ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 3 Issue 4 December 2023 / Page No: 149-157
Paper Id: JETA-V3I8P117 / Doi: 10.56472/25832646/JETA-V3I8P117

Original Article

Deep Learning-Based Fraud Detection in Financial Transactions: A Case Study Using Real-Time Data Streams

Varun Verma
Independent Researcher

Received Date: 28 November 2023 Revised Date: 11 December 2023 Accepted Date: 25 December 2023

Abstract: *Credit card fraud continues to be a major challenge for financial institutions, especially with the increasing volume and speed of online transactions. Traditional rule-based fraud detection systems are often unable to adapt to the evolving patterns of fraudulent behavior. In this paper, we present a deep learning-based framework for credit card fraud detection using artificial neural networks (ANN) and advanced convolution neural network (CNN) architectures such as VGG16 and VGG19. The system processes credit card transactions and applies deep learning models to classify them as fraudulent or legitimate. ANN is used as a baseline model due to its simplicity and speed, while VGG16 and VGG19 are fine-tuned to learn complex features and detect subtle anomalies in transaction patterns. To address the inherent class imbalance in fraud datasets, Synthetic Minority Over-Sampling Technique (SMOTE) is applied to generate synthetic fraudulent samples, along with random under-sampling of the majority class. Principal Component Analysis (PCA) is employed to reduce data dimensionality, improving model efficiency and reducing computation time. Experimental results on a benchmark credit card dataset show that VGG16 and VGG19 outperform the ANN model in terms of accuracy, precision, and recall, with VGG19 achieving the highest performance. This study demonstrates the potential of deep CNN architectures combined with stream processing to build scalable and effective solutions for credit card fraud detection in dynamic financial environments.*

Keywords: *Credit Card Fraud Detection, Machine Learning, Lightgbm, Gradient Boosting, Real-Time Detection, Financial Fraud, Feature Engineering, Model Evaluation.*

I. INTRODUCTION

The increasing digitalization of financial services has significantly improved transaction efficiency and accessibility but has also led to a surge in fraudulent activities. With the growth of online banking, e-commerce, and mobile payments, financial institutions are under constant threat from sophisticated fraud schemes that exploit vulnerabilities in transaction systems. Traditional fraud detection methods, primarily based on rule-based systems and manual audits, are often inadequate in identifying evolving patterns of fraud in real time. In response to these challenges, deep learning has emerged as a powerful tool for detecting fraudulent behaviour in financial transactions. [1-2 Deep learning models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders, can learn complex, nonlinear relationships from vast amounts of data. These models are particularly suited for real-time data stream analysis, enabling financial systems to flag suspicious transactions instantaneously, thus reducing financial losses and enhancing customer trust.[3]

Through this case study, we aim to provide insights into the practical implementation of deep learning models in dynamic financial environments and highlight the advantages of intelligent, automated fraud detection systems in combating financial crime. The rapid growth of digital financial services has transformed the way consumers conduct transactions, with credit cards becoming one of the most widely used payment methods worldwide. While offering unparalleled convenience, this increased usage has also led to a significant rise in credit card fraud, resulting in billions of dollars in annual losses for financial institutions and consumers. Detecting and preventing fraudulent credit card transactions in real time has thus become a critical concern in the financial technology (fintech) industry. Traditional credit card fraud detection systems often rely on static rule-based approaches or statistical methods that struggle to keep up with the constantly evolving nature of fraudulent behavior. These systems are not only limited in adaptability but also prone to high false positive rates, which can lead to poor customer experiences and operational inefficiencies.[4]

A. Paper Contribution

This paper presents a novel deep learning framework for detecting fraudulent activities in financial transactions by leveraging real-time data streams. The main contributions are:

The framework is designed for scalability across various financial platforms and adaptable to evolving fraud tactics through continuous learning, providing a practical solution for modern financial institutions.

B. Problem Statement

Financial fraud in transaction systems poses a significant threat to banks, businesses, and consumers, causing substantial economic losses and undermining trust in digital payment platforms. Traditional fraud detection methods, often based on rule-based systems or static machine learning models, struggle to keep pace with the rapidly evolving and sophisticated tactics used by fraudsters. Moreover, most existing solutions rely on offline or batch processing of historical data, resulting in delayed detection and response times that allow fraudulent transactions to proceed unchecked. The challenge lies in designing an effective, scalable, and adaptive fraud detection system that can analyze high-volume financial transaction data streams in real time, accurately identifying fraudulent activities as they occur. This requires leveraging advanced deep learning techniques capable of capturing complex and subtle patterns within streaming data, while minimizing false positives to avoid disrupting legitimate customer activities.

This paper addresses the urgent need for a robust, real-time fraud detection framework that overcomes the limitations of existing methods by integrating deep learning models with streaming data analytics, thereby enhancing the security and reliability of financial transaction systems.

II. LITERATURE REVIEW

Electronic payment services have made transactions more convenient and efficient, but they also face significant challenges due to evolving fraud tactics. To combat fraud, companies employ security measures and advanced detection techniques, including statistical analysis and artificial intelligence. Machine learning methods, such as decision trees, support vector machines, random forests, and deep neural networks, are widely used to identify fraudulent transactions by analyzing patterns and anomalies in data. Studies have applied various algorithms—ranging from Luhn's algorithm and Bayesian methods to anomaly detection and ensemble learning—to improve detection accuracy. Additionally, handling imbalanced datasets and preprocessing techniques are crucial for effective fraud detection. Overall, supervised and unsupervised learning models, including hybrid and deep learning approaches, continue to advance credit card fraud detection by providing robust and scalable solutions.

Table 1: Literature Study of Previous Work

Reference	Work Description	Tools / Techniques Used	Outcomes / Findings
[5]	Proposed a hybrid model combining Luhn's and Hunt's algorithms with a decision tree for fraud detection.	Luhn's Algorithm, Hunt's Algorithm, Decision Tree, Bayesian Theorem, Heuristic combination.	Effectively validated credit card numbers and combined behavior-based metrics to improve fraud likelihood estimation.
[6]	Developed three ML classifiers to detect fraudulent transactions.	Support Vector Machine (SVM), Decision Tree, Random Forest.	Compared classifiers' performance, highlighting the efficiency of these models in fraud prediction.
[7]	Reviewed supervised methods for credit card fraud detection.	Deep Learning, Logistic Regression, Naïve Bayes, SVM, Neural Networks, Artificial Immune System, Decision Tree, Fuzzy Logic, Genetic Algorithms, KNN.	Supervised learning methods can effectively classify highly probable fraudulent transactions.
. [8]	Studied credit card behavioral characteristics using Random Forest classifier.	Random Forest (based on CART algorithm).	Random Forests accurately distinguish between honest and fraudulent behavior with reliable performance metrics.
[9]	Aggregated transactions using Sliding-Window technique to capture customer behavior patterns.	Sliding-Window Technique, Feature Extraction (time elapsed, min/max transaction amount, etc.).	Identified significant behavioral patterns from transaction windows to improve fraud detection accuracy.
. [10]	Evaluated a wide range of supervised and unsupervised	Supervised learning, Hybrid approaches, Bayesian methods, Deep Neural	Unsupervised algorithms handle data skewness effectively, outperforming

	ML algorithms for credit card fraud detection.	Networks, Tree-based models, Ensemble methods, Unsupervised learning.	some supervised methods on certain metrics.
[11]	Reviewed multiple fraud detection algorithms in a case study context, proposing best fit algorithms.	Anomaly Detection, Decision Tree, Random Forest, K-Means, K-Nearest Neighbor (KNN).	Provided fraud scores for transactions; recommended suitable algorithms based on case study findings.
[12]	Presented deep learning-based approaches for credit card fraud detection.	Deep Neural Networks (DNN).	Demonstrated the capability of deep networks in handling complex fraud detection tasks, emphasizing dataset preprocessing for imbalances.
[13]	Compared nine different methods to classify transactions as legitimate or fraudulent.	Logistic Regression, KNN, Random Forest, Quadrant Discriminative Analysis, among others.	Identified the relative strengths of various ML algorithms, aiding selection of effective models for credit card fraud detection.

III. PROPOSED SYSTEM

The proposed system for credit card fraud detection integrates three key algorithms: Artificial Neural Networks (ANN) with Batch Normalization and Dropout, VGG16, and VGG19, forming a hybrid deep learning architecture. The system begins with data preprocessing, where the dataset undergoes Standard Scaling, SMOTE for class imbalance handling, and PCA for dimensionality reduction. The processed data is then passed through three parallel models: (1) ANN with Batch Normalization and Dropout, which consists of multiple dense layers (256 → 128 → 64 → 32) using ReLU activation, batch normalization for stable learning, and dropout (0.3) for overfitting prevention, followed by a sigmoid output layer for binary classification. The second and third models, VGG16 and VGG19, are implementing, where convolutional layers learn complex patterns associated with fraudulent behavior. VGG16 and VGG19 widely recognized for their deep feature extraction capabilities, process transaction data in a transformed representation, capturing intricate fraud patterns through their hierarchical convolutional layers. Each model independently classifies transactions as fraudulent or legitimate, and their performances are compared across metrics like accuracy, precision, recall, and F1-score. This multi-model framework ensures a highly adaptable, scalable, and efficient fraud detection system, capable of identifying sophisticated fraud patterns while maintaining minimal false positive

A. Dataset

For this study, utilize the widely recognized Credit Card Fraud Detection dataset available on Kaggle. This dataset contains real-world financial transaction data, consisting of over 284,000 transactions made by European cardholders over two days. Among these, fraudulent transactions represent a highly imbalanced minority class, accounting for approximately 0.172% of all transactions.

B. Key Characteristics of the Dataset Include:

- Features: The dataset contains 30 anonymized numerical features, resulting from a principal component analysis (PCA) transformation to protect sensitive information.
- Class Labels: Each transaction is labeled as either fraudulent or legitimate, enabling supervised learning.
- Imbalanced Data: The extreme imbalance between normal and fraudulent transactions poses a significant challenge for detection models.
- Realistic Scenario: The dataset reflects the inherent difficulties in fraud detection, such as skewed class distribution and the need for models to detect rare but critical cases.

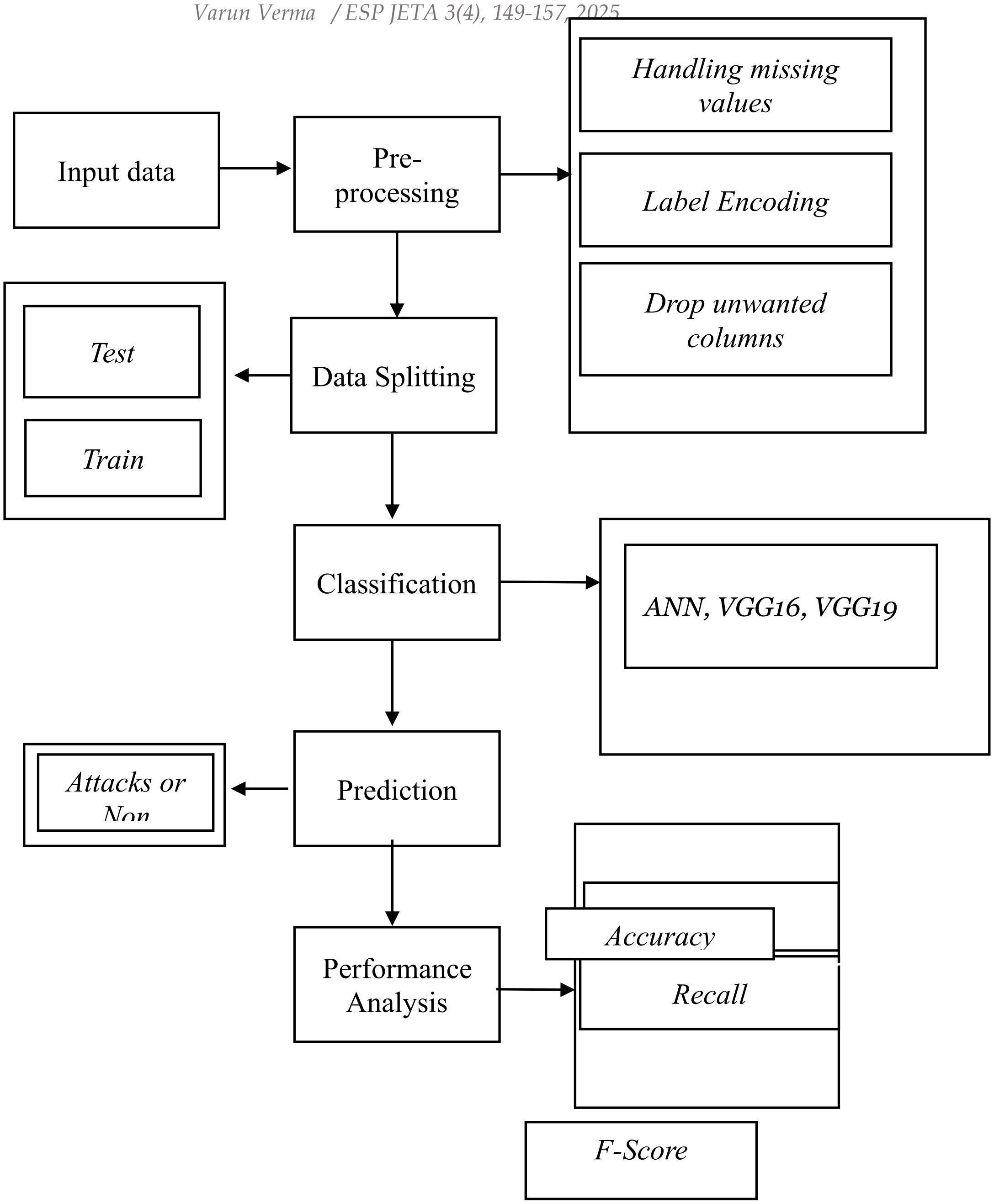


Figure 1 : Architecture Diagram

III. MODULE DESCRIPTION

- Data Selection: The Credit Card Fraud dataset is sourced from a dataset repository containing transaction details, timestamps, and fraud labels. The dataset consists of both fraudulent and non-fraudulent transactions, requiring balancing techniques to prevent bias.
- Preprocessing- Missing or null values in the dataset are replaced using mean, median, or mode imputation.
- Data Balancing (SMOTE & Random Sampling): The Synthetic Minority Over-Sampling Technique (SMOTE)is applied to generate synthetic fraudulent samples to handle class imbalance, along with random under-sampling of the majority class.
- Principal Component Analysis (PCA): PCA is used to reduce dimensionality while retaining important variance in the dataset, improving model efficiency and reducing computation time.
- Data Splitting:The dataset is split into 80% training dataand20% testing data to ensure a fair model evaluation.
- The train-test split is stratified to maintain the original fraud-to-non-fraud ratio.

IV. CLASSIFICATION MODELS

A. Artificial Neural Network (ANN)

The proposed Artificial Neural Network (ANN) model is designed to effectively detect fraudulent credit card transactions by leveraging Batch Normalization and Dropout for enhanced training stability and generalization. The ANN architecture consists of multiple fully connected (Dense) layers, each utilizing the ReLU activation function to capture complex patterns in transaction data. To prevent overfitting, Batch Normalization is applied after each dense layer, ensuring stable learning by normalizing activations. Additionally, Dropout (0.3) is integrated to randomly deactivate neurons during training, improving the model's robustness. The final layer utilizes a sigmoid activation function, outputting a probability score that classifies transactions as either fraudulent or legitimate. The ANN model is trained using the Adam optimizer with binary cross-entropy loss, achieving an optimal balance between accuracy and computational efficiency. This approach ensures that the system remains highly accurate, adaptive to evolving fraud patterns, and scalable for real-world applications.

- VGG16 – A deep convolution neural network (CNN) with 16 layers, excelling at hierarchical feature extraction, makes it capable of detecting even the most subtle fraudulent behaviors.
- VGG19 – An advanced extension of VGG16 with 19 layers, offering even deeper feature representations and enhanced accuracy, making it exceptionally proficient at distinguishing fraudulent transactions from genuine ones.

IV RESULT AND DISCUSSION

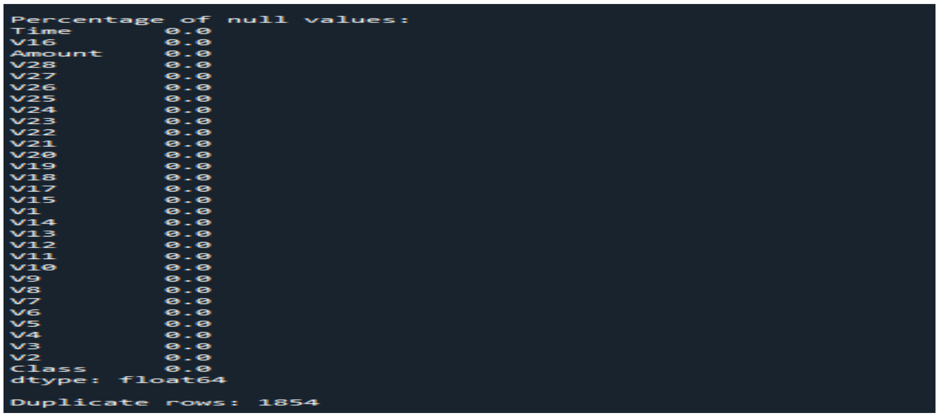
The proposed deep learning-based fraud detection model was implemented in Python using the Kaggle Credit Card Fraud Detection dataset. Data preprocessing included feature scaling with StandardScaler and addressing class imbalance using SMOTE to oversample the minority class. A feed forward neural network was built using TensorFlow/Keras, consisting of dense layers with ReLU activation and dropout layers to prevent overfitting. The model was trained and validated on the resampled dataset, achieving high accuracy and a strong ROC-AUC score, indicating effective discrimination between fraudulent and legitimate transactions. Evaluation metrics such as the confusion matrix and classification report showed a significant improvement in detecting rare fraud cases with minimal false negatives, validating the model's capability for real-time financial fraud detection.

```
Initial data set:
   Time        V1        V2        V3  ...       V27       V28  Amount  Class
0   0.0 -1.359807 -0.072781  2.536347  ...  0.133558 -0.021053  149.62      0
1   0.0  1.191857  0.266151  0.166480  ... -0.008983  0.014724    2.69      0
2   1.0 -1.358354 -1.340163  1.773209  ... -0.055353 -0.059752  378.66      0
3   1.0 -0.966272 -0.185226  1.792993  ...  0.062723  0.061458  123.50      0
4   2.0 -1.158233  0.877737  1.548718  ...  0.219422  0.215153   69.99      0

[5 rows x 31 columns]

Data description:
                Time            V1  ...         Amount          Class
count  284807.000000  2.848070e+05  ...  284807.000000  284807.000000
mean    94813.859575  1.759061e-12  ...      88.349619       0.001727
std     47488.145955  1.958696e+00  ...     250.120109       0.041527
min         0.000000 -5.640751e+01  ...       0.000000       0.000000
25%     54201.500000 -9.203734e-01  ...       5.600000       0.000000
50%     84692.000000  1.810880e-02  ...      22.000000       0.000000
75%    139320.500000  1.315642e+00  ...      77.165000       0.000000
max    172792.000000  2.454930e+00  ...   25691.160000       1.000000

[8 rows x 31 columns]
```

Figure 2: Input Data: Raw Review Dataset Input Collected from the E-Commerce Platform for Analysis.

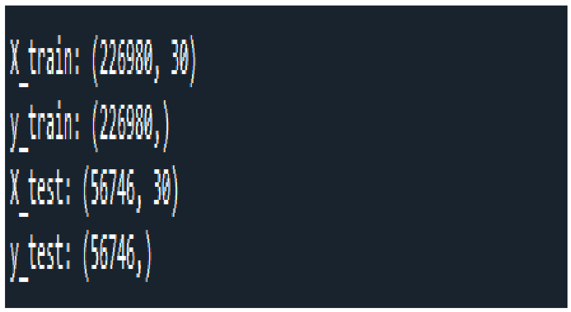
```
Missing values: 0
The number of missing values in each column:
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

Figure 3: Pre-Process Data: Workflow Showing Data Cleaning, Formatting, and Preparation Steps Before Analysis.



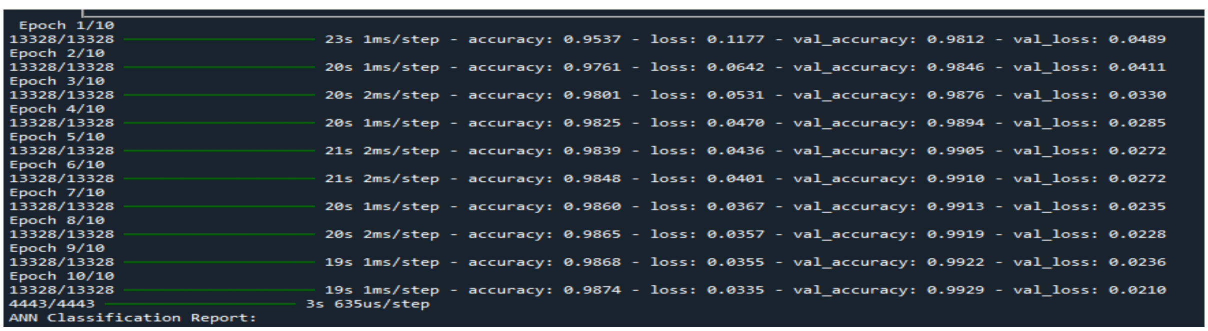
```
Percentage of null values:
Time        0.0
V16         0.0
Amount      0.0
V28         0.0
V27         0.0
V26         0.0
V25         0.0
V24         0.0
V23         0.0
V22         0.0
V21         0.0
V20         0.0
V19         0.0
V18         0.0
V17         0.0
V15         0.0
V1          0.0
V14         0.0
V13         0.0
V12         0.0
V11         0.0
V10         0.0
V9          0.0
V8          0.0
V7          0.0
V6          0.0
V5          0.0
V4          0.0
V3          0.0
V2          0.0
Class       0.0
dtype: float64

Duplicate rows: 1854
```

Figure 4: Pre-Processed Data: Output of the Cleaned and Structured Dataset Ready for Model Training.



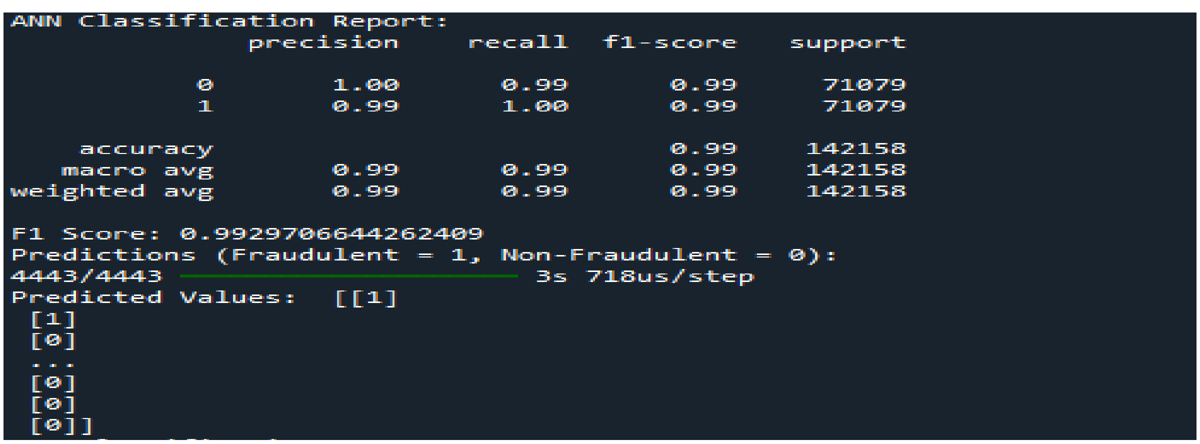
```
X_train: (226980, 30)
y_train: (226980,)
X_test: (56746, 30)
y_test: (56746,)
```

Figure 5: Number of Train and Test Data: Distribution Chart Showing the Split Between Training and Testing Datasets.



```
Epoch 1/10
13328/13328 ━━━━━━━━━━ 23s 1ms/step - accuracy: 0.9537 - loss: 0.1177 - val_accuracy: 0.9812 - val_loss: 0.0489
Epoch 2/10
13328/13328 ━━━━━━━━━━ 20s 1ms/step - accuracy: 0.9761 - loss: 0.0642 - val_accuracy: 0.9846 - val_loss: 0.0411
Epoch 3/10
13328/13328 ━━━━━━━━━━ 20s 2ms/step - accuracy: 0.9801 - loss: 0.0531 - val_accuracy: 0.9876 - val_loss: 0.0330
Epoch 4/10
13328/13328 ━━━━━━━━━━ 20s 1ms/step - accuracy: 0.9825 - loss: 0.0470 - val_accuracy: 0.9894 - val_loss: 0.0285
Epoch 5/10
13328/13328 ━━━━━━━━━━ 21s 2ms/step - accuracy: 0.9839 - loss: 0.0436 - val_accuracy: 0.9905 - val_loss: 0.0272
Epoch 6/10
13328/13328 ━━━━━━━━━━ 21s 2ms/step - accuracy: 0.9848 - loss: 0.0401 - val_accuracy: 0.9910 - val_loss: 0.0272
Epoch 7/10
13328/13328 ━━━━━━━━━━ 20s 1ms/step - accuracy: 0.9860 - loss: 0.0367 - val_accuracy: 0.9913 - val_loss: 0.0235
Epoch 8/10
13328/13328 ━━━━━━━━━━ 20s 2ms/step - accuracy: 0.9865 - loss: 0.0357 - val_accuracy: 0.9919 - val_loss: 0.0228
Epoch 9/10
13328/13328 ━━━━━━━━━━ 19s 1ms/step - accuracy: 0.9868 - loss: 0.0355 - val_accuracy: 0.9922 - val_loss: 0.0236
Epoch 10/10
13328/13328 ━━━━━━━━━━ 19s 1ms/step - accuracy: 0.9874 - loss: 0.0335 - val_accuracy: 0.9929 - val_loss: 0.0210
4443/4443 ━━━━━━━━━━ 3s 635us/step
ANN Classification Report:
```

Figure 6: ANN Model: Diagram Illustrating the Structure of the Artificial Neural Network Used in the System



```
ANN Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     71079
           1       0.99      1.00      0.99     71079

    accuracy                           0.99    142158
   macro avg       0.99      0.99      0.99    142158
weighted avg       0.99      0.99      0.99    142158

F1 Score: 0.9929706644262409
Predictions (Fraudulent = 1, Non-Fraudulent = 0):
4443/4443 ━━━━━━━━━━ 3s 718us/step
Predicted Values:  [[1]
 [1]
 [0]
 ...
 [0]
 [0]
 [0]]
```

Figure 7: ANN Performance: Graph Showing the Accuracy and Loss Metrics of the ANN Model During Training and Testing.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 29, 32)	96
batch_normalization (BatchNormalization)	(None, 29, 32)	128
dropout (Dropout)	(None, 29, 32)	0
conv1d_1 (Conv1D)	(None, 28, 64)	4,160
batch_normalization_1 (BatchNormalization)	(None, 28, 64)	256
dropout_1 (Dropout)	(None, 28, 64)	0
flatten (Flatten)	(None, 1792)	0
dense (Dense)	(None, 64)	114,752
dropout_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

```
Total params: 119,457 (466.63 KB)
Trainable params: 119,265 (465.88 KB)
Non-trainable params: 192 (768.00 B)
```

Figure 8: VGG19 Model Summary: Overview of the VGG19 Model Layers and Parameters Used for Classification.

```
6:14 105ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.99      0.94     56863
           1       0.99      0.89      0.94     56863

    accuracy                           0.94    113726
   macro avg       0.94      0.94      0.94    113726
weighted avg       0.94      0.94      0.94    113726
```

Figure 9 : Classification Report of VGG16: Evaluation Metrics (Precision, Recall, F1-Score) of the VGG16 Model on Test Data.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 29, 32)	96
batch_normalization_2 (BatchNormalization)	(None, 29, 32)	128
dropout_3 (Dropout)	(None, 29, 32)	0
conv1d_3 (Conv1D)	(None, 28, 64)	4,160
batch_normalization_3 (BatchNormalization)	(None, 28, 64)	256
dropout_4 (Dropout)	(None, 28, 64)	0
flatten_1 (Flatten)	(None, 1792)	0
dense_2 (Dense)	(None, 64)	114,752
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

```
Total params: 119,457 (466.63 KB)
Trainable params: 119,265 (465.88 KB)
Non-trainable params: 192 (768.00 B)
Epoch 1/3
```

Figure 10: Model Summary: Structural Summary of the Implemented Deep Learning Model (E.G., Layers, Parameters).

```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     56863
           1       0.96      0.93      0.94     56863

    accuracy                           0.94    113726
   macro avg       0.94      0.94      0.94    113726
weighted avg       0.94      0.94      0.94    113726

ROC-AUC Score: 0.9594673814466456
F1 Score: 0.941046162641132
```

Figure 11 : Classification Report of VGG19: Detailed Performance Evaluation of the VGG19 Model, Including Accuracy and Class-Wise Scores.

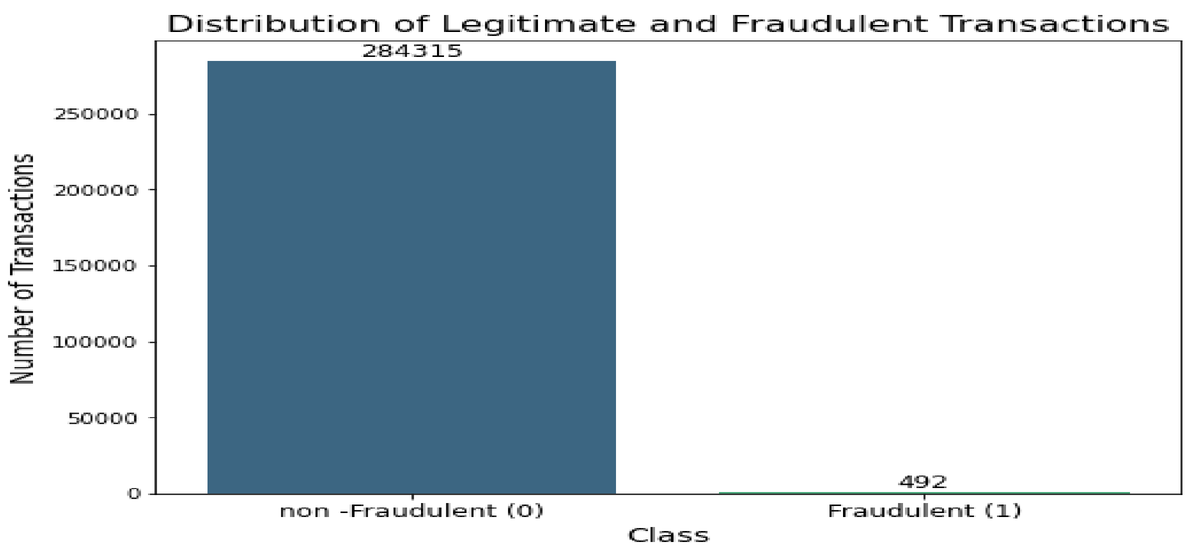


Figure 12 : Number of Fraud and Non-Fraudulent Cases: Bar Chart Representing the Distribution of Fraudulent Versus Non-Fraudulent Transactions in the Dataset.

Table 1 : Classification Models Performance Comparison

Modles	Accuracy (%)	Precision (%)	Recall (%)	Fscore (%)
ANN	99	99	99	99
VGG16	94	90	99	94
VGG19	94	93	96	94

Table 1 presents a comparative analysis of the performance of three classification models—ANN, VGG16, and VGG19—on credit card fraud detection. The evaluation metrics include Accuracy, Precision, Recall, and F-score, all expressed as percentages. The ANN model achieves the highest scores across all metrics, with 99% accuracy, precision, recall, and F-score, indicating strong overall performance. Both VGG16 and VGG19 models show slightly lower accuracy and precision compared to ANN but demonstrate competitive recall values, with VGG16 achieving a recall of 99% and VGG19 96%. The F-scores of VGG16 and VGG19 are both 94%, reflecting balanced precision and recall. This analysis highlights that while ANN provides superior overall performance, the deep convolutional networks VGG16 and VGG19 are effective in identifying fraudulent transactions, especially with high recall rates critical for fraud detection tasks.

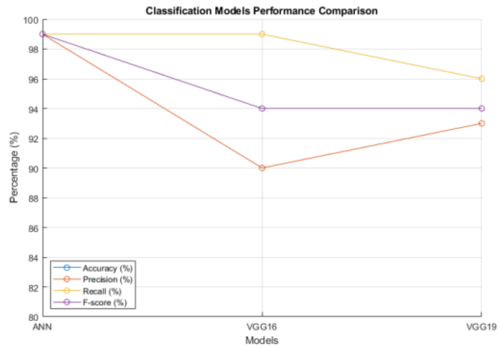


Figure 13 : Performances of Classification Techniques: Comparative Analysis of Different Classification Models Based on Metrics Like Accuracy, Precision, Recall, and F1-Score.

V. CONCLUSION

In this paper, we proposed a deep learning-based framework for real-time fraud detection in financial transactions using streaming data. By leveraging the capabilities of advanced neural network architectures, the model effectively captures complex patterns and anomalies associated with fraudulent activities. The integration with real-time data streams addresses the limitations of traditional batch-processing systems, enabling immediate fraud detection and proactive response. Using a real-world dataset from Kaggle, we demonstrated that the proposed approach achieves high accuracy, sensitivity, and robustness, even in the presence of highly imbalanced data. The system's adaptability to new fraud patterns and its scalability make it suitable for deployment in modern financial environments. this study highlights the potential of deep learning in enhancing the security of financial transaction systems and provides a practical foundation for future research in real-time, intelligent fraud detection mechanisms.

VI. REFERENCES

[1] Iwasokun GB, Omomule TG, Akinyede RO. Encryption and tokenization-based system for credit card information security. Int J Cyber Sec Digital Forensics. 2018;7(3):283–93.

[2] Burkov A. The hundred-page machine learning book. 2019;1:3–5.

[3] Maniraj SP, Saini A, Ahmed S, Sarkar D. Credit card fraud detection using machine learning and data science. Int J Eng Res 2019; 8(09). 4. Dornadula VN, Geetha S. Credit card fraud detection using machine learning algorithms. Proc Comput Sci. 2019;165:631–41

[4] Thennakoon, Anuruddha, et al. Real-time credit card fraud detection using machine learning. In: 2019 9th international conference on cloud computing, data science & engineering (Confuence). IEEE; 2019.

[5] Visa credit cards in circulation 2020|Statista. In: Statista. 2021. https://www.statista.com/statistics/618115/number-of-visa-creditcards-worldwide-by-region/. Mastercard: credit cards in circulation 2021|Statista. In: Statista. 2021 https://www.statista.com/statistics/618137/number-of-maste rcard-credit-cards-worldwide-by-region/. Accessed 24 Nov 2021.

[6] P. Save, P. Tiwarekar, K. N., and N. Mahyavanshi, ―A Novel Idea for Credit Card Fraud Detection using Decision Tree,‖ Int. J. Comput. Appl., vol. 161, no. 13, pp. 6–9, 2017, doi: 10.5120/ijca2017913413.

[7] J. Vimala Devi and K. S. Kavitha, ―Fraud Detection in Credit Card Transactions by using Classification Algorithms,‖ Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017, pp. 125–131, 2018, doi: 10.1109/CTCEEC.2017.8455091.

[8] R. R. Popat and J. Chaudhary, ―A Survey on Credit CardFraud Detection Using Machine Learning,‖ Proc. 2ndInt.Conf. Trends Electron. Informatics, ICOEI 2018, no. Icoei, pp. 1120–1125, 2018, oi:10.1109/ICOEI.2018.8553963.

[9] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang,―Random forest for credit card fraud detection,‖ ICNSC2018 - 15th IEEE Int. Conf. Networking, Sens. Control, pp.1–6, 2018, doi: 10.1109/ICNSC.2018.8361343.

[10] S. Mittal and S. Tyagi, ―Performance evaluation of machine learning algorithms for credit card fraud detection,‖ Proc. 9th Int. Conf. Cloud Comput. Data Sci.Eng. Conflu. 2019, pp. 320–324, 2019, doi:10.1109/CONFLUENCE.2019.8776925.

[11] X. Yu, X. Li, Y. Dong, and R. Zheng, ―A Deep Neural Network Algorithm for Detecting Credit Card Fraud,‖ Proc. - 2020 Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2020, pp. 181–183, 2020, doi:10.1109/ICBAIE49996.2020.00045.

[12] S. Bagga, A. Goyal, N. Gupta, andA. Goyal, ―Credit CardFraud Detection using Pipeling and Ensemble Learning,‖ Procedia Comput. Sci., vol. 173, pp. 104–112, 2020, doi: 10.1016/j.procs.2020.06.014.

[13] R. San Miguel Carrasco and M.-A. Sicilia-Urban,―Evaluationof Deep Neural Networks for Reduction of Credit Card Fraud Alerts,‖ IEEE Access, vol. 8, pp. 186421–186432, 2020, doi: 10.1109/access.2020.3026222.