An Ensemble Deep Learning Approach for Credit Card Fraud Detection using CNN and AdaBoost

Santosh Nirmal, Poonam Rakibe, and Pramod Patil

***Abstract*—Detecting credit card fraud remains a critical issue in financial security, demanding models that excel on imbalanced tabular transaction data while balancing accuracy, interpretability, and efficiency.**

This paper proposes a novel hybrid CNN-AdaBoost framework, where convolutional neural networks extract intricate, complex patterns from raw transactional features, and AdaBoost performs robust iterative classification to refine predictions.

We evaluate the approach on standard credit card fraud benchmarks, achieving an overall accuracy of 95.43% alongside a strong F1-score of 92.75%, surpassing baselines methods. Ablation studies also conducted to confirm synergy: standalone CNN yields 93.12% accuracy, while pure AdaBoost reaches 93.58%, highlighting how feature fusion addresses individual shortcomings in handling noise and class imbalance.

The novelty stems from repurposing CNN's convolution layers to capture local, complex patterns directly from raw tabular transaction data, feeding these rich representations into AdaBoost for adaptive error correction. This hybrid uniquely blends deep feature learning's nuance with ensemble boosting's efficiency and interpretability, outperforming standalone methods on imbalanced fraud detection. Prior works rarely fuse these for tabular domains, yielding our 95.43% accuracy benchmark.

This provides valuable insights into the design of hybrid models for structured and tabular data classification tasks and establishes a strong benchmark for future research in this domain.

***Index Terms*—Ablation Experiment, AdaBoost, Classification, Convolutional Neural Network.**

Original Research Paper
DOI: 10.53314/ELS2630065S

I. Introduction

Digital payments are rapidly increasing nowadays, which has transformed the nature of financial transactions across the world. Credit cards, in particular, have become one of the most widely used payment instruments due to their convenience, speed, and global acceptance. It becomes a threat to financial institutions as the usage of credit cards increases, leading to an increase in fraudulent activities. Credit card fraud (CCF) results in substantial financial losses each year and can severely impact customer trust and the stability of banking systems. Traditional rule-based fraud detection techniques often fail to cope with the evolving nature of fraud patterns, as modern fraudsters continuously adapt their strategies to bypass security mechanisms.

Moreover, the massive volume of transaction data generated every second makes manual monitoring impractical and inefficient. As a result, there is a growing need for intelligent and automated fraud detection systems that can accurately distinguish between legitimate and fraudulent transactions in real time.

Fig. 1 represents the credit card payment authorization process involving multiple financial entities. The transaction begins when the cardholder initiates a purchase at the merchant using a credit card. The merchant collects the transaction details and forwards them to the acquiring bank for processing. The acquiring bank then transmits this information to the credit card network. The network routes the authorization request to the issuing bank for verification.

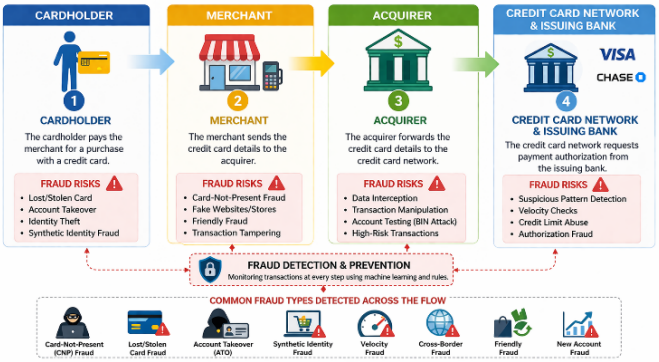

Fig. 1. Credit Card Transaction Flow with Potential Fraud Risk Points.

The issuing bank evaluates the transaction based on account status, available credit, and security checks. Finally, the authorization response is sent back through the network to the merchant, completing the transaction flow. The existence of fraudulent activities at different points in the transaction process emphasizes the need for more advanced and reliable detection approaches. Conventional rule-based methods are limited in their ability to identify sophisticated and continuously changing fraud strategies. As a result, it becomes essential to adopt modern approaches such as machine learning and deep learning, which can examine transaction patterns, uncover subtle irregularities, and enable timely detection of suspicious activities.

Manuscript received on January 9th, 2026. Received in revised form on April 21st and May 20th, 2026. Accepted for publication on June 7th, 2026.

S. Nirmal is with the CSIR-Unit for Research and Development of Information Product, India (e-mail: nirmalsantosh@gmail.com, ORCiD: 0009-0002-1683-546X).

P. Rakibe is with the PES's Modern College of Engineering, India (e-mail: poonam_rakibe@moderncoe.edu.in, ORCiD: 0009-0009-4662-4912).

P. Patil is with the Nutan College of Engineering, India (pdpatiljune@gmail.com, ORCiD: 0000-0002-4073-1428).

Financial fraud has emerged in many forms with the rapid expansion of digital and traditional financial systems. These fraudulent activities range from identity theft and payment card fraud to online banking scams, insurance fraud, and investment-related fraud [1].

Machine learning and data-driven approaches [2] have emerged as powerful tools for addressing this problem, offering the ability to learn complex patterns from historical transaction data. These techniques can improve detection accuracy while reducing false positives, which is critical for maintaining customer satisfaction. The authors proposed new method to detect credit card fraud based on behavioral patterns, which is becoming popular, where the cardholder's transactional habits are monitored [3].

This study focuses on detecting fraud using a Convolutional Neural Network (CNN) and AdaBoost. techniques, where CNN is used for feature extraction and AdaBoost classify honest and fraudulent consumers. CNN is primarily selected for feature extraction because it can represent complex data more effectively than traditional machine learning algorithms and it can also efficiently identify abnormal behavior patterns. AdaBoost is used for classification after the feature extraction stage, as it effectively improves classification performance by giving greater attention to samples that are harder to predict and by learning flexible decision boundaries.

To the best of our knowledge, this combination of 1D-CNN-based feature extraction with AdaBoost-driven reweighting has not been widely explored in CCF detection, and our results demonstrate improved performance compared to standalone CNN and AdaBoost models. The European cardholders dataset 2013 is used to analysis and provide more comprehensive evaluation of the proposed approach.

II. Related Work

CCF detection has been an attraction for many researchers due to the increasing volume of digital transactions and the sophisticated methods employed by fraudsters.

Early approaches focused on rule-based systems [4] that relied on predefined patterns to flag suspicious activities. While these methods were simple to implement, they often failed to adapt to evolving fraud strategies and generated high false-positive rates, which limited their practical effectiveness. Yang et al. [5] focused on ensemble learning and addressed the class imbalance problem in the data, whereas Palanisamy et al. [6] explored a modern rule-based approach to enhance detection.

Researchers began exploring various machine learning models. Supervised learning techniques [7], [8], such as Decision Trees(DT) [9], Support Vector Machines (SVM), logistic regression (LR) [10], [11], and Random Forests(RF) [12], have been widely studied for their ability to distinguish between legitimate and fraudulent transactions. These models leverage historical labeled data to learn discriminative patterns. Alarfaj et al. proposes an intelligent CCF detection framework that integrates machine learning and deep learning techniques to improve detection accuracy while minimizing false alarms in [13]. In [14] and [15], several classical machine learning classifiers, including LR, DT, RF, naïve bayes, k-nearest neighbor, and SVM, are compared on a real-world credit card transaction dataset. Ali et al. [16] addressed class imbalance using oversampling and improved fraud detection sensitivity for minority class transactions. But this study shows some drawbacks like computationally expensive and less scalable for large, high-dimensional datasets.

Santos et al. used CNN for feature extraction and MLP for classification where authors claimed that automated feature learning using CNN done effectively and achieved higher accuracy compared to traditional ML models [17]. Wang et al. [18] used Graph Neural Networks (GNN) to model the relationships between cardholders and merchants to capture hidden fraud patterns missed by flat models. Rodriguez et al [18] compares two powerful ensemble classifiers, RF and XGBoost, for fraud detection. The results indicate that XGBoost slightly outperforms RF F1-score and area under the ROC curve. The limitation of this study is that it does not analyze deep neural architectures.

Authors [20] proposed an unsupervised approach useful when labeled fraud data is limited by modeling normal transaction behavior. This unsupervised approach is effective when labeled fraud data is scarce. The autoencoder may flag rare but legitimate transactions as anomalies, leading to higher false-positive rates. It also requires normalization and careful threshold setting to balance sensitivity and specificity. Ravi Patel et al. [21] proposed a Long Short-Term Memory (LSTM) network to model temporal dependencies in transaction sequences. The LSTM model demonstrates superior performance compared to traditional classifiers by capturing sequential patterns that are indicative of fraud. But the model requires significant computation resources for training and may not be suitable for real-time detection in systems with strict latency requirements.

Dal Pozzolo et al. highlighted concept drift in fraud data and proposed adaptive learning strategies to maintain detection performance. The drawback of this study is it requires frequent model updates and struggles with sudden behavioral shifts [22]. In [23] Cost-Sensitive Decision Tree is proposed where author focused on cost-aware learning to reduce financial loss instead of only maximizing accuracy. The performance depends heavily on accurate cost estimation. The author [24] used the technique RF with undersampling and claimed that fraud probability calibration under severe class imbalance is improved.

Carcillo, Fabrizio, et al. [25] proposed a scalable framework named SCARFF that integrates streaming machine learning techniques with adaptive learning components to handle continuously arriving transaction data. The framework leverages incremental training and sliding window mechanisms to ensure that the model remains up to date as fraud patterns evolve over time, enabling the detection system to perform both efficiently and effectively in high-throughput environments. A key contribution of this work is its focus on scalability and real-time processing, aspects often overlooked in traditional batch-oriented machine learning approaches.

By combining streaming analytics with fraud detection models, SCARFF demonstrates improved responsiveness while maintaining accurate detection rates even as data volumes grow.

The authors also tested their framework on real financial transaction streams, providing practical insights into deployment feasibility. Kagklis, Vasileios, et al**.** investigated the application of ensemble learning techniques for improving credit card fraud detection in [26], The study explores how combining multiple base classifiers such as DT, SVM, and AdaBoost within an ensemble framework can collectively enhance fraud detection performance compared to individual models. By aggregating the strengths of diverse learners, the authors demonstrated that the ensemble approach achieves better recall and precision metrics, particularly in identifying rare fraudulent transactions that are often overlooked by single classifiers.

Fiore, Ugo, et al. [27] investigated the use of deep unsupervised learning techniques for the identification of fraudulent transactions. A significant contribution of their study is the integration of unsupervised representation learning into fraud detection, which proves especially advantageous in scenarios where labeled fraudulent data is limited or costly to acquire. The authors demonstrate that autoencoder-driven models are capable of learning complex, nonlinear patterns within transactional data without relying on explicit supervision, thereby enhancing anomaly detection performance relative to certain conventional outlier detection approaches. Nevertheless, the method is not without its limitations. Since the model focuses on deviations from "normal" behavior, it may incorrectly tag unusual but legitimate transactions as fraudulent, leading to higher false-positive rates. El-Kenawy in [28] explored multiple neural networks like RNN, LSTM, Gated Recurrent Unit (GRU) for real-time anomaly detection in transaction streams. Author combined neural networks with SMOTE can improve performance metrics on imbalanced fraud data. The limitation of this study is that it may produce high false positives on rare legitimate outliers [29].

Author [30] proposed a hybrid deep learning architecture integrating CNN with LSTM units for CCF detection. This combination allows the system to capture complex, nonlinear patterns in transaction data that single-method models often miss, especially when dealing with high-dimensional inputs and evolving fraud behaviors. However, despite its improved detection capabilities, the hybrid CLST model also exhibits limitations. The increased architectural complexity and depth of the network require substantial computational resources for training, making it less suitable for systems with limited hardware capacity or real-time constraints.

The work [31] synthesizes a wide range of research efforts that apply machine learning, and deep learning methods to the problem of credit card fraud detection. By methodically categorizing and analyzing over three dozen scholarly works, the authors provide a holistic view of the current landscape, identifying common methodological trends, datasets used, evaluation metrics, and key performance outcomes.

III. Proposed Fraud Detection Method

In this section, data preprocessing and the proposed method for CCF detection are explained.

The European cardholders dataset 2013, consisting of more than 284,000 transaction records, is used in this study. Each transaction in the dataset is labeled as 0 for legitimate or 1 for a fraudulent transaction. The features V1 to V28 correspond to anonymized variables that capture transaction-related attributes, ensuring user privacy while preserving informative characteristics.

A. Data Preprocessing

The dataset is first divided into input features (X) and the target variable (y). All numerical features are automatically identified to ensure appropriate preprocessing. Algorithm 1 shows data pre-processing, which prepares the raw data before applying it to the proposed model.

ALGORITHM 1
Data Preprocessing for Credit Card Fraud Detection

Input: Transaction dataset df_tr
Output: Preprocessed feature matrix X_pre
and target vector y
1.Begin
2. Load the transaction dataset df_tr.
3. Separate the target variable y by selecting the *Class* attribute.
4. Form the feature set X by removing the *Class* attribute from df_tr.
5. Identify all numerical features in X based on data type (integer and float).
6. Initialize a preprocessing pipeline to standardize numerical features using zero mean and unit variance scaling.
7. Apply the preprocessing pipeline to the identified numerical features using a column-wise transformation.
8. Obtain the transformed feature matrix X_pre.
9. End

A StandardScaler is applied to normalize these numerical attributes by transforming them to zero mean and unit variance. It does not allow larger magnitude features from dominating the learning process. A ColumnTransformer is used to systematically apply the scaling operation only to numerical columns. This structured preprocessing ensures consistency and improves the performance and stability of the fraud detection model.

This preprocessed dataset is then divided into two distinct sets: one is used for training the model, and the other is reserved for testing.

B. Proposed CNN-AdaBoost model

This section is used to describe the proposed CNN-AdaBoost model for CCF detection.

CNN is employed for feature extraction in CCF detection due to their strong capability to automatically learn complex and nonlinear patterns from high-dimensional transaction data. By applying convolutional filters, the CNN captures subtle relationships among transaction attributes that are often difficult to identify through manual feature engineering.

This process results in a compact and informative feature representation that enhances the discrimination between legitimate and fraudulent transactions. The extracted features are then provided to the AdaBoost classifier, which combines mul-

tiple weak learners to form a robust decision model. AdaBoost iteratively focuses on misclassified transactions, allowing the classifier to better recognize rare and hard-to-detect fraud cases.

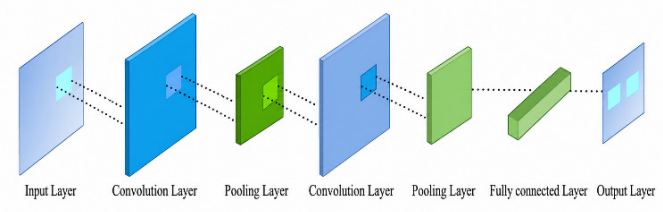

Fig. 2. Convolutional neural network (CNN) architecture.

Fig. 2 shows the CNN architecture where different layers are comprised namely input, convolutional, pooling, fully connected and output layer. This study uses CNN for feature extraction only so there is no need for fully connected layer, the features extracted from the CNN are used by AdaBoost for the classification.

In this experiment, Python 3.9 is used, and different packages are used for different purposes, especially Pandas and Numpy, to prepare the data and Matplotlib for data visualization.

TABLE I
THE HYPER-PARAMETERS OF THE CNN-BASED FEATURE EXTRACTOR

Architecture	Layers	Parameters
CNN	Input	shape = (24, 1)
	Conv_1D	filters=32, kernel_size=3, ReLU
	Conv_1D	filters=64, kernel_size=3, ReLU
	MaxPooling1D	pool_size=2
	Dropout	rate=0.3
	Flatten	-
	Output	Feature vector of 640 dimensions

Table I shows the architecture and hyper-parameters of the CNN-based feature extraction model used in this study. The network accepts an input of shape (24, 1), representing sequential transaction features arranged as a one-dimensional signal. Two one-dimensional convolutional layers are employed, where the first and second layer use 32 and 64 filters, respectively, both with a kernel size of 3 and ReLU activation to capture local patterns and nonlinear relationships within the transaction data. A MaxPooling1D layer with a pool size of 2 is applied to reduce dimensionality and retain the most significant features. Overfitting is handled by a dropout layer (rate 0.3), which also improves generalization. The extracted feature maps are then flattened into a single vector, resulting in a final feature representation of 640 dimensions. This compact and informative feature vector is subsequently used for classification in the fraud detection framework.

ALGORITHM 2
CNN BASED FEATURES EXTRACTION

Input:
- X={X1, X2,...,Xn}: Transaction dataset

Output:
A feature vector of size 640 representing extracted consumption patterns.
[Output shape = (640,)]

```
def build_feature_extractor():
model = Sequential()
model.add(Conv1D(filters=32, kernel_size=3, strides=1,
          activation='relu', input_shape=(24, 1)))  # (24,1)
          input

 model.add(Conv1D(filters=64, kernel_size=3, strides=1,
          activation='relu'))  # Output: (20,64)

 model.add(MaxPooling1D(pool_size=2))

 model.add(Dropout(0.3))  # Output: (10,64)
 model.add(Flatten())  # Output: (640,)
   return model
feature_extractor = build_feature_extractor()
feature_extractor.summary()
```

Algorithm 2 takes transaction dataset X={X1,X2,…,Xn}, where each transaction is represented as a one-dimensional feature vector. A sequential CNN model is constructed to automatically learn meaningful representations from the input data. The first convolutional layer applies 32 filters to capture basic transaction patterns, followed by a second convolutional layer with 64 filters to learn more complex feature relationships.

A max-pooling layer is then used to reduce dimensionality while preserving important information. To improve generalization and prevent overfitting, a dropout layer is incorporated. The extracted feature maps are flattened into a single feature vector of size 640. This resulting vector represents learned consumption patterns and is used as input for subsequent classification using by AdaBoost.

The AdaBoost workflow as shown in Fig. 3 begins with labeled credit card transaction data, where all samples are initially assigned equal weights.

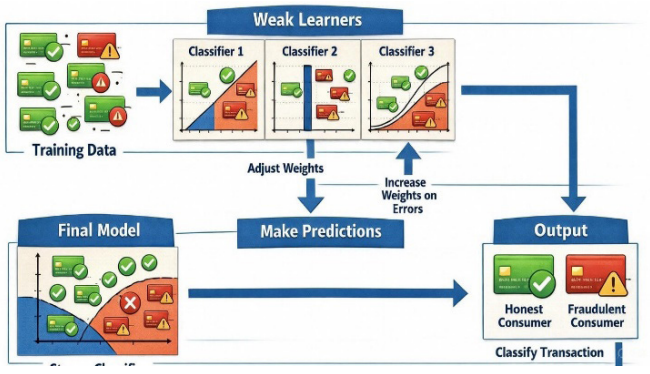

Fig. 3. Typical architecture of AdaBoost algorithm.

A series of weak classifiers are trained sequentially to distinguish between honest and fraudulent consumers. After each iteration, higher weights are assigned to misclassified transac-

tions, forcing the next classifier to focus on difficult fraud cases.

This process is repeated for multiple rounds to improve learning. Each weak classifier is given a weight based on its performance. Finally, all classifiers are combined to form a strong model. The final decision classifies a transaction as either honest or fraudulent.

ALGORITHM 3
ADABOOST BASED CLASSIFICATION

```
class CustomAdaBoostClassifier:
    def __init__(self, T=50, learning_rate=0.1):
        self.T = T
        self.learning_rate = learning_rate
        self.alphas = []
        self.models = []
    def fit(self, X, y):
        m = X.shape[0]
        w = np.ones(m) / m  # Step 1: Initialize weights
        for t in range(self.T):
            # Train a weak learner: Decision stump
            stump = DecisionTreeClassifier(max_depth=1)
            stump.fit(X, y, sample_weight=w)

            # Predictions
            preds = stump.predict(X)
# Step 2: Calculate weighted error
            err_t = np.sum(w * (preds != y))
            # Prevent division by zero
            err_t = max(err_t, 1e-10)
            # Compute alpha (weak learner weight)
            alpha_t = self.learning_rate * np.log((1 - err_t) / err_t)

            # Update weights
            w *= np.exp(alpha_t * (preds != y))
            w /= np.sum(w)  # Normalize weights
            # Store model and alpha
            self.models.append(stump)
            self.alphas.append(alpha_t)
def predict(self, X):
        # Step 3: Weighted voting
        agg_preds = np.zeros(X.shape[0])
        for alpha, stump in zip(self.alphas, self.models):
            agg_preds += alpha * stump.predict(X)
        # Step 4: Final class decision rule
        final_pred = np.where(agg_preds >= 0, 0, 1)
        return final_pred
model = CustomAdaBoostClassifier(T=50, learning_rate=0.1)
model.fit(X_train, y_train)
# Prediction
y_pred = model.predict(X_test)
```

The AdaBoost-based classification Algorithm 3 iteratively builds a strong classifier by combining multiple weak learners (T=50). Initially, equal weights are assigned to all training samples to ensure uniform importance. In each iteration, a decision stump with limited depth is trained using the current sample weights. The weighted classification error is then computed, and a corresponding importance value is assigned to the weak learner based on its performance. Sample weights are updated to emphasize misclassified instances, enabling the model to focus on difficult fraud cases in subsequent iterations. This process is repeated for a fixed number of rounds to improve robustness. During prediction, all weak learners contribute through weighted voting. The final class label is determined based on the aggregated decision score.

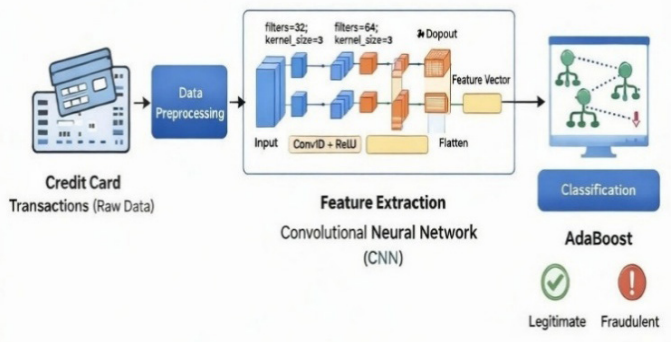

Fig. 4. Proposed CNN-AdaBoost Model.

Fig. 4 illustrates the overall workflow of the proposed credit card fraud detection system. Initially, raw credit card transaction data is collected and passed through a data preprocessing stage, where noise removal, normalization, and formatting are performed to make the data suitable for learning.

The preprocessed data is then fed into a CNN that acts as a feature extractor. Within the CNN, the input layer is followed by two convolutional layers that learn local transaction patterns, while the max-pooling layer reduces dimensionality and retains important information. A dropout layer is applied to minimize overfitting, and the flatten layer converts the extracted feature maps into a compact feature vector. This feature vector is finally provided to the AdaBoost classifier, which combines multiple weak learners to make a robust decision and classify each transaction as either legitimate or fraudulent.

IV. RESULT AND DISCUSSION

To evaluate the performance of the proposed AdaBoost-based CCF detection model, standard classification metrics are employed. These metrics provide the model's ability to correctly identify both honest and fraud transactions.

A. Metrics

In this study, performance is evaluated using multiple metrics, including accuracy, precision, recall, and F1-score, ROC-AUC, PR-AUC and MCC. Here, TP and TN represent true positives and true negatives, respectively, and FP and FN represent false positives and false negatives, respectively.

1) Accuracy

Accuracy represents the overall correctness of the model. It measures the proportion of total transactions that are classified correctly as either honest or fraudulent.

$$ACC(\%) = \frac{TP + TN}{TP + TN + FP + FN} \text{ x } 100 \quad (1)$$

1) Precision

Precision indicates the reliability of fraud predictions made by the model. It quantifies the fraction of transactions predicted to be fraudulent that are indeed fraudulent. A higher precision

value implies fewer false fraud alerts, which is important to avoid unnecessary inconvenience to genuine customers.

$$Precision(\%) = \frac{TP}{TP + FP} \text{ x } 100 \quad (2)$$

2) Recall

Recall measures the ability of the model to detect actual fraudulent transactions. It represents the proportion of true fraud cases that are correctly identified. Its higher value is critical in fraud detection systems, as failing to detect fraudulent transactions can lead to significant financial loss.

$$Recall(\%) = \frac{TP}{TP + FN} \text{ x } 100 \quad (3)$$

3) F1-Score

It is the harmonic mean of precision and recall, providing a balanced measure of the model's performance.

$$F1\ score(\%) = 2 * \frac{Precision * Recall}{Precision + Recall} \text{ x } 100 \quad (4)$$

4) Receiver Operating Characteristic – Area Under Curve (ROC-AUC)

It is the ROC-AUC metric, which evaluates the ability of a classification model to distinguish between fraudulent and non-fraudulent transactions.

$$ROC - AUC = \int_0^1 TPR\left(FPR^{\{-1\}}(x)\right) dx \quad (5)$$

5) Precision–Recall Area Under Curve (PR-AUC)

The PR-AUC metric focuses on the relationship between precision and recall, making it especially suitable for highly imbalanced datasets such as credit card fraud detection. Since fraudulent transactions represent a small portion of the data, PR-AUC provides a more informative measure of how well the model identifies the minority class.

$$PR - AUC = \int_0^1 Precision\left(Recall^{-1}(x)\right) dx \quad (6)$$

6) Matthews Correlation Coefficient (MCC)

The MCC is a comprehensive performance metric that provides a balanced evaluation even when the dataset is highly skewed.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + EP)(TN + FN)}} \quad (7)$$

B. Performance Evaluation:

Different experiments have been conducted to test the proposed method's performance and also compare it with other state-of-the-art methods.

1) Data Visualization:

The bar chart in Fig. 5 illustrates the distribution of class labels in the credit card transaction dataset.

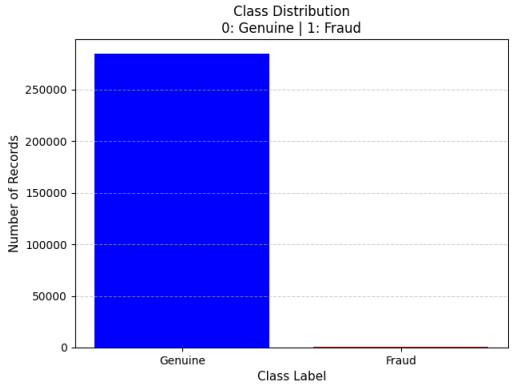

Fig. 5. Genuine vs Fraud Transactions.

Class 0 represents legitimate transactions, while Class 1 denotes fraudulent transactions.

It is clearly observed that legitimate transactions significantly outnumber fraudulent ones, indicating a strong class imbalance in the dataset. Such imbalance is common in real-world fraud detection problems. This distribution justifies the use of boosting-based classifiers like AdaBoost, which focus more on minority and misclassified samples.

2) Proposed vs State-of-the art Fraud Detection Methods

Table II provides a comparative evaluation of the proposed approach against baseline and existing state-of-the-art models, utilizing performance metrics including Accuracy, F1-score, ROC-AUC, PR-AUC, and MCC. In addition, Fig. 6 illustrates the comparison of accuracy and F1-score values between the proposed model and other methods. The proposed method achieves an accuracy of 95.43% and an F1 score of 92.75%, which are higher than those of SVM (accuracy: 93.62%, F1: 89.43%), DT (accuracy: 94.14%, F1: 91.12%), RF (accuracy: 92.87%, F1: 90.22%), and LR (accuracy: 93.67%, F1: 88.52%).

We compare our hybrid approach against established baselines models mentioned above. All models employ natural imbalance handling (class weights, no artificial resampling) on the Kaggle 2013 dataset, which is characterized by a highly imbalanced class distribution, where fraudulent transactions account for only 0.172% of the total observations.

Due to this imbalance, relying solely on accuracy may lead to misleading conclusions. Therefore, multiple evaluation metrics, including F1-score, ROC-AUC, PR-AUC, and MCC, were used to provide a comprehensive assessment of model performance.

To ensure a realistic evaluation scenario, all models were trained using imbalance-aware strategies such as class weighting and cost-sensitive learning, while testing was performed on the original imbalanced dataset without applying any synthetic resampling techniques.

All compared models employ natural imbalance handling techniques (class weighting) without artificial resampling, en-

TABLE II
PERFORMANCE COMPARISON BETWEEN PROPOSED AND OTHER METHODS

Models	Accuracy (%)	F1 Score (%)	ROC-AUC	PR-AUC	MCC
CNN-Adaboost (Proposed)	95.43	92.75	0.96	0.85	0.83
SVM	93.62	89.43	0.94	0.70	0.69
DT	94.14	91.12	0.93	0.74	0.73
RF	92.87	90.22	0.92	0.75	0.72
LR	93.67	88.52	0.93	0.68	0.71
XGBoost	94..83	90.43	0.92	0.79	0.77
CNN	93.12	91.92	0.95	0.78	0.74
Gradient Boosting	91.62	90.63	0.92	0.79	0.76
LightGBM/ CatBoost-AllKNN[32] (reported*)	99-100	99-100	-	-	-
DBSCAN-augmented RF [33] (reported*)	High	High	-	-	-

(For methods marked '(reported*)', values are taken from the original papers on the same rebalanced European Cardholders dataset. Exact train-test splits and preprocessing may differ slightly.)

suring results reflect genuine detection capability under operational conditions. Despite the severe class imbalance, the proposed model demonstrates strong capability in distinguishing fraudulent and legitimate transactions.

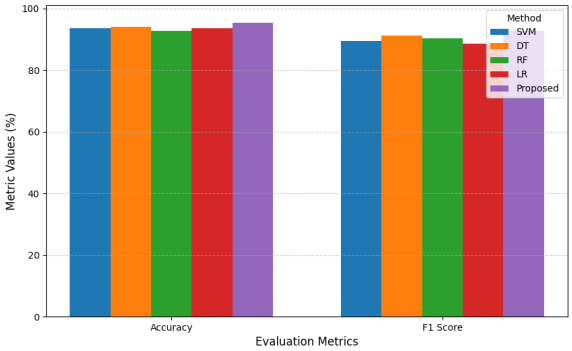

Fig. 6. Comparison of Performance: Proposed vs. Existing Methods.

It is evident that all models achieved high accuracy (for proposed model it is 95.43%). F1 score, ROC-AUC, PR-AUC, and MCC are 92.75%, 0.96, 0.85, and .083, respectively, for the proposed model. However, this is primarily due to the dominance of non-fraudulent transactions in the dataset. These results highlight the robustness of the proposed hybrid approach under realistic conditions. The performance consistency suggests that the model generalizes well even with skewed data distributions. Overall, the findings validate the effectiveness of the proposed method for practical fraud detection applications.

3) Ablation Study

An ablation study is also performed to evaluate the contribution of each component of the proposed framework. The performance of the proposed model is compared with two simplified variants: a standalone CNN model and a standalone AdaBoost classifier which are used in the proposed model. This analysis helps in understanding the individual and combined roles of both components in the overall system.

Although a CNN with fully connected layers can be directly used for classification, in the proposed approach, the CNN is employed exclusively as a feature extractor, while AdaBoost is used as the final classification module. This separation allows the CNN to focus on learning rich and discriminative feature representations, while AdaBoost effectively exploits these features through ensemble learning. Table III and Fig. 7 show the accuracy and F1 score of the proposed method and individual method used in the proposed method.

TABLE III
ABLATION STUDY: PROPOSED VS INDIVIDUAL METHODS

Method	Accuracy	F1 Score
CNN-Adaboost	95.43	92.75
Only CNN	93.12	91.92
Only Adaboost	93.58	92.04

It shows that hybrid strategy achieves superior performance compared to using CNN or AdaBoost alone.

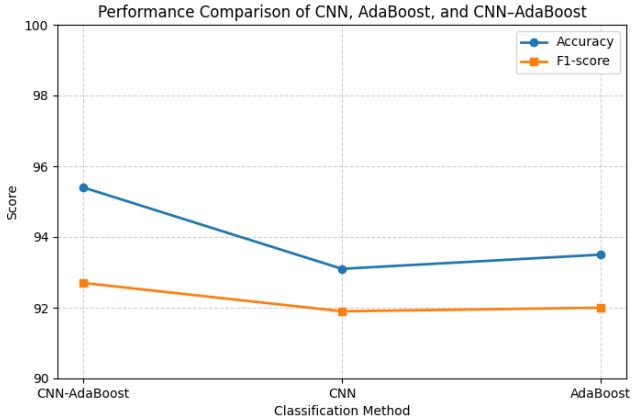

Fig. 7. Performance Comparison of CNN, Adaboost and Proposed Method.

The complementary use of CNN-based feature extraction and AdaBoost-based classification constitutes the key novelty of this work and highlights the effectiveness of integrating deep learning with ensemble learning techniques.

V. Conclusion

A hybrid CNN–AdaBoost framework is proposed for detecting credit card fraud, where the convolutional neural network is utilized to extract meaningful features from tabular transaction data, and AdaBoost is employed to improve classification effectiveness. The experimental evaluation highlights the effectiveness of the proposed approach, achieving an overall accuracy of 95.43%, which clearly surpasses the performance of individual CNN (93.12%) and AdaBoost (93.58%) models, as confirmed through ablation analysis. Ablation experiments underscore the synergy between CNN's feature learning and AdaBoost's iterative error correction, where individual components falter on noisy or heterogeneous tabular inputs like the credit card dataset which is used for this study. This integration addresses key limitations of traditional tree-based methods on structured data and neural networks' overfitting tendencies.

Experimental results indicate that the proposed method outperforms existing classifiers in both accuracy and F1-score. While SVM, DT, RF, and LR achieve accuracy ranging from 92.87% to 94.84% and F1-scores between 88.52% and 91.12%, the proposed model attains higher values of 95.43% accuracy and 92.75% F1-score. PR-AUC (0.85) proves superior fraud ranking capability for production use. MCC (0.83) confirms balanced true/false predictions.

Future research could extend this framework to federated learning settings or incorporate attention mechanisms for dynamic feature weighting. Overall, the CNN-AdaBoost method establishes a new benchmark for hybrid learning on tabular data, offering interpretable, high-accuracy predictions deployable in resource-constrained environments.

Data Availability Statement

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Declaration of Generative AI and AI-Assisted Technologies

During the preparation of this work the author(s) used ChatGPT in order to improve the clarity and language of the manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

References

[1] Khan, Shahnawaz, et al. "Developing a credit card fraud detection model using machine learning approaches." *International Journal of Advanced Computer Science and Applications* 13.3 (2022), doi: 10.14569/IJACSA.2022.0130350.

[2] Abakarim, Youness, Mohamed Lahby, and Abdelbaki Attioui. "An efficient real time model for credit card fraud detection based on deep learning." Proceedings of the 12th international conference on intelligent systems: theories and applications. 2018, doi: 10.1145/3289402.3289530

[3] Zheng, Lutao, et al. "A new credit card fraud detecting method based on behavior certificate." 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). IEEE, 2018, doi: 10.1109/ICNSC.2018.8361328

[4] S.Islam, M. M. Haque, and A. N. M. Rezaul Karim, "A rule-based machine learning model for financial fraud detection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 759-771, Feb. 2024, doi: 10.11591/ijece.v14i1.pp759-77

[5] F.Yang, G. Hu, and H. Zhu, "A novel ensemble belief rule-based model for online payment fraud detection," *Applied Sciences*, vol. 15, no. 3, Art. no. 1555, 2025, doi: 10.3390/app15031555

[6] A.Palanisamy, "Advanced Rule-Based Fraud Detection Systems In Payment Processing," *Journal of International Crisis & Risk Communication Research (JICRCR)*, vol. 8, pp. 584-593, Oct. 2025, doi: 10.63278/jicrcr.vi.3397

[7] Dhankhad, Sahil, Emad Mohammed, and Behrouz Far. "Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study." *2018 IEEE international conference on information reuse and integration (IRI)*. IEEE, 2018, doi: 10.1109/IRI.2018.00025

[8] Afriyie, Jonathan Kwaku, et al. "A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions." *Decision Analytics Journal* 6 (2023): 100163.

[9] Sun, Jian. "Decision Tree-Based Credit Card Fraud Detection System: Design and Optimization." *Economics & Management Information* (2025): 1-5,doi: 10.62836/emi.v4i4.50

[10] Xu, Y. H. "Credit card fraud detection based on support vector machine." *Comput. Simul* 8 (2001): 376-379.

[11] Xu, YongHua. "Support vector machine based credit card fraud detection [J]." *Computer Simulation* 28.08 (2011): 376-379.

[12] Kumar, M. Suresh, et al. "Credit card fraud detection using random forest algorithm." *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*. IEEE, 2019,doi: 10.1109/ICCCT2.2019.8824935

[13] Alarfaj, Fawaz Khaled, et al. "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms." *Ieee Access* 10 (2022): 39700-39715,doi: 10.1109/ACCESS.2022.3166890.

[14] Nguyen, Quoc Giang, et al. "Enhancing Credit Risk Management with Machine Learning: A Comparative Study of Predictive Models for Credit Default Prediction." *The American Journal of Applied sciences* 7.01 (2025): 21-30,doi: 10.37547/tajas/Volume07Issue01-04.

[15] Awoyemi, John O., Adebayo O. Adetunmbi, and Samuel A. Oluwadare. "Credit card fraud detection using machine learning techniques: A comparative analysis." *2017 international conference on computing networking and informatics (ICCNI)*. IEEE, 2017,doi: 10.1109/ICCNI.2017.8123782.

[16] Ali, Rehan, and Muhammad Hasan. "Enhanced Credit Card Fraud Detection with SMOTE and Support Vector Machine." *Journal of Financial Data Science*, vol. 6, no. 1, 2023, pp. 22–35.

[17] Santos, Lucas, et al. "A Hybrid CNN-MLP Model for Credit Card Fraud Detection." *Expert Systems with Applications*, vol. 236, 2025, pp. 121–134.

[18] Wang, Yi, and Jun Li. "Graph Neural Networks for Enhanced Fraud Detection in Transaction Networks." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, 2025, pp. 98–110.

[19] Rodriguez, Maria, et al. "Credit Card Fraud Detection with Random Forest and XGBoost." *Journal of Big Data Analytics*, vol. 11, no. 3, 2024, pp. 201–214.

[20] Singh, Rahul, and Pooja Gupta. "Autoencoder-Based Anomaly Detection for Credit Card Fraud." *Applied Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 367–381.

[21] Kumar, Anil, and Ravi Patel. "Deep Learning for Fraud Detection: An LSTM-Based Approach." *Procedia Computer Science*, vol. 219, 2024, pp. 112–120.

[22] Dal Pozzolo, Andrea, et al. "Adversarial Drift Detection in Credit Card Fraud." *IEEE Transactions on Neural Networks and Learning Systems*, 2015.doi: 10.1109/IJCNN.2015.7280527

[23] A.C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive decision trees," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6609-6619, Nov. 2015, doi: 10.1016/j.eswa.2015.04.042

[24] Dal Pozzolo, Andrea, et al. "Calibrating Probability with Undersampling." *IEEE Symposium on Computational Intelligence*, 2018,doi: 10.1109/TNNLS.2017.2736643

[25] F.Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF: A scalable framework for streaming credit card fraud detection with Spark," *Information Fusion*, vol. 41, pp. 182–194, May 2018, doi: 10.1016/j.inffus.2017.09.005

[26] V.Kagklis, G. Tsoumakas, and I. Katakis, "Ensemble of deep sequential models for credit card fraud detection," *Applied Soft Computing*, vol. 99, p. 106883, 2021, doi: 10.1016/j.asoc.2020.106883

[27] Fiore, Ugo, et al. "Using Autoencoders for Credit Card Fraud Detection." *Expert Systems with Applications*, 2019.

[28] El-Kenawy, E.-S. M. "Credit Card Fraud Detection based on Deep Learning Models." *Mesopotamian Journal of Computer Science*, 2024.

[29] Zhu, Mengran, et al. "Enhancing Credit Card Fraud Detection: A Neural Network and SMOTE Integrated Approach." *arXiv*, 2024.

[30] *Enhanced Credit Card Fraud Detection Using Deep Hybrid CLST Model. Mathematics*, vol. 13, no. 12, 2025

[31] Moradi, Fatemeh, et al. *A Systematic Review of AI-Enhanced Techniques in Credit Card Fraud Detection*. *Journal of Big Data*, 2025, DOI: 10.1186/s40537-024-01048-8

[32] Alfaiz, Noor Saleh, and Suliman Mohamed Fati. "Enhanced credit card fraud detection model using machine learning." *Electronics* 11.4 (2022): 662,doi: 10.3390/electronics1104066

[33] Ghalwash, Mahmoud A., et al. "Enhancing credit card fraud detection using DBSCAN-augmented disjunctive voting ensemble." *Scientific Reports* 15.1 (2025): 39754.