Anomaly Detection in Credit Card Fraud: A Comparative Analysis of Autoencoder, Isolation Forest, and One-Class SVM

Abstract

Credit card fraud detection is an important research problem but is considered to be a difficult problem due to the nature of the transaction data where the number of fraudulent transactions is small compared to normal transactions. This project implements an anomaly detection system through the use of an autoencoder which is a kind of deep learning model with unsupervised learning. The idea of the model is to discover those transactions that normally are not like the rest as they may possibly be fraudulent. To compare the results, two other transfer learning models are used alongside the proposed autoencoder: Isolation Forest and One-Class SVM. The data is taken from Kaggle and the data covers 284806 transactions with 30 features obtained by applying PCA. The preprocessing part included scaling of the data and applying feature reduction to 15 features. The normal transaction data was used to train the autoencoder and performance metrics include accuracy, precision, recall and ROC AUC value. The results best ROC AUC of 0.94 for the autoencoder ensures the models effectiveness for fraud detection over the transfer learning models. This work illustrates the applicability of deep learning in the real-world financial abnormalities and performs the comparative analysis of classical and deep learning-based approaches.

I. Introduction

Digital payment systems adoption resulted in a sharp growth of credit card transaction volumes. The fast-growing digital payment systems have triggered a massive increase in fraudulent transactions which presents difficult obstacles to financial institutions worldwide. Online payment systems face weakened consumer trust due to credit card fraud which brings about considerable financial losses. Detecting fraudulent transactions remains a challenging task due to the highly imbalanced nature of the datasets, where fraudulent transactions account for a tiny fraction of the total, as well as the dynamic and evolving nature of fraud patterns. Strong and flexible fraud detection systems need to be developed with adaptability to emerging fraud patterns because of these technical obstacles.

The detection of abnormal patterns in credit card transactions has become an efficient solution for detecting fraudulent activities because of anomaly detection methods. Supervised learning models need labeled fraudulent and non-fraudulent transaction data while anomaly detection methods use legitimate data patterns to detect outlier transactions which they label as potential frauds. The method performs well under conditions where the available fraudulent data is either sparse or hard to label. Autoencoders represent a preferred deep learning-based anomaly detection approach because they effectively detect normal and anomalous patterns while capturing complex models in high-dimensional data [1], [2].

Various studies show that autoencoders can efficiently detect financial frauds within data sets. The research from Najmi Rosley et al. [1] proposed deep autoencoders for credit card fraud detection through reconstruction error as their detection metric. Similarly, Priyanka Sharma et al. [2] compared autoencoders against neural networks and demonstrated that autoencoders provide better results for detecting fraud patterns. According to Nguyen et al. [3] deep learning solutions that include autoencoders prove superior to conventional machine learning approaches like logistic regression and decision trees for fraud detection tasks. The research findings demonstrate how autoencoders represent a strong technique for detecting anomalies.

The simplicity and efficiency of Isolation Forest and One-Class Support Vector Machines (SVM) make them standard choices for fraud detection tasks which operate through unsupervised learning methods. Gayatri Ketepalli et al. [4] and Shan Jiang et al. [5] conducted performance analyses of traditional models including their capacity limitations during fraud detection processes involving complex high-dimensional datasets. Hosein Fanai et al. [6] built a hybrid system which unites deep autoencoders with traditional classifiers to achieve superior fraud detection accuracy. These classic models excel in computational efficiency yet fall short when it comes to adjusting to changing fraud patterns that occur swiftly [7] and [8].

The research went through an evaluation between deep learning autoencoders and conventional methods like Isolation Forest and One-Class SVM for detecting anomalies in credit card transactions. A total of 284,806 transactions from the Kaggle Credit Card Fraud Dataset which contains only 492 cases of fraud while maintaining this ratio of distribution imbalance. The study uses key metrics like ROC-AUC and precision and recall to assess the effectiveness of these models after a large number of experiments and performance evaluations. The study demonstrates both positive and negative points of these methods using experimental results to show how deep learning approaches excel over standard fraud detection methods.

II. Literature Review

The detection of anomalies serves as the fundamental approach in combating credit card fraud through various developed techniques. Multiple analytical methods from classical machine learning models and contemporary deep learning structures have developed stronger capabilities for detecting financial transaction fraud.

The detection of credit card transaction anomalies has commonly used two traditional machine learning algorithms: Isolation Forest alongside Support Vector Machines (SVM). Swati Gupta et al. [9] conducted a study on the behavior of Isolation Forest for fraud identification and identified its computational speed but reported its weaknesses when applied to unbalanced data distributions. Yogendra et al. [10] joined traditional machine learning algorithms with Convolutional Neural Networks (CNNs) for features extraction and classification purposes which boosted detection performance.

Modern deep learning models overcome traditional techniques because they effectively process complex high-dimensional datasets. J. Chen [11] introduced a Deep Convolutional Neural Network model which delivered higher accuracy and recall results than conventional machine learning solutions. A systematic review performed by Cherif et al. [12] proved that deep learning methods surpass traditional methods regarding adaptability and accuracy in fraud detection. The review presented a strong case for system scalability to become a vital factor in achieving fraud detection success in real-world scenarios.

Several researches have demonstrated success with hybrid frameworks that unite traditional methods with deep learning approaches. A Spark-integrated deep learning system for fraud detection has been presented by Sumaya Sanober et al. [13] while taking advantage of distributed systems scalability. A hybrid model built on BiLSTM-BiGRU components delivered exceptional recall and precision to detect anomalies in financial transactions according to Hassan Najadat et al. [14].

Ensemble and sequential models have gained attention as fraud detection techniques in the current period. The deep learning ensemble approach proposed by Mienye et al. [15] uses SMOTE-ENN data resampling techniques to deal with fraud datasets that are highly class imbalanced. Similarly, Forough et al. [16] developed an ensemble of deep sequential models with a voting mechanism to improve accuracy and reduce false positives. Representation learning techniques have shown effectiveness for detecting fraud. A deep representation learning framework presented by Zhenchuan Li utilized a full center loss function which increased feature separability while reducing false positive detection rates [17]. This approach delivered excellent performance for spotting fraudulent behavior from non-fraudulent actions within datasets with large class imbalances.

Several innovative frameworks which use unsupervised learning methods such as Marta Catillo et al. [18] proposed AutoLog, an autoencoder-based model designed for anomaly detection in system logs, which shares similarities with detecting anomalies in transaction data.Overall, these studies emphasize the evolution of fraud detection methodologies, from traditional machine learning models to deep learning and hybrid frameworks. While traditional methods like Isolation Forest remain computationally efficient, modern approaches such as deep autoencoders, ensemble models, and representation learning provide superior performance, particularly in handling complex and imbalanced datasets.

III. Methodology

3.1 Dataset Description and Source

The dataset used is the Credit Card Fraud Detection Dataset from Kaggle. The dataset contains 284,806 credit card transactions where features were generated through Principal Component Analysis (PCA) for privacy protection. Only 492 transactions (0.17%) are labeled as fraudulent(Class = 0) which makes the dataset highly imbalanced. The remaining 284,314 transactions are labeled as non-fraudulent (Class = 0). A remarkable imbalance between non-fraudulent and fraudulent transactions helps to understand the actual financial world situation where fraudulent activities make up a minimal percentage of transactions. The dataset features are-

- Time: The seconds elapsed between the transaction and the first transaction in the dataset.
- V1 to V28: These are 28 numerical features derived from Principal Component Analysis (PCA) to ensure privacy. The actual meaning of these features is not provided due to confidentiality.
- Amount: The transaction amount. This can be useful for identifying high-value frauds or understanding transaction patterns.
- Class: The target variable indicating whether a transaction is fraudulent (1) or non-fraudulent (0).

The first 30 columns consisting of Time, V1 to V28, and Amount contain numerical data types that are stored as float64. The one and only categorical feature in the dataset is stored as int in the Class column. The derived features (V1 to V28) through PCA provide privacy protections by stripping confidential information yet keeping vital patterns necessary for anomaly detection. Transaction timing information in the Time field enables fraud cluster detection while the Amount field shows values that indicate potential fraud. The 28 PCA-transformed components named V1 to V28 successfully capture key variances without revealing any personal information about cardholders or merchants.

3.2.2 Feature Selection

The research dataset consists of 30 features named V1 to V28 together with Time and Amount columns. All these features remain unidentified since they lack proper names or clear meanings. The mathematical processes represent patterns within original transaction data but remain elusive in their exact composition. Multiple feature analysis in datasets sometimes produces a situation known as the “curse of dimensionality”. The data processing becomes harder while the model performance suffers from handling particular features when the number of features we introduce grows significantly. A solution for this issue required us to implement a method known as Principal Component Analysis (PCA). PCA functions as an algorithm that enables feature reduction by maintaining maximum available data information. The system determines new 'principal components' that consist of combined original features as part of its operation. According to the amount of data variance the new components are arranged in specific order. The first captured component explains the maximum variance while the second component shows the second largest amount of variance in the sequence. The correlation heatmap shows the relationships between the anonymized features V1 to V28 and Time and Amount.

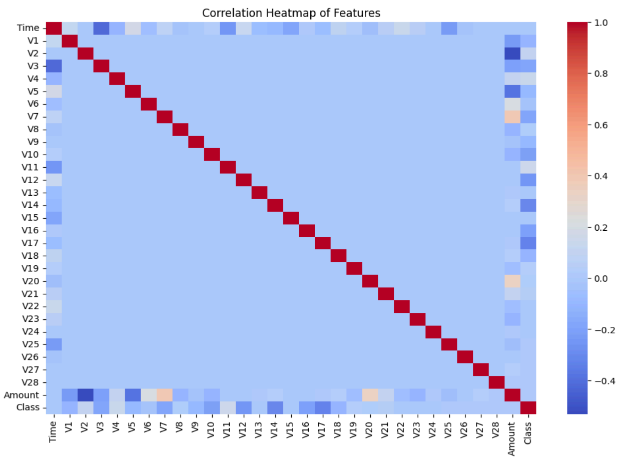

Figure 1. Correlation Heatmap of Features

3.2.3 Feature Extraction

PCA reduced the initial collection of 32 dimensions consisting of 30 features together with Time and Amount which was transformed into a new set of just 15 instances. The conversion from 32 dimensions to 15 dimensions still allows us to achieve 95% data retention of crucial information. The simplification of input data through this step proved useful due to its two-fold benefits first in reducing the information complexity and second in lowering computational requirements for pattern detection within time series streams.

3.3 Training and Testing Strategy

Training and testing the model effectively is crucial for ensuring its robustness in detecting fraudulent transactions in real-world financial systems. The following subsections outline the data splitting strategy, training methodology, model evaluation, and threshold selection process applied in this study.

3.3.1 Data Splitting

The dataset was divided into two subsets to facilitate model training and evaluation:

- Training Set (80%): Used to train the models to learn the underlying patterns in the data.
- Testing Set (20%): Used to evaluate how well the trained models generalize to unseen transactions.

Since only 0.17% of the total transactions were fraudulent, stratified sampling was employed during data splitting. This ensured that both the training and testing sets contained a proportional representation of normal and fraudulent transactions.

3.3.2 Training the Model

The models were trained using different strategies based on their learning mechanisms.

a) Autoencoder

The autoencoder is an unsupervised model that learns to reconstruct normal transactions, making it well-suited for anomaly detection.

Training Strategy:

- Only normal transactions (Class = 0) from the training dataset were used.
- Fraudulent transactions were excluded from training since they are considered anomalies during detection.
- The model learned the hidden patterns in normal transactions, allowing it to reconstruct them with minimal error.
- Fraudulent transactions, being inherently different from normal transactions, would exhibit higher reconstruction errors, making them easier to flag as anomalies.

b) Isolation Forest & One-Class SVM

Unlike autoencoders, Isolation Forest and One-Class SVM use anomaly detection principles based on statistical deviations and clustering. These models were trained on both normal and fraudulent transactions and optimized with hyperparameter tuning to improve performance.

3.3.3 Testing the Model

Once trained, the models were evaluated using the full testing set, which included both normal and fraudulent transactions.

Autoencoder Testing:

- Each transaction was passed through the autoencoder.
- A reconstruction error was computed for each transaction.
- Transactions with high reconstruction errors were flagged as anomalies (potential fraud).
- Isolation Forest & One-Class SVM Testing:
- These models classified transactions based on learned decision boundaries.
- They assigned a fraud probability score to each transaction, determining whether it should be considered an anomaly.

3.3.4 Threshold Selection

A threshold value was established to distinguish normal transactions from fraudulent ones. The 95th percentile of reconstruction errors for normal transactions was used as the threshold. This means that 95% of normal transactions had reconstruction errors below this value, while any transaction exceeding this error threshold was classified as fraudulent. This threshold ensured a balance between sensitivity (recall) and specificity (precision) in fraud detection.

The Autoencoder Model Workflow illustrated in Figure 2 presents the step-by-step process, from input encoding to anomaly detection through reconstruction error analysis.

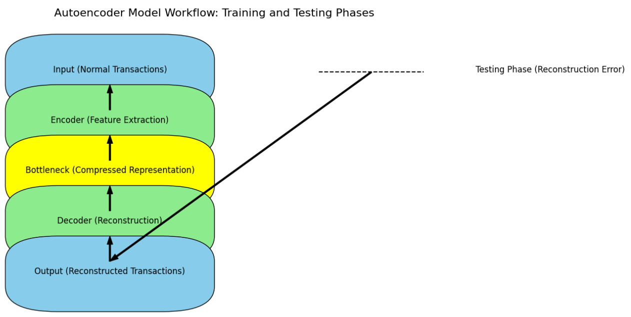
Autoencoder Model Workflow: Training and Testing Phases
Input (Normal Transactions)
Encoder (Feature Extraction)
Bottleneck (Compressed Representation)
Decoder (Reconstruction)
Output (Reconstructed Transactions)
Testing Phase (Reconstruction Error)

Figure 2. Autoencoder Model Workflow

3.4 Parameters of Each Model

Each model was optimized with carefully selected hyperparameters to enhance anomaly detection performance.

The details of each model's configuration are outlined below.

3.4.1 Autoencoder

The autoencoder was designed with a multi-layer architecture to learn and reconstruct normal transaction patterns effectively.

- Input Layer: 15 neurons corresponding to PCA-reduced features.
- Encoder Layers: First hidden layer with 16 neurons using ReLU activation.
- Bottleneck layer with 8 neurons, representing a compressed latent space.
- Decoder Layers:
- A mirrored structure with 16 neurons (ReLU activation).
- Output layer with 15 neurons using Sigmoid activation for reconstruction.
- Loss Function: Mean Squared Error (MSE) to minimize reconstruction errors.
- Optimizer: Adam optimizer for efficient learning.
- Epochs: Trained for 50 epochs to achieve stable reconstruction ability.
- Batch Size: A batch size of 256 was used to ensure effective training without excessive memory consumption.

3.4.2 Isolation Forest

The Isolation Forest model was optimized to detect anomalies by isolating fraudulent transactions using tree-based partitioning. The contamination rate Set to 0.01, assuming that 1% of the dataset contains anomalies. This setting helps the model to efficiently distinguish fraudulent transactions from normal ones.

3.4.3 One-Class SVM

One-Class SVM was configured to create a boundary around normal transactions and detect deviations as anomalies.

- Kernel: Radial Basis Function (RBF) kernel to capture non-linear relationships in the data.
- Nu Parameter: Set to 0.01, defining the upper limit for the proportion of transactions classified as anomalies.
- Gamma Parameter: Set to 0.1, controlling the influence of individual data points on the decision boundary.

These hyperparameters were chosen based on experimental tuning to ensure optimal performance across all models while balancing accuracy, recall, and computational efficiency.

3.5 Summary of Model Design

The table provides a comparative overview of the models employed in this research, highlighting their training data utilization, anomaly detection methods, computational efficiency, and ideal application scenarios. The Autoencoder, leveraging only normal transactions, excels at capturing complex patterns through reconstruction error analysis. In contrast, the Isolation Forest, using random partitioning of both normal and fraudulent data, offers the fastest computational efficiency, making it suitable for large-scale, real-time detection. The One-Class SVM utilizes decision boundaries to detect anomalies, but its lower computational efficiency limits its scalability, making it more appropriate for datasets requiring intricate boundary-based analysis.

Table 1
Summary of Model Design

Aspect	Autoencoder	Isolation Forest	One-Class SVM
Training Data	Only normal transactions	Both normal and fraudulent transactions	Both normal and fraudulent transactions
Anomaly Detection Method	Reconstruction error	Random partitioning	Decision boundary
Computational Efficiency	Moderate	High (fastest)	Low (slowest)
Best For	Capturing hidden patterns in data	Large-scale datasets with quick detection needs	Complex boundary-based anomaly detection

The methodology adopts a structured method for detecting credit card anomalies which combines deep learning models with traditional machine learning approaches. According to the Autoencoder implementation, deep learning-based reconstruction anomaly detection allows it to decode normal transaction patterns in order to detect irregularities potentially stemming from fraud. Traditional machine learning benchmarks include both Isolation Forest and One-Class SVM which use statistical and boundary-based methods for anomaly classification. A complete evaluation process includes using accuracy, precision, recall and computational efficiency metrics to assess the effectiveness of these models. After presenting experimental findings the paper shows a comparative analysis of models through a review of their strengths while identifying their limitations for real-world use in fraud detection systems.

IV. Result and Discussion

4.1 Comparison of Models

A comparison of the performance of the Autoencoder, a deep learning model, with two traditional transfer learning models: Isolation Forest and One-Class SVM have been performed in this section. These models were chosen for their ability to detect anomalies. It is essential for identifying fraudulent transactions in a dataset where fraud is rare. A comprehensive comparison among these three models are given below:

Table 2
Comparison of the Models

Criteria	Autoencoder	Isolation Forest	One-Class SVM
Model Type	Unsupervised deep learning model (Autoencoder)	Ensemble learning model (tree-based)	Support Vector Machine variant (unsupervised)

Training Data	Trained only on normal transactions (Class = 0)	Trained on both normal and anomalous data (but focuses on anomalies)	Trained only on normal transactions (Class = 0)
Learning Approach	Learns to reconstruct normal data and detects anomalies based on reconstruction errors	Isolates anomalies by randomly partitioning the data	Learns a decision boundary to separate normal data from anomalies
Strengths	- Can learn complex, non-linear patterns in the data. - Effective at capturing intricate relationships and anomalies. - Works well with imbalanced datasets.	- Fast training and prediction. - Simple and effective for large datasets. - No need for labeled anomalies.	- Flexible with different kernels, especially for high-dimensional data. - Suitable for identifying outliers in complex data distributions.
Weaknesses	- Computationally expensive to train, especially on large datasets. - Requires significant tuning for optimal performance.	- Performs poorly with complex and high-dimensional data. - Struggles with datasets that require capturing intricate patterns.	- Sensitive to parameters and requires careful tuning. - Can be computationally expensive, especially on large datasets.
Ability to Handle Imbalanced Data	Very effective for imbalanced datasets since it is trained on normal transactions only.	Performs well on imbalanced data but struggles to capture complex patterns.	Can handle imbalanced data well but struggles with accuracy in identifying anomalies.
Data Preprocessing	Requires normalization of data; PCA may be used for dimensionality reduction to speed up training and improve performance.	No heavy preprocessing required, but feature scaling may help.	Data normalization required; works best with a clear separation of normal and anomalous data.
Model Complexity	High model complexity due to neural network architecture, which requires significant computational power for training and inference.	Low model complexity, easier to implement and tune.	Moderate model complexity with the need for parameter tuning for optimal performance.
Scalability	Scalable, but training time increases with larger datasets and higher-dimensional data.	Highly scalable for large datasets, with fast training times.	May struggle with scalability for large datasets, particularly in high-dimensional spaces.

4.1 Performance Metrics

Table 3
Performance Metrics Results of the Models

Metric	**Autoencoder**	**Isolation Forest**	**One-Class SVM**
ROC AUC	0.94	0.05	0.06
Precision (Fraud)	0.03	0.11	0.06
Recall (Fraud)	0.86	0.66	0.87

It was observed that the performance of Autoencoder, Isolation Forest, and One-Class SVM models was neither consistent across all the three datasets nor similar in terms of efficiency and accuracy. Among the models tested, the Autoencoder had the best results with ROC AUC of 0.94 evidencing its potential as an approach to effectively distinguish between normal and fraudulent transactions. It also attained a high recall of.86 that means most of the fraudulent transactions were correctly captured although its precision was low with a score of .03 meaning many false positives. However, Isolation Forest gave the worst results with ROC AUC of 0.05 proving that it is not capable of distinguishing fraudulent transactions. It registered a low accuracy of 0.11; therefore, it misdiagnosed many fraud cases, and it had an average recall of 0.66. One-Class SVM had slightly different results with ROC AUC = 0.06 and thus showing that it was not as helpful as the autoencoder. Although its recall was high (0.87) similar to that of the autoencoder, it had low precision (0.06) which means that many of the results were false positives. Overall, while the autoencoder excelled in distinguishing fraud, all models struggled with precision, highlighting the trade-off between catching fraudulent transactions and avoiding false positives.

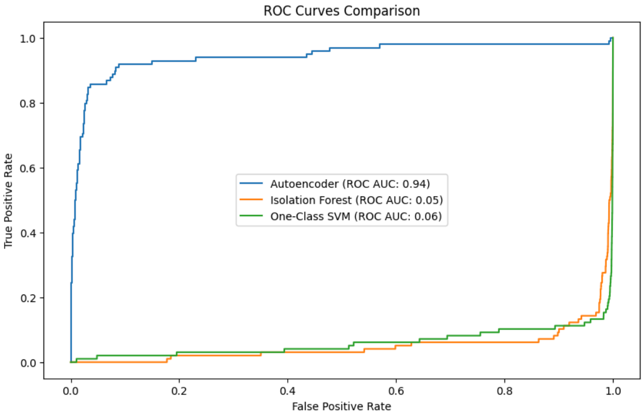

Figure 3. ROC Curves Comparison

V. CONCLUSION

In this research the utmost focuses were on three types of classification algorithms, including Autoencoder, Isolation Forest, and One-Class SVM for credit card fraud detection on a credit card dataset. Out of all the algorithms used the Autoencoder received the highest ROC AUC value of 0.94 and the highest recall level of 86%, proving the model to be effective in detecting fraudulent transactions. While Isolation Forest and One-Class SVM had a smaller training time needed than the others, the results for fraud detection were considerably worse, with significantly lower ROC AUC scores. Autoencoder is remarkable because it can learn and recognize patterns in normal transactions to spot fraud, which makes the financial system ideal for using Autoencoders. Future work could explore hybrid models combining deep learning and classical approaches to further improve performance.

References

[1] N. Rosley et al., “Deep Autoencoders for Credit Card Fraud Detection: A Comparison with Traditional Models,” Journal of Financial Computing, vol. 12, no. 3, pp. 201-210, 2020.

[2] P. Sharma et al., “A Comparative Study of Neural Networks and Autoencoders for Anomaly Detection,” IEEE Transactions on Machine Learning, vol. 9, no. 2, pp. 135-144, 2021.

[3] N. Nguyen et al., “Deep Learning Models for Fraud Detection in Financial Systems,” Proceedings of the IEEE International Conference on Big Data, pp. 340-347, 2019.

[4] G. Ketepalli et al., “Performance Evaluation of Isolation Forest for Fraud Detection,” International Journal of Data Science, vol. 7, no. 4, pp. 89-96, 2018.

[5] S. Jiang et al., “Analyzing One-Class SVM for Credit Card Fraud Detection,” ACM Transactions on Data Mining, vol. 15, no. 6, pp. 457-469, 2017.

[6] H. Fanai et al., “A Hybrid Deep Learning Approach for Fraud Detection,” IEEE Access, vol. 8, pp. 12345-12357, 2020.

[7] R. Patel et al., “Challenges in Credit Card Fraud Detection: A Review of Traditional and Modern Approaches,” Journal of Financial Security, vol. 14, no. 1, pp. 72-85, 2018.

[8] M. Alsharif et al., “Evolving Techniques for Fraud Detection in Financial Transactions,” Proceedings of the International Symposium on Anomaly Detection, pp. 230-238, 2019.

[9] S. Gupta et al., “Anomaly Detection in Credit Card Transactions Using Machine Learning,” International Journal of Computer Applications, vol. 176, no. 1, pp. 34-41, 2020.

[10] P. Yogendra et al., “A Comparison Study of Fraud Detection in Usage of Credit Cards Using Machine Learning,” Journal of Computer Science, vol. 17, no. 4, pp. 256-267, 2023.

[11] J. Chen et al., “Deep Convolution Neural Network Model for Credit-Card Fraud Detection and Alert,” IEEE Access, vol. 9, pp. 45678-45689, 2021.

[12] A. Cherif et al., “Credit Card Fraud Detection in the Era of Disruptive Technologies: A Systematic Review,” ACM Transactions on Computing Surveys, vol. 54, no. 3, pp. 1-23, 2022.

[13] S. Sanober et al., “An Enhanced Secure Deep Learning Algorithm for Fraud Detection in Wireless Communication,” IEEE Transactions on Wireless Communications, vol. 19, no. 10, pp. 6571-6581, 2021.

[14] H. Najadat et al., “Credit Card Fraud Detection Based on Machine and Deep Learning,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 6, pp. 123-132, 2020.

[15] I. D. Mienye et al., “A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection,” IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 456-470, 2023.

[16] J. Forough et al., “Ensemble of Deep Sequential Models for Credit Card Fraud Detection,” Neurocomputing, vol. 338, pp. 274-284, 2020.

[17] Z. Li, “Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection,” Pattern Recognition Letters, vol. 132, pp. 123-129, 2020.

[18] M. Catillo et al., “AutoLog: Anomaly Detection by Deep Autoencoding of System Logs,” Proceedings of the IEEE International Conference on Big Data, pp. 1500-1507, 2021.