) and 10 trials (right). Both graphs plot the TPR against the FPR, with a dashed diagonal line representing random chance performance. Each graph shows an ROC curve in orange with its corresponding AUC value of approximately 0.9999. The nearly identical AUC values and curve shapes between the 30-trial and 10-trial conditions suggest that the model performs exceptionally well and consistently regardless of the number of trials, as both curves hug the top-left corner of the plot, which indicates near-perfect classification performance.

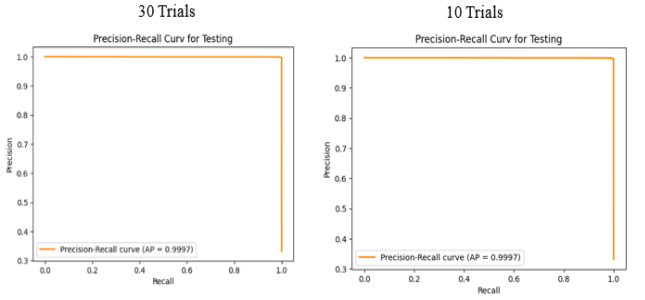

Fig. 10. Precision-Recall Curve for FNN Models

Figure 10 presents the Precision-Recall (PR) curves for FNN models under two conditions: 30 trials (left) and 10 trials (right). Both curves exhibit nearly identical performance, with an Average Precision (AP) score of 0.9997, indicating exceptional classification capability. The PR curves maintain a precision of 1.0 across most recall values before sharply dropping at high recall, a characteristic of a well-performing classifier. The resemblance between the two conditions implies that, independent of trial count, the model continuously produces high precision and recall.

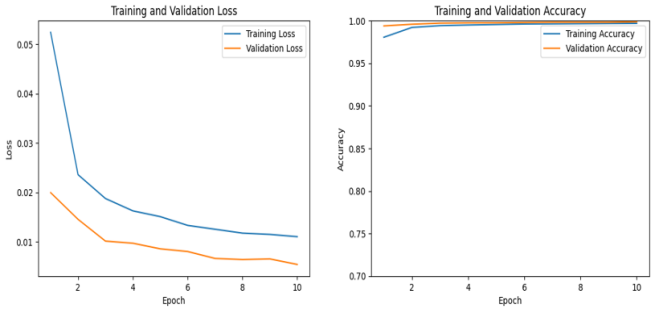

Fig. 11. Accuracy and loss for FNN in 30 trials

After 30 trials, the FNN model's training and validation results are shown in Figure 11. An increasing decrease in the training and validation loss as the epoch count rises, as seen in the left figure, indicates efficient learning and less error. The right plot presents training and validation accuracy, both of which remain high throughout the epochs, suggesting strong generalization and stable performance. The minimal gap between training and validation curves in both plots highlight the model's robustness and absence of significant overfitting.

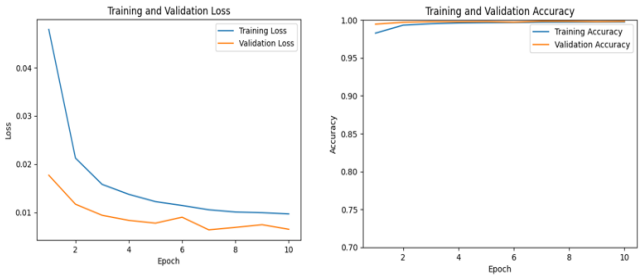

Fig. 12. Accuracy and loss for FNN in 10 trials

The accuracy (right) and training and validation loss (left) for the FNN model across 10 trials are shown in Figure 12. A steady decrease in training and validation loss, as shown in the loss graph, indicates that the model is learning well. Minimal overfitting is shown by the accuracy graph, which shows a high and consistent training accuracy closely followed by validation accuracy. The model has impressive generalizability within the 10-trial configuration, according to the findings.

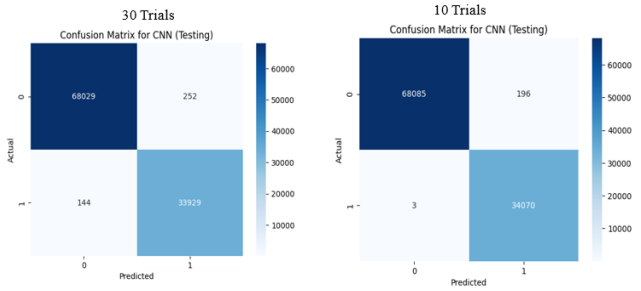

Fig. 13. Confusion matrix for CNN Models

Figure 13 presents confusion matrices for the CNN model evaluated on the dataset after 30 and 10 trials, respectively. In the 30-trial scenario, the model correctly classified 68,029 legitimate transactions and 33,929 fraudulent transactions, with 252 FP (legitimate transactions misclassified as fraud) and 144 FN (fraudulent transactions misclassified as legitimate). In contrast, the 10-trial configuration resulted in 68,085 TN and 34,070 TP, with only 196 FP and 3 FN. The significant reduction in FN in the 10-trial scenario suggests improved fraud detection sensitivity, which is critical for minimizing financial risks. This indicates that fewer trials may enhance model generalization and convergence, warranting further investigation.

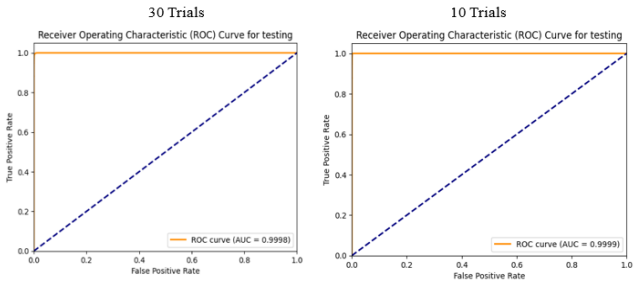

Fig. 14. ROC Curve for CNN Models

The CNN model's ROC curves on the CCFD dataset after 30 and 10 trials are shown in Figure 14. Thanks to their respective AUC values of 0.9998 and 0.9999, both models

demonstrate almost flawless categorization. There is a clear separation between valid and fraudulent transactions, according to the ROC curves. The minimal difference in AUC suggests that fewer training trials do not significantly impact performance. These results confirm the model's high effectiveness in fraud detection.

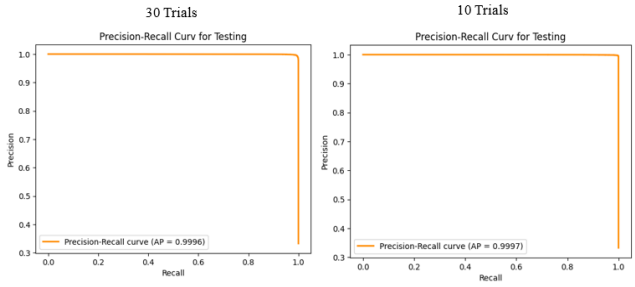

Fig. 15. Precision-Recall Curve for FNN Models

Figure 15 presents the Precision-Recall (PR) curves for the CNN model on the CCFD dataset after 30 and 10 trials. The PR curves illustrate the trade-off among precision and recall, highlighting model performance in handling imbalanced data. The average precision (AP) values for the 30-trial and 10-trial models are 0.9996 and 0.9997, respectively, indicating near-perfect classification. The minimal difference in AP suggests that fewer training trials do not significantly affect precision-recall performance. This proves that the model is quite accurate at spotting fraudulent transactions.

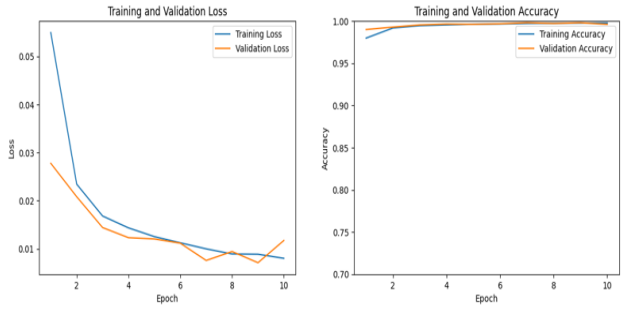

Fig. 16. Accuracy and loss for CNN in 30 trials

Figure 16 shows the CNN model's accuracy (right) and loss (left) curves for 30 trials on the CCFD dataset, as well as validation (left). When it comes to training, the loss starts at around 0.06 and drops to about 0.05; when it comes to validation, it starts at about 0.03 and stays around 0.01. The accuracy curves indicate strong performance, with training accuracy starting at approximately 97% and reaching nearly 99.9%, while validation accuracy follows a similar trend, converging at around 99.8%. These results confirm the model's high reliability and minimal overfitting, making it effective for fraud detection.

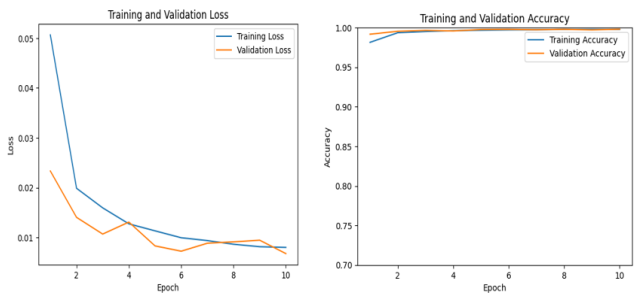

Fig. 17. Accuracy and loss for CNN in 10 trials

Figure 17 shows the CNN model's accuracy (right) and training loss (left) curves using the CCFD dataset across 10 trials. Training loss begins at about 0.05 and goes down to about 0.004 whereas validation loss starts at about 0.025 and settles around 0.008. Accuracy curves indicate a good model performance, where the training accuracy begins at around 97 percent and ends at 99.95 percent, whereas the validation accuracy represents a similar pattern, converging at around 99.9 percent. These results indicate that the model effectively learns and generalizes well even with fewer training trials, maintaining high reliability in fraud detection.

A. Comparison and Discussion

The following proposed models are compared with existing models with same dataset and performance matrix. The proposed models are trained on the 30 and 10 trials while existing model train on the 100 and 70 trials. The following comparison are shown in Table III.

TABLE III. COMPARISON OF PROPOSED AND BASELINE MODELS FOR FRAUD DETECTION ON DATASET

Propose models				
Performance Metric	**30 Trials**		**10 trials**	
	CNN	***FNN***	***FNN***	***CNN***
Accuracy	99.61	99.87	99.82	99.81
Precision	99.61	99.87	99.82	99.81
Recall	99.61	99.87	99.82	99.81
F1 score	99.61	99.87	99.82	99.81
Geometric mean	1.0	1.0	1.0	1.0
Base models				
Performance Metric	**100 Trials**		**70 trials**	
	RNN	***ANN***	***RNN***	***ANN***
Accuracy	93.24	92.22	91.21	91.89
Precision	97.72	93	92.85	95.52
Recall	88.35	91.09	89.04	87.67
F1 score	92.80	92.04	90.90	91.42
Geometric mean	92.92	92.04	90.90	91.51

The above Table III shows the comparison between base and proposed models' performance. In this comparison, CNN and FNN, demonstrate superior performance across all metrics. With 30 trials, CNN achieves an accuracy, precision, recall, and F1-score of 99.61%, while FNN outperforms slightly with 99.87% in each metric. In the 10-trial evaluation, FNN records 99.82%, and CNN follows closely with 99.81% for all metrics. Both models achieve a perfect geometric mean of 1.0, indicating robust classification performance. In contrast, the base models, evaluated over 100 and 70 trials, exhibit lower accuracy, with RNN achieving 93.24% (100 trials) and 91.21% (70 trials), while ANN records 92.22% (100 trials) and 91.89% (70 trials). Precision is notably higher for RNN at 97.72% in 100 trials but drops to 92.85% in 70 trials, whereas ANN maintains relatively stable precision around 93–95%. Recall is lower for base models, especially for RNN at 88.35% (100 trials) and 89.04% (70 trials), while ANN fluctuates between 87.67% and 91.09%. The F1-score and geometric mean follow a similar pattern, confirming that CNN and FNN provide superior performance, making them the optimal choices for high-accuracy classification tasks.

The proposed CNN and FNN models demonstrate significant advantages over traditional RNN and ANN models for CCFD, achieving exceptionally high accuracy, precision, recall, and F1 scores, even with fewer training trials. Both models maintain a perfect geometric mean of 1.0, indicating robust classification performance and strong generalization capabilities. In comparison to the baseline models trained with 100 and 70 trials (having lower accuracy and unreliable recall

values), CNN and FNN would have stable results with minimal FP and FN. Specifically, the models have good tradeoffs between training time and classification accuracy, thereby qualifying them as being ideal to use in a real-time fraud detection system. In addition, their dependability in managing unbalanced datasets is shown by the improved AUC and precision-recall curves. This ensures the precise detection of fraud with the least amount of financial threats.

V. Conclusion And Future Scope

Financial transaction fraud has grown in recent years into a major international problem, endangering the safety of financial institutions and resulting in huge monetary losses. The widespread use of electronic payment systems has led to a dramatic increase in fraud, especially with credit cards. This research explores the use of scalable ML models to identify fraudulent financial transactions. It specifically examines credit card transaction data from European customers. Credit card fraud detection is an area where the suggested CNN and FNN models shine, consistently obtaining high levels of accuracy and reliability in various experiments. After 30 trials, the FNN model achieved 99.87% accuracy, precision, and recall, whereas the CNN model achieved 99.61% accuracy across all parameters. During the course of 10 trials, the CNN model attained an F1-score, precision, recall, and accuracy of 99.81%, whereas the FNN model attained an F1-score and accuracy of 99.82%. The consistently high performance across multiple runs underscores the robustness of these models. However, a key limitation is their dependency on a specific dataset, which may impact generalization to real-world financial transactions with evolving fraud patterns. Additionally, the computational cost of CNNs remains a challenge for real-time deployment. To improve interpretability and flexibility in ever-changing financial contexts, future studies should investigate combining deep learning models with explainability approaches like SHAP, as well as real-time fraud detection systems.

References

[1] S. J. Wawge, "A Survey on the Identification of Credit Card Fraud Using Machine Learning with Precision , Performance , and Challenges," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 4, p. 8, 2025.

[2] N. Malali, "AI Ethics in Financial Services: A Global Perspective," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 2, 2025, [Online]. Available: https://www.ijisrt.com/assets/upload/files/IJISRT25FEB316.pdf

[3] S. Chitraju Gopal Varma and B. Chaudhari, "Federated Learning in Financial Data Privacy A Secure AI Framework for Banking Applications-edited," 2025. doi: 10.63282/3050-9246.ICCSAIML25-112.

[4] M. Shah, P. Shah, and S. Patil, "Secure and Efficient Fraud Detection Using Federated Learning and Distributed Search Databases," in *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, IEEE, Feb. 2025, pp. 1–6. doi: 10.1109/ICAIC63015.2025.10849280.

[5] S. Tyagi, "Analyzing Machine Learning Models for Credit Scoring with Explainable AI and Optimizing Investment Decisions," vol. 5, no. 01, pp. 5–19, 2022, [Online]. Available: http://arxiv.org/abs/2209.09362

[6] S. B. Shah, "Improving Financial Fraud Detection System with Advanced Machine Learning for Predictive Analysis and Prevention," pp. 2451–2463, 2024.

[7] Suhag Pandya, "A Machine and Deep Learning Framework for Robust Health Insurance Fraud Detection and Prevention," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 1332–1342, Jul. 2023, doi: 10.48175/IJARSCT-14000U.

[8] H. Kali, "Optimizing Credit Card Fraud Transactions Identification And Classification In Banking Industry Using Machine Learning Algorithms," *Int. J. Recent Technol. Sci. Manag.*, vol. 9, no. 11, pp. 1–12, 2024.

[9] E. Pan, "Machine Learning in Financial Transaction Fraud Detection and Prevention," *Trans. Econ. Bus. Manag. Res.*, vol. 5, pp. 243–249, 2024, doi: 10.62051/16r3aa10.

[10] A. Kumar, S. Bansal, and R. Hooda, "A Review Paper on Feature Selection in Credit Card Fraud Detection," *SSRN Electron. J.*, 2024, doi: 10.2139/ssrn.4502031.

[11] S. N. Pundkar and M. Zubei, "Credit Card Fraud Detection Methods: A Review," *E3S Web Conf.*, vol. 453, pp. 1–11, 2023, doi: 10.1051/e3sconf/202345301015.

[12] Sourabh and B. Arora, "A Review of Credit Card Fraud Detection Techniques," *Lect. Notes Electr. Eng.*, vol. 832, no. May, pp. 485–496, 2022, doi: 10.1007/978-981-16-8248-3_40.

[13] H. Sinha, "An examination of machine learning-based credit card fraud detection systems," *Int. J. Sci. Res. Arch.*, vol. 12, no. 01, pp. 2282–2294, 2024, doi: https://doi.org/10.30574/ijsra.2024.12.2.1456.

[14] D. Patel, "Enhancing Banking Security: A Blockchain and Machine LearningBased Fraud Prevention Model," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 06, pp. 1–8, 2023.

[15] R. Q. Majumder, "A Review of Anomaly Identification in Finance Frauds Using Machine Learning Systems," *Available SSRN 5267287*, 2025.

[16] A. J. Rahul Dattangire, Ruchika Vaidya, Divya Biradar, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," *2024 1st Int. Conf. Adv. Comput. Emerg. Technol.*, pp. 1–6, 2024.

[17] J. Kumar Chaudhary, S. Tyagi, H. Prapan Sharma, S. Vaseem Akram, D. R. Sisodia, and D. Kapila, "Machine Learning Model-Based Financial Market Sentiment Prediction and Application," in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, IEEE, May 2023, pp. 1456–1459. doi: 10.1109/ICACITE57410.2023.10183344.

[18] S. Tyagi, T. Jindal, S. H. Krishna, S. M. Hassen, S. K. Shukla, and C. Kaur, "Comparative Analysis of Artificial Intelligence and its Powered Technologies Applications in the Finance Sector," in *Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022*, 2022. doi: 10.1109/IC3I56241.2022.10073077.

[19] L. Hernandez Aros, L. X. Bustamante Molano, F. Gutierrez-Portela, J. J. Moreno Hernandez, and M. S. Rodríguez Barrero, "Financial fraud detection through the application of machine learning techniques: a literature review," *Humanit. Soc. Sci. Commun.*, vol. 11, no. 1, pp. 1–22, 2024, doi: 10.1057/s41599-024-03606-0.

[20] N. Prajapati, "The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications," *ESP J. Eng. Technol. Adv.*, vol. 5, no. 2, pp. 16–22, 2025, doi: 10.56472/25832646/JETA-V5I2P103.

[21] R. Q. Majumder, "Machine Learning for Predictive Analytics: Trends and Future Directions," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 04, pp. 3557–3564, 2025.

[22] B. Chaudhari, S. C. G. Verma, and S. R. Somu, "Transforming Financial Lending: A Scalable Microservices Approach using AI and Spring Boot," *Int. J. Sci. Res. Mod. Technol.*, pp. 72–81, Aug. 2024, doi: 10.38124/ijsrmt.v3i8.527.

[23] M. R. S. and P. K. Vishwakarma, "The Assessments Of Financial Risk Based On Renewable Energy Industry," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 06, no. 09, pp. 758–770, 2024, [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_9_september_2024/61473/final/fin_irjmets1726058754.pdf

[24] G. S. Chaitanya, K. Deepika, G. S. Prabhav, R. B. Patil, and M. A. Jabbar, "Credit Card Fraud Detection using Hidden Naive Bayes and Bayesian Belief Network," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, 2024, pp. 1–6. doi: 10.1109/I2CT61223.2024.10544328.

[25] I. K. Nti and A. R. Somanathan, "A Scalable RF-XGBoost Framework for Financial Fraud Mitigation," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 2, pp. 1556–1563, 2024, doi: 10.1109/TCSS.2022.3209827.

[26] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud Detection in Banking Data by Machine Learning Techniques," *IEEE Access*, 2023, doi: 10.1109/ACCESS.2022.3232287.

[27] A. Arram, M. Ayob, M. A. A. Albadr, A. Sulaiman, and D. Albashish, "Credit card score prediction using machine learning models: A new dataset," 2023.

[28] S. Geetha, Y. Mohammed Khan, R. Sujay, S. P. Yoganand, and R. B, "Fraudulent URL and Credit Card Transaction Detection System Using Machine Learning," in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, 2023, pp. 709–714. doi: 10.1109/ICAECIS58353.2023.10170677.

[29] K. Alsufyani, A. AlMuallim, M. AlShahrani, A. Alsufyani, O. Alhanaya, and A. Zerguine, "Credit Card Fraud Detection via Machine Learning," in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022, pp. 64–67. doi: 10.1109/SSD54932.2022.9955815.

[30] F. Ahmed and R. Shamsuddin, "A Comparative Study of Credit Card Fraud Detection Using the Combination of Machine Learning Techniques with Data Imbalance Solution," in *2021 2nd International Conference on Computing and Data Science (CDS)*, 2021, pp. 112–118. doi: 10.1109/CDS52072.2021.00026.

[31] J. Zhang and I. Mani, "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction," in *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

[32] B. Boddu, "Ensuring Data Integrity and Privacy: A Guide for Database Administrators," *Int. J. Multidiscip. Res.*, vol. 4, no. 6, Nov. 2022, doi: 10.36948/ijfmr.2022.v04i06.10880.

[33] H. Sinha, "Advanced Deep Learning Techniques for Image Classification of Plant Leaf Disease," *J. Emerg. Technol. Innov. Res. www.jetir.org*, vol. 11, no. 9, pp. b107–b113, 2024.

[34] S. Nokhwal, P. Chilakalapudi, P. Donekal, S. Nokhwal, S. Pahune, and A. Chaudhary, "Accelerating Neural Network Training: A Brief Review," *ACM Int. Conf. Proceeding Ser.*, pp. 31–35, 2024, doi: 10.1145/3665065.3665071.

[35] Z. J. Pei, X. Z. Song, H. T. Wang, Y. Q. Shi, S. C. Tian, and G. S. Li, "Interpretation and characterization of rate of penetration intelligent prediction model," *Pet. Sci.*, vol. 21, no. 1, pp. 582–596, 2024, doi: 10.1016/j.petsci.2023.10.011.

[36] R. Tandon, "Face mask detection model based on deep CNN techniques using AWS," *Int. J. Eng. Res. Appl.*, vol. 13, no. 5, pp. 12–19, 2023, doi: 10.9790/9622-13051219.

[37] K. Ullah *et al.*, "Short-Term Load Forecasting: A Comprehensive Review and Simulation Study with CNN-LSTM Hybrids Approach," *IEEE Access*, vol. 12, no. July, pp. 111858–111881, 2024, doi: 10.1109/ACCESS.2024.3440631.

[38] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A Survey of CNN-Based Network Intrusion Detection," 2022. doi: 10.3390/app12168162.

