AJIT-e Academic Journal of Information Technology

10.5824/ajite.2025.03.002.x

2025 • Summer – Yaz • Volume – Cilt: 16 • Issue – Sayı: 3

# Advanced Mobile Money Fraud Detection Using CNN-BiLSTM and Optimized SGD with Momentum

**Niamatu Yussif**, *Kwame Nkrumah University of Science and Technology, Department of Computer Science, yussifniamatu@gmail.com,* *0009-0005-4637-8740*

**Kate Takyi,** *Kwame Nkrumah University of Science and Technology, Department of Computer Science, Ph.D., takyikate@knust.edu.gh,* *0000-0002-2016-6169*

**Rose-Mary Owusuaa Mensah Gyening**, *Kwame Nkrumah University of Science and Technology, Department of Computer Science, Ph.D., rmo.mensah@knust.edu.gh,* *0000-0002-8087-5207*

**Samuelson Israel Boadu-Acheampong,** *Kwame Nkrumah University of Science and Technology, Department of Computer Science, samuelsonacheampong@gmail.com,* *0009-0003-9287-5989*

**ABSTRACT**

*The accelerated adoption of mobile money systems has significantly increased fraudulent activity, compromising their security and trustworthiness. This research presents an enhanced method for detecting mobile money fraud by modifying a CNN-BiLSTM model with momentum using Stochastic Gradient Descent (SGD). We computed salient features from transaction data using a pre-processed hybrid CNN-BiLSTM model and trained the model to identify trends in the data that included geographical and temporal aspects. The model performed remarkably using industry-standard testing approaches: an F1 score of0.9928, precision of 0.9927, accuracy of 0.9928, and recall of 0.9929. The proposed model can identify dishonesty and has a low false positive rate. According to the study, the model improves feature selection and incorporates various optimization techniques, making it more flexible and suitable for different mobile money systems.*

**Keywords** : Fraud Detection, Machine Learning, Neural Networks, Stochastic Gradient Descent

# CNN-BiLSTM ve Momentum ile Optimize Edilmiş SGD Kullanılarak Gelişmiş Mobil Para Dolandırıcılığı Tespiti

**ÖZ**

*Mobil para sistemlerinin hızla benimsenmesi, dolandırıcılık faaliyetlerinde önemli bir artışa yol açarak güvenliklerini ve itibarlarını tehlikeye atmıştır. Bu araştırma, Stokastik Gradyan İnişi (SGD) kullanarak momentumla bir CNN-BiLSTM modelini değiştirerek mobil para dolandırıcılığını tespit etmek için gelişmiş bir yöntem sunmaktadır. Önceden işlenmiş bir hibrit CNN-BiLSTM modeli kullanarak işlem verilerinden belirgin özellikleri hesapladık ve modeli coğrafi ve zamansal yönleri içeren verilerdeki eğilimleri belirlemek üzere eğittik. Model, endüstri standardı test yaklaşımlarını kullanarak dikkat çekici bir performans gösterdi: 0.9928'lik bir F1 puanı, 0.9927'luk bir hassasiyet, 0.9928'lık bir doğruluk ve 0.9929'lık bir geri çağırma. Önerilen model, sahtekârlığı tespit edebilir ve düşük bir yanlış pozitif oranına*

**Araştırma Makalesi / Research Article**
**Date Received:** 10.02.2025
**Publishing Date:** 31.08.2025

*sahiptir. Çalışmaya göre, model özellik seçimini iyileştirir ve çeşitli optimizasyon tekniklerini birleştirerek onu daha esnek ve farklı mobil para sistemleri için uygun hale getirir.*

**Anahtar Kelimeler** : Sahtecilik Tespiti, Makine Öğrenmesi, Sinir Ağları, Stokastik Gradyan İnişi

## INTRODUCTION

Given the growing number of people using mobile banking, it becomes vital to identify and stop fraud. Mobile money, which provides convenience and access to banking services through mobile devices, has revolutionized financial transactions, especially in developing nations. The quick expansion of mobile financial services has, however, also resulted in a sharp rise in fraudulent activity, which puts users and service providers at great risk. Banks find it more difficult to spot fraud with more mobile money companies. Unauthorized transactions, identity theft, and social engineering attempts are common forms of fraud in mobile money networks. Due to the high amount of transactions, changing fraud trends, and the unbalanced structure of fraud datasets, where fraudulent cases are far less common than genuine ones, these threats are challenging to identify. Therefor, it is both technically and practically necessary to design reliable and effective fraud detection models. Complex fraud activities are frequently too difficult to detect with traditional rule-based systems. The challenge of traditional fraud detection systems in adjusting to different setups raises the probability of thievery. The study must use fresh approaches if fraud detection is to overcome these difficulties (El Kafhali & Tayebi, 2024: 1-25). This study presents a unique and more exact approach based on robust deep learning algorithms and momentum-optimized stochastic gradient descent (SGD) to detect mobile money fraud (Lokanan, 2023: 1-24). Combining the most potent SGD model with momentum, Convolutional Neural Network (CNN), and the Bidirectional Long Short-Term Memory (BiLSTM) model improves network performance and creates a fraud detection tool (Agarwal et al., 2021: 2552-2560). The two most successful machine learning methods are deep learning and efficient stochastic gradient descent (SGD with momentum). This study presents a unique method using a momentum-optimal SGD and CNN-BiLSTM architecture to identify mobile money offenders. Researchers inability to successfully fight mobile money theft is their neglect of modern approaches such as CNN-BiLSTM and upgraded SGD + Momentum algorithms (Igwesi, 2023). CNNs and BiLSTM provide location information and study temporal correlations, thereby improving fraud detection's speed, accuracy, and data management. Using state-of-the-art deep learning methods, including CNN-BiLSTM structures and momentum SGD implementation, our study aims to enhance the training process and help identify mobile money scams. In machine learning, the SGD optimizer is one of the most straightforward and widely utilized optimization algorithms, especially for applications like financial fraud detection (Yang et al., 2023: 1-26). Our approach involves gathering a dataset and its documentation, standardizing the number of columns, and maintaining the category data to ensure every attribute is regularly scaled (Botchey et al., 2021: 45-56). In this work, CNN-BiLSTM deep learning model together with the Decision Tree Classifier is used to find the most essential characteristics for fraud detection.

## 1. RELATED WORKS

With an emphasis on the application of Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and the Stochastic Gradient Descent (SGD) optimizer improved with momentum, this literature review examines recent developments in mobile money fraud detection. The capacity of these algorithms to identify temporal and spatial trends in transaction data has drawn a lot of attention, making them ideal for identifying complex fraudulent activities. This review outlines the fundamental approaches, points out gaps in the literature, and sugggests fresh lines of inquiry for future research with the goal of enhancing detection accuracy and practical applicability through a methodical analysis of previous works.

The paper aimed to make a system that can spot mobile money scams using deep-learning tools. The RNN-based system did much better than the ANN and LSTM systems separately by a significant amount. Bayesian methods were used to fine-tune the RNN model (El Kafhali & Tayebi, 2024: 1-25).

(Botchey et al., 2021: 45-56) examines the issue of identifying fraudulent activity in mobile money transactions, focusing on developing nations with increased financial inclusion. The authors use synthetic minority oversampling and adaptive synthetic sampling, among other sample techniques, to raise the models' performance.

Discussed in the paper is a fresh approach (Zhang et al., 2020: 25210-25220) presented for better spotting fraud for customers who do not make regular transactions. The NM (Naive Bayes-based Multi-behavior) model, which integrates transaction status, individual user conduct, and group behaviour, significantly enhances detection accuracy. Despite occasional recall shortcomings, the NM model outperforms competing models in accuracy and F1 score. The model reduces the disturbance rate, preventing the incorrect classification of legitimate transactions as fraudulent. Further research aims to improve the model's internet-based updates.

Long short-term memory (LSTM) is employed in this study to create a deep learning model for finding bad financial habits. This model has constantly shown excellent results in forecast tasks over time. The model often does a better job than well-known machine learning methods at finding cases of financial wrongdoing in large datasets. (Alghofaili & Rassam, 2020: 498-516).

According to (Almazroi & Ayub, 2023: 137188-137203), The RXT-J model uses advanced machine learning algorithms to identify online payment fraud, overcoming challenges like imbalanced data and extensive models. It uses SMOTE, Ensemble Autoencoder, and ResNet models for feature recovery and attribute extraction.

This project aims to build a network architecture to detect fraudulent online transactions, especially those involving Zimbabwean institutions. It will do this by studying expenditure trends hidden Markov model and deep neural network anomaly detection (Mbunge et al., 2015: 2319-7064).

Mobile money transfer (MMT) services, controlled by carriers, are growing globally. To combat money laundering, MMT systems should identify fake chains. A new method, Predictive Security Analysis at Runtime (PSA@R), outperforms conventional detection methods with 99.81% precision and 90.18% recall (Zhdanova et al., 2014: 11-20).

Researchers found that when there isn't enough training data, understanding what people are saying may help mobile e-commerce systems make better predictions. This includes help from experts, blacklists, and past dishonest behaviour. Data mining methods raise memory and accuracy rates by taking risk variables out of transaction data (Sun et al., 2021: 1-17).

This study will use the Kaggle IEEE-CIS dataset to test new deep-learning methods for finding fraudulent scams. Using deep neural networks and attention methods to find wrong classifications. It shows a CNN-Bi-LSTM-ATTENTION model about 95% of the time (Agarwal et al., 2021: 2552-2560).

The detection of mobile money fraud has been the subject of numerous studies, few have integrated CNN-BiLSTM architectures with momentum-optimized SGD. Although traditional approaches continue to capturing the complete intricacy of transaction patterns. Further study of hybrid deep learning models that can successfully handle the magnitude and dynamic nature of mobile money fraud is obviously needed.

## 2. METHODOLOGY

The aim of this research is to enhance the detection of mobile money fraud with the implementation of a momentum-based approach and CNN-BiLSTM. Data collection and project success assessment are two critical components. The integration of CNN-BiLSTM with momentum and SGD efficiency techniques can facilitate detection. The study uses an existing dataset to develop a deep learning system for fraud detection. This dataset provided a realistic basis for challenges involving fraud detection by simulating mobile money transfers. Following data conversion and standardization, we used resampling to balance the classes in order to address the significant disparity between fraud and non-fraud cases. The data was divided into training and testing sets. Feature importance was assessed using a Decision Tree to guide model design. We combined CNN and BiLSTM to create a hybrid deep learning model that can identify both short-term and long-term trends. A model's F1 score, accuracy, recall, and ROC curve are frequently assessed by users to determine its ability to detect fraud with minimal false positives. Figure 1 shows the conceptual framework of the designed for the proposed work.

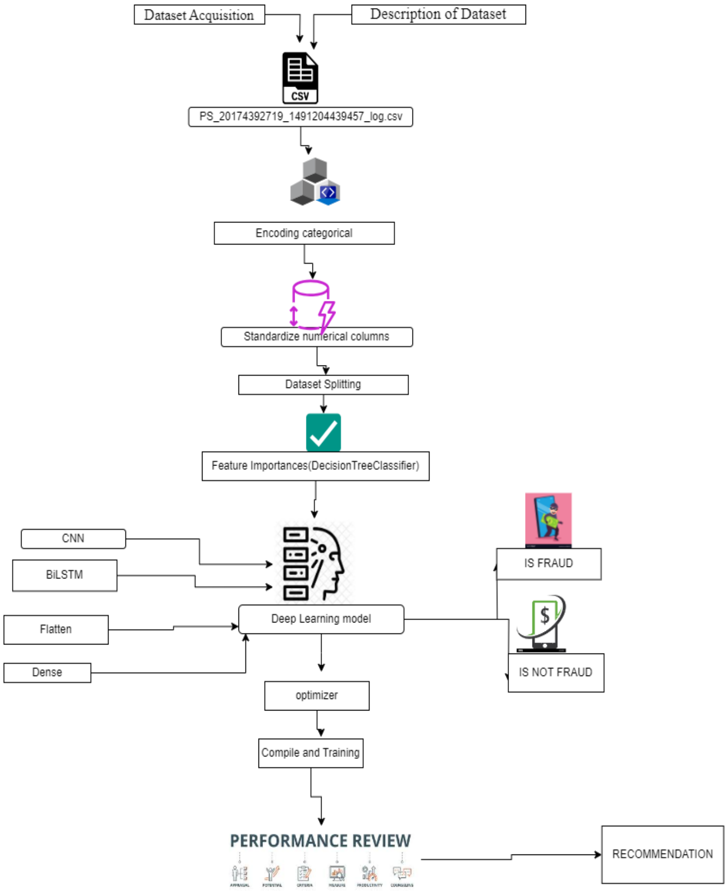


**Figure 1:** The proposed Conceptual Framework

## 2.1. Data Collection

To identify mobile money fraud, the CNN-BILSTM model dataset was created using the Paysim simulator found on the Kaggle dataset named Synthetic Financial Datasets for Fraud Detection (Kaggle, 2023). The dataset includes columns with names such as sorts, quantities, old balance, new balance, fraud, etc. The dataset has 6,362,620 rows and 11 columns, with 6354407 labelled as not fraudulent transaction (0) and 8213 labelled as a fraudulent transaction (1). This dataset was used in the detection of mobile money fraud using the CNN-BILSTM algorithm to improve security in mobile money transactions. It simulates mobile money transactions in a unique way and incorporates fraudulent activity. This dateset made it possible to create and assess sophisticated fraud detection models like CNN-BiLSTM by offering a realistic depiction of financial transactions. The extensive structure of the dataset, which included a vriety of transaction kinds and difficult features, allowed for in-depth testing and analysis of the suggested model. Utilizing this information, the study sought to develop fraud detection and security techniques for mobile banking services. The dataset were chosen because they offer a special combination of simulated fraudulent activities and real-world financial transactions, making them the perfect setting for testing and training sophisticated fraud detection models like CNN-BiLSTM.

### 2.2. Encoding Categorical

The Momentum SGD optimization and CNN-BiLSTM scam detection for mobile money were enhanced through categorical column encoding, requiring sci-kit-learn LabelEncoders to convert category data into numbers. The "type" section displays transaction types like cash-in, cash-out, DEBIT, PAYMENT, or TRANSFER. "nameDest" and "nameOrig" indicate trade sender and receiver. The 'fit_transform' function assigned a LabelEncoder object to each category column, with a different number assigned to each group. Machine learning techniques were employed to speed up data processing.

### 2.3.Standardize Numerical

Algorithms like CNN-BiLSTM and Momentum performed better against mobile money frauds when using consistent number fields. For this, the study turned to the Scikit-learn StandardScaler. By setting the mean to 0 and the standard deviation to 1, it is possible to scale numerical data. When there was no bias, and each attribute had an equal probability of progressing the model, the fraud detection approach performed better.

### 2.4. Dataset Balancing (Resample)

The research identifies mobile money theft using CNN-BiLSTM and Momentum-optimized SGD. Due to a discrepancy in the datasets' classes, the SGD was adjusted. When looking for solutions to societal issues, minorities are given top priority. The majority group is more likely to be a member of both the legitimate and dishonest categories in the sample. Consistently distributed phonetic samples may be obtained by resampling using Scikit-learn. The algorithm can more easily detect unusual activity with an even collection.

### 2.5.K-Fold Cross Validation

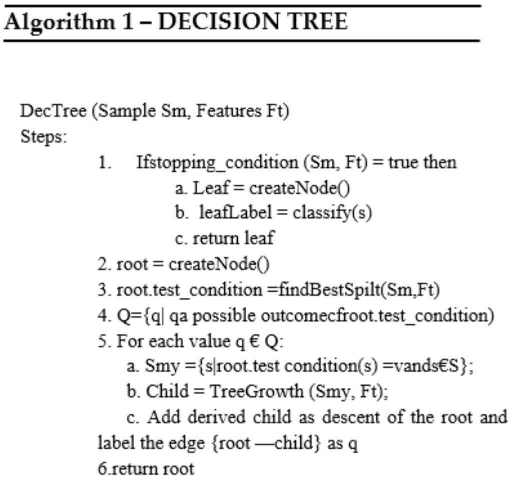
Applying the K-Fold cross validation method with five folds brings up the most accurate model evaluation. Stratified K-Fold was used for the class distribution of all folds to be preserved. Each fold is used for both training and testing, but in the process, a different set is achieved. Training data was used to train a CNN-BiLSTM model, and the model was then improved using a modified SGD with the momentum method. The results of each fold were collected to be later compared to the other folds and were used to measure the extent of the developed model’s new capabilities. This same procedure was repeated for the next five origianl folds until the whole dataset was covered.

### 2.6. Training and Testing Set

CNN-BiLSTM is a model used to detect mobile money fraud by analyzing attributes and behaviours. It is easily understandable and verifiable. The model uses Scikit-learn to train and test the model., This method expands the applicability and stability of the model, enabling the identification of scammers using mobile money.

### 2.7. Feature Importances (Decision Tree Classifier)

Decision Tree Classifier was used for feature importance assessment to improve CNN-BiLSTM-based mobile money fraud detection. This approach identified the dataset's fraud detection strengths—first, a Decision Tree Classifier assessed feature significance. X indicates features, and y fraud/non-fraud was used to train the classifier. The relevance of each feature was determined by the trained decision tree classifier using the 'feature importances_' option. Lower-ranked characteristics related to fraud detection were judged to be the most important. Significant ratings were shown in the end, and feature importance was arranged for every characteristic. This made clear the standards the model used to identify mobile money fraud. Investigation of feature values allowed them to identify dishonesty. The pseudo-code for adopted the decision tree approach is depicted in Figure 2 (Hambali Moshood, n.d.)



```
Algorithm 1 – DECISION TREE

DecTree (Sample Sm, Features Ft)
Steps:
        1.   Ifstopping_condition (Sm, Ft) = true then
                a. Leaf = createNode()
                b.  leafLabel = classify(s)
                c. return leaf
        2. root = createNode()
        3. root.test_condition =findBestSpilt(Sm,Ft)
        4. Q={q| qa possible outcomecfroot.test_condition)
        5. For each value q € Q:
            a. Smy ={s|root.test condition(s) =vands€S};
            b. Child = TreeGrowth (Smy, Ft);
            c. Add derived child as descent of the root and
        label the edge {root —child} as q
        6.return root
```

**Figure 2:** Algorithm for Decision Tree classifier

### 2.8. Deep Learning Model

The deeep learning model for Advanced Mobile Money Fraud Detection was built on a sequential architecture. İt started with a Conv1D layer that localized features from transaction sequences, then a MaxPooling1D layer reduced the dimensionality. The application of BatchNormalization was meant to make training more stable and faster. A Bidirectional wrapper around the BiLSTM layer helped in getting the inputs from both directions along the time axis. Dropout was used here to eliminate overfitting. The network also had a Flatten layer that transformed the features into a vector format, then it was followed by two fully connected Dense layers that were used for capturing complex patterns. It was trained with an optimized SGD optimizer with momentum to accelerate convergence.

### 2.9. CNN Layer

To capture important local temporal patterns in the sequential mobile money transaction data, the model added a Conv1D layer with 64 filters and a kernel size of 3. This layer, using the ReLU activation, re- mapped the raw input signals such as trasaction time gaps and amounts into feature maps that were more informative and highlighted potential critical indicators of fraud. The input shape was specified to correspond to the preprocessed training data that allowed the same data format to be passed to the following layers. This convolutional method enabled the system to directly learn complex fraud-related behaviours without the need for manual feature engineering, which is very important for the fast-changing and evolving nature of mobile money fraud detection cases.

### 2.10. MaxPooling Layer

The research utilised a MaxPooling1D layer of size 2 for the purpose of lowering the dimensionality of the convolutional feature maps. This pooling operation targeted the most prominent signals related to fraud while eliminating the less relevant noise, thus enabling better computation efficiency. By aggregating adjacent features, it contributed to the steadiness of detection accuracy across different transaction patterns. This phase therefore was crucial in feature extraction compression, avoiding overfitting, and allowing the following BiLSTM layer to effectively read the most important time-series fraud hints hidden in the transaction sequences.

### 2.11. Batch Normalization

Following the pooling, the investigation utilised Batch Normalisation to standardize the featuer maps, thus helping to solve the issue of internal covariate shift. This normalisation operation not only made training more stable and quicker, but also ensured that each feature across the batches had the same mean and variance, which is very important for financial data that is susceptible to scale changes. İn our highly sophisticated mobile money fraud detection pipeline, this layer played a significant role in improved convergence when we combined it with Optimised SGD with Momentum, thus allowing the model to handle various fraud patterns, different types of transactions, and the changes in transaction behaviours that occur in different seasons effectively.

### 2.12. BiLSTM Layer

Researchers conducted experiments with a Bidirectional wrapper around an LSTM layer of 64 units and return sequences = True, which allowed the model to understand transaction sequences from both forward and backwards directions. This dual-context approach enable the model to not only understand simple dependencies but also more complicated ones, such as regurently fraud and transaction reversals. By jointly considering the past and future context, the BiLSTM boosted the model's capability to spot tricky fraud patterns that are usually overlooked by unidirectional models. This move was essential to getting the model more temporally aware, which made it fit the nature of the continously changing scam cases in the mobile money sector perfectly.

### 2.13. Dropout

In order to avoid overfitting, we also used a Dropout layer which has a dropout rate of 0.2. This layer, throughout the training, randomly switched off 20% of the neurons, thus reducing the co-adaptation of features and helping the model to be more robust when faced with fraud cases that it has not seen before. Dropout, in the case of mobile money fraud detection, ensured that the network learned the fraud representations which were spread out rather than actually memorizing the specific transactions, therefore, it was able to maintain high performance even if the fraud strategies changed.

### 2.14. Flatten Layer

After going through the sequential and convolutional layers, we used a Flatten layer to change the multi-dimensional feature maps into one vector. This operation not only facilitated the dense layers by collapsing the spatial and temporal fraud cues in to a single format but also made it easier for the model to move from extracting temporal features to high-level fraud risk scoring. Flattening was a key step in connecting sequential learning with fully connected classification.

### 2.15. Dense Layer

To conclude, this research also employed two Dense layers: initially with 64 neurons and ReLU activation to extract more fraud features and secondly with a single neuron and sigmoid activation for the final fraud probability output. The dense layers carried out intricate nonlinear operations to distinguish fraud cases from non-fraud ones. The Optimised SGD with Momentum facilitated efficient and stable convergence, hence the model was able to attain accurate fraud detection by recognizing inconspicuous but vital transaction patterns.

### 2.16. Optimizer

The focus of the study is to enhance mobile money fraud detection using a momentum-based stochastic gradient descent (SGD) model using a CNN-BiLSTM optimizer. The optimizer accelerated and stabilized the training process, adjusting model weights and learning rate of 0.01. The momentum coefficient of 0.9 and learning rate of 0.01 improved the consistency and fluidity of parameter updates. This strategy allowed the model to understand complex patterns associated with fraudulent mobile money transactions quickly. Researchers may modify the momentum and learning rate to optimize the model's fraud detection capabilities.

### 2.17. Compiling and Training

CNN-BiLSTM models using the TensorFlow backend and Keras framework were trained to enhance the detection of mobile money fraud. We used the SGD with momentum optimizer to improve the model by measuring the difference between the predicted and real class labels. The performance of the model was assessed using accuracy measures. The fit approach used data from all batch sizes of 64 to train the algorithm. The train, test and validation data obtained is used to evaluate the performance of the performance. Researchers wanted to make mobile money transfers more secure; therefore, they worked to make the model better at detecting fraud.

### 2.18. Performance Metrics

This study examines key performance indicators in two stages: training and testing. The evaluation metrics Accuracy, Precision, recall and F1-score, False Positve Rate (FPR) and False Negative Rate (FNR) are mostly considered in measuring the performance of a model. The values from True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) and N (where N is the sum of (TP + TN + FP + FN) are the metrics used to calculate these measures. The different metrics are as follows:

a) F1 score: $2 * \frac{\mathbf{Prec*Rec}}{\mathbf{Prec + Rec}}$ (Chatterjee & Namin, 2019: 227-32)

b) Recall: $\frac{TP}{\mathbf{TP + FN}}$ (Chatterjee & Namin, 2019: 227-32)

c) Precision: $\frac{TP}{\mathbf{TP + FP}}$ (Gibson et al., 2020: 187914-187932)

d) Accuracy: $\frac{(\mathrm{TP + TN})}{\mathrm{N}}$ (Gibson et al., 2020: 187914-187932)

e) False Positive Rate: $\frac{(\mathrm{FP})}{(\mathrm{FP+TN})}$

f) False Negative Rate: $\frac{(\mathrm{FN})}{(\mathrm{FN+TP})}$

These performance metrics are considered in the study and used to evaluate the effectiveness proposed model with Batch Normalisation for detecting mobile money fraud, and any necessary adjustments are made to increase the model's performance.

## 3. RESULTS AND ANALYSIS

The paper explores a method for detecting mobile money fraud using CNN-BiLSTM and Momentum-optimized SGD. It emphasizes the model's usefulness in detecting fraud and its impact on developing safe digital financial ecosystems. The dataset, which includes transaction details, account names, balances, and binary fraud flags, is analysed, highlighting the need for dataset-balancing strategies. The findings will guide future efforts in fraud detection.

### 3.1.Results of Balance the Dataset (Resample)

The "isFraud" field of the dataset shows a way to do resampling that is meant to help find more mobile money scams. The class difference in the dataset indicates that almost all the trades are real, while only a tiny number are fake. After the resampling method, there were no clear differences between fake activities and those that were not. This showed that the distribution was fair. The fair distribution of the dataset makes it easier for the model to find mobile money transfer networks because it makes the dataset more representative and less biased. Results obtained from the data balancing process are shown in Table 1.

**Table 1:** The results obtained from the balancing of the dataset

| CLASS | FROM | TO |
|---|---|---|
| IS NOT FRAUD (0) | 6354407 | 6354407 |
| IS FRAUD (1) | 8213 | 6354407 |

### 3.2. Results of Feature Importance Selection (Decision Tree Classifier)

The "newbalanceOrig" which is a core component of the Decision Tree approach is crucial and beneficial for identifying mobile money fraud. Some attributes, such as "type" and "step," are more significant than others. The system achieved a memory score 1.0 due to its accurate identification of all fraudulent transactions. It may be worth contemplating for legal and illicit enterprises due to its durability and favourable F1 rating. The system became perplexed while attempting to sort through the 207 authentic offers. Figure 3 shows the results of feature improvements while Figure 4 shows the ROC curve for the featured result. Figure 5 shows the bar graph of the evaluation Metric score for the feature improvements.

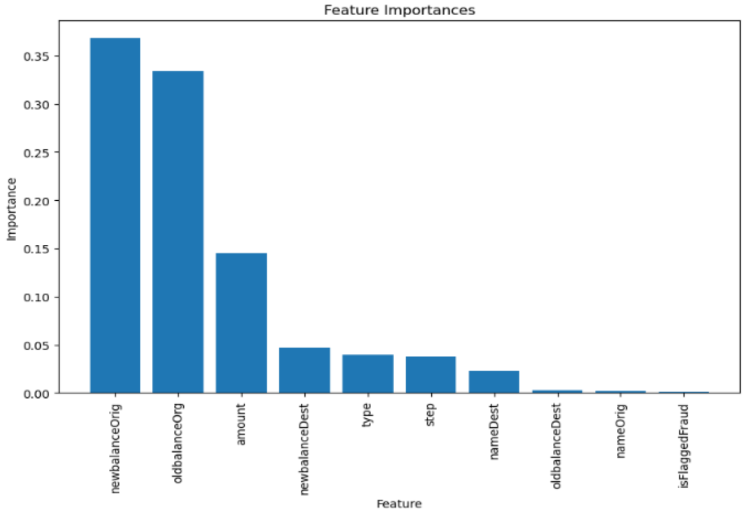


**Figure 3:** Bar graph representation of feature improvements against its importance

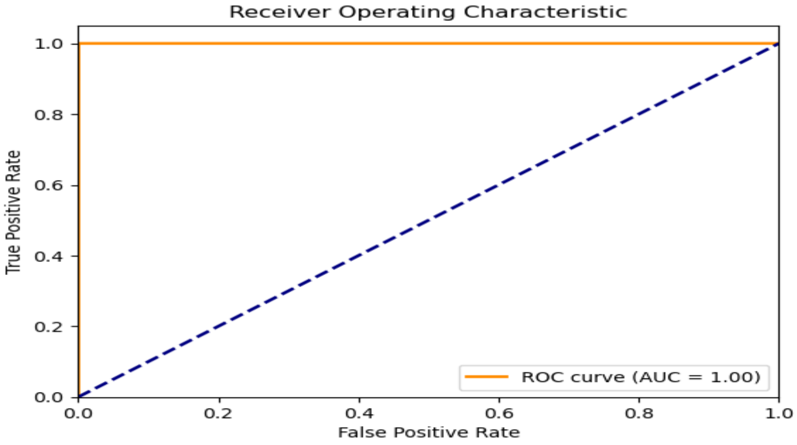


Figure 4: ROC curve of feature improvements

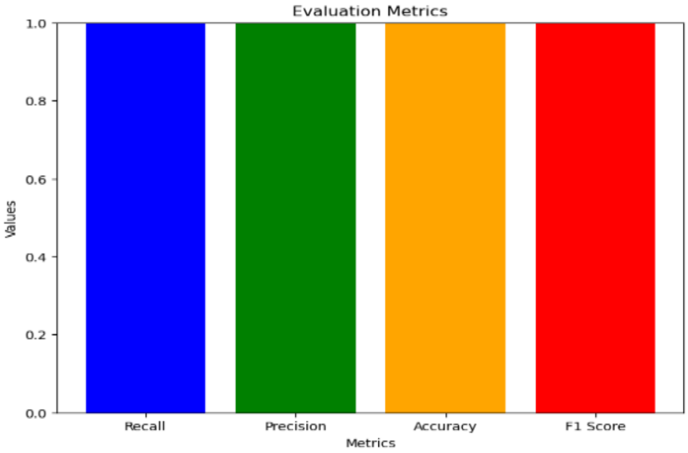


**Figure 5:** Evaluation Metric Score

### 3.3. Test Results with Variations of Epoch and Batch Size

Optimal SGD with Momentum is used in this work to test how well the CNN-BiLSTM model can find mobile money plans. For the training, 5-fold cross validation was chosen among the the range of 5 to 10 to reduce the computational costs, in addtion to its quick iteration. Furthermore 5-fold cross validation gives a little higher variance with respect to performance estimation which is acceptable in the case of improved speed exchange, which is the expected goal of the training.

Fold 1: Loss was gradudually getting smaller by each subsequent batch, eventually getting to 0.0464 in the 10th epoch. The peak of accuracy was 99.50%, while the validation accuracy went up to 99.55%, hence a clear sign of the better learning stability from the increment of the training.

Fold 2: The initial loss was 0.6500, but the error descended to 0.0535 by epoch 10. The accuracy became 99.41% at best with the validation accuracy using 99.55%, leading to the fact that it was a learning and generalisation process that was efficient.

Fold 3: The journey started with a loss of 0.6302 and got down to 0.0455 by the 10th epoch. The validation accuracy hit the highest bar of 99.8%, which suggests that the model’s performance has grown significantly with the increase of the batch.

Fold 4: By having obtained 0.6186 as initial loss, the model improved to 0.0456. The accuracy grew to 99.55%, and the validation accuracy maintaind stability over 99.00%, hence the process of learning dynamics was resilient across all batch sizes.

Fold 5: The last part of the loss climbed down from 0.6562 to 0.0624, and the final accracy was 99.19%. The figures for validation accuracy were consistently above 99.00%, even at the highest of 99.55, which showed the strenghth of the model in detecting fraud.Table 2, Table 3, Table 4, Table 5 and Table 6 shows the results with variations of epoch and batch size for fold 1, fold 2, fold 3, fold 4 and fold 5 respectively. Figure 6 and Figure 7 shows the training and validation loss of the epoch and batch size and accuracies obtained resepectively.

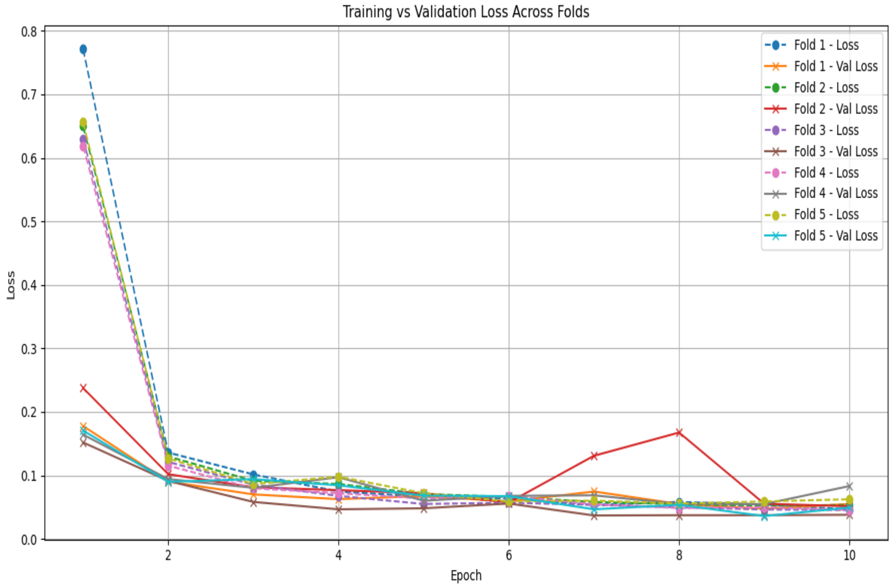


**Figure 6:** The training and validation loss with respect to epoch and batch size

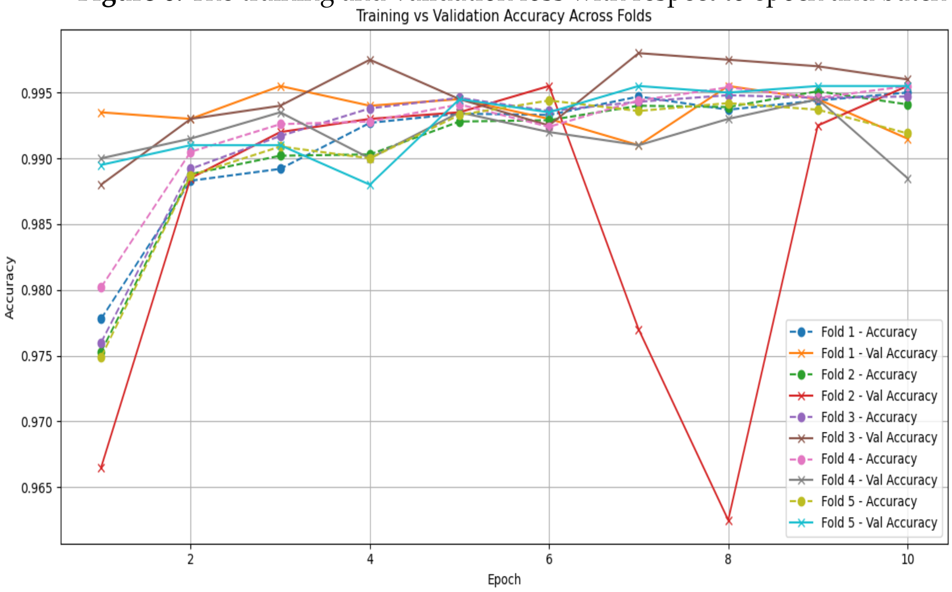


**Table 2:** The results with variations of epoch and batch size

| No of Batches | Loss | Accuracy | Val_Loss | Val_Accuracy |
|---|---|---|---|---|
| 1 | 0.7713 | 0.9778 | 0.1777 | 0.9935 |
| 2 | 0.1359 | 0.9883 | 0.0902 | 0.9930 |
| 3 | 0.1015 | 0.9892 | 0.0701 | 0.9955 |
| 4 | 0.0746 | 0.9927 | 0.0626 | 0.9940 |
| 5 | 0.0666 | 0.9934 | 0.0675 | 0.9945 |

| | | | | |
|---|---|---|---|---|
| 6 | 0.0635 | 0.9933 | 0.0590 | 0.9930 |
| 7 | 0.0536 | 0.9947 | 0.0746 | 0.9910 |
| 8 | 0.0582 | 0.9937 | 0.0550 | 0.9945 |
| 9 | 0.0541 | 0.9944 | 0.0474 | 0.9945 |
| 10 | 0.0464 | 0.9950 | 0.0550 | 0.9915 |

**Table 3:** The results with variations of epoch and batch size for fold 2

| No of Batches | Loss | Accuracy | Val_Loss | Val_Accuracy |
|---|---|---|---|---|
| 1 | 0.6500 | 0.9753 | 0.2376 | 0.9665 |
| 2 | 0.1298 | 0.9888 | 0.1019 | 0.9885 |
| 3 | 0.0914 | 0.9902 | 0.0812 | 0.9920 |
| 4 | 0.0865 | 0.9903 | 0.0768 | 0.9930 |
| 5 | 0.0706 | 0.9928 | 0.0719 | 0.9935 |
| 6 | 0.0662 | 0.9929 | 0.0574 | 0.9955 |
| 7 | 0.0582 | 0.9940 | 0.1309 | 0.9770 |
| 8 | 0.0546 | 0.9939 | 0.1675 | 0.9625 |
| 9 | 0.0521 | 0.9951 | 0.0554 | 0.9925 |
| 10 | 0.0535 | 0.9941 | 0.0514 | 0.9955 |

**Table 4:** The results with variations of epoch and batch size for fold 3

| No of Batches | Loss | Accuracy | Val_Loss | Val_Accuracy |
|---|---|---|---|---|
| 1 | 0.6302 | 0.9760 | 0.1519 | 0.9880 |
| 2 | 0.1212 | 0.9892 | 0.0913 | 0.9930 |
| 3 | 0.0854 | 0.9917 | 0.0582 | 0.9940 |
| 4 | 0.0677 | 0.9938 | 0.0466 | 0.9975 |
| 5 | 0.0550 | 0.9946 | 0.0482 | 0.9945 |
| 6 | 0.0570 | 0.9936 | 0.0557 | 0.9925 |
| 7 | 0.0537 | 0.9943 | 0.0369 | 0.9980 |
| 8 | 0.0497 | 0.9948 | 0.0373 | 0.9975 |
| 9 | 0.0463 | 0.9946 | 0.0373 | 0.9970 |
| 10 | 0.0455 | 0.9947 | 0.0380 | 0.9960 |

**Table 5:** The results with variations of epoch and batch size for fold 4

| No of Batches | Loss | Accuracy | Val_Loss | Val_Accuracy |
|---|---|---|---|---|
| 1 | 0.6186 | 0.9802 | 0.1642 | 0.9900 |
| 2 | 0.1154 | 0.9905 | 0.0942 | 0.9915 |
| 3 | 0.0797 | 0.9926 | 0.0804 | 0.9935 |
| 4 | 0.0721 | 0.9928 | 0.0971 | 0.9900 |
| 5 | 0.0635 | 0.9941 | 0.0601 | 0.9935 |
| 6 | 0.0675 | 0.9924 | 0.0678 | 0.9920 |
| 7 | 0.0546 | 0.9944 | 0.0686 | 0.9910 |
| 8 | 0.0486 | 0.9954 | 0.0552 | 0.9930 |
| 9 | 0.0500 | 0.9946 | 0.0548 | 0.9945 |
| 10 | 0.0456 | 0.9955 | 0.0832 | 0.9885 |

**Table 6:** The results with variations of epoch and batch size for fold 5

| No of Batches | Loss | Accuracy | Val_Loss | Val_Accuracy |
|---|---|---|---|---|
| 1 | 0.6562 | 0.9749 | 0.1707 | 0.9895 |
| 2 | 0.1267 | 0.9887 | 0.0898 | 0.9910 |
| 3 | 0.0864 | 0.9909 | 0.0941 | 0.9910 |
| 4 | 0.0979 | 0.9900 | 0.0844 | 0.9880 |
| 5 | 0.0719 | 0.9933 | 0.0682 | 0.9945 |
| 6 | 0.0590 | 0.9944 | 0.0673 | 0.9935 |
| 7 | 0.0603 | 0.9936 | 0.0466 | 0.9955 |
| 8 | 0.0556 | 0.9942 | 0.0537 | 0.9950 |
| 9 | 0.0590 | 0.9937 | 0.0359 | 0.9955 |
| 10 | 0.0624 | 0.9919 | 0.0494 | 0.9955 |

### 3.4. Results Of The K-Folds Validation

The model's performance was consistently strong and impressive in all of the K-Folds validation outcomes. The prototype from the first fold was marked by an accuracy of 99.15% and a loss of 0.0560. The second fold had further ascended in its performance as the associated loss was down to 0.0514 with an improved accuracy rate of 99.55%. Following this pattern, the third fold was better than the previous one since the loss was 0.0360, and the accuracy was the highest of 99.60%. An insignificant drop was noticed in the fourth fold, leading to a 0.0832 loss and 98.85% accuracy. The last fold, like the second one, obtained the same level of accuracy at 99.55% with 0.0495 loss. Table 7, shows the loss and accuracy of each fold. Figure 8 shows the loss and accuracy concerning each fold.

**Table 7:** The loss and accuracy with respect to each fold

| No of Folds | Loss | Accuracy |
|---|---|---|
| 1 | 0.0560 | 0.9915 |
| 2 | 0.0514 | 0.9955 |
| 3 | 0.0360 | 0.9960 |
| 4 | 0.0832 | 0.9885 |
| 5 | 0.0495 | 0.9955 |

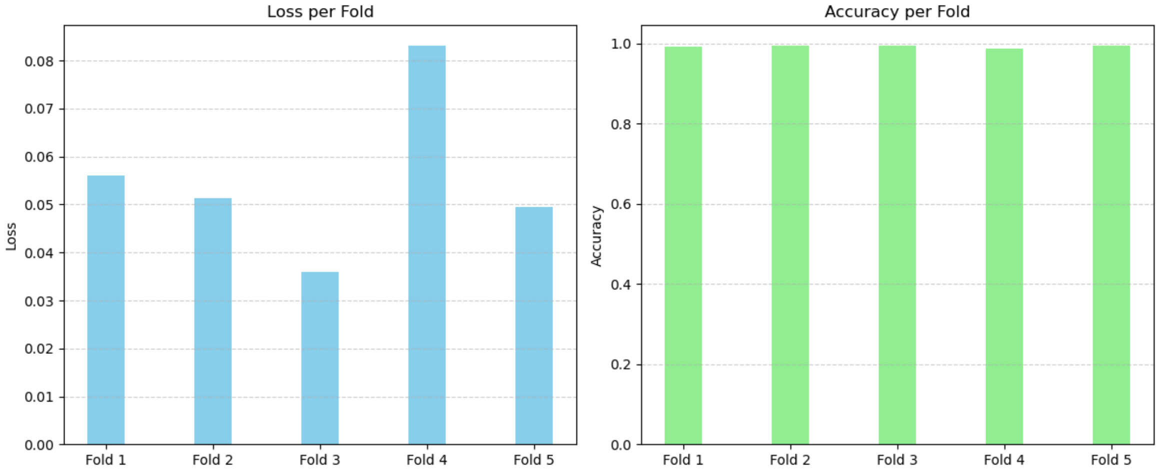


**Figure 8:** The loss and accuracy with respect to each fold

### 3.5. Results of the CNN + Bi-LSTM

For the detection of high-tech mobile money fraud, the CNN-BiLSTM model set a benchmark for the evaluation metrics. İt had recorded an average recall score of 0.9929, a precision score of 0.9927, and and an accuracy of 0.9928. The average F1 score also conferred with it with the highest value of 0.9928, thus an excellent balance of precision and recall was achieved. The fasle positive rate was very low at 0.0073, and the fasle negative rate was 0.0071. The model made no mistakes in classifiying 1,261,877 fraud cases (TP) and 1,261, 550 non-fraud cases (TN). Moreover, it was able to detect 1,261,877 fraud cases (TP) and 1,261,550 situations of non-fraud (TN) with 9,300 fasle positives and 9,036 false negatives, respectively. Table 8 summarizes the results obtained from the CNN + BILSTM model. Figure 9 shows the ROC curve of the CNN + BILSTM model.

**Table 8:** Summary of the proposed CNN + Bi-LSTM model

| **MODEL EVALUATION** | **MODEL RESULT** |
|---|---|
| Recall Score | 0.9929 |
| Precision Score | 0.9927 |
| Accuracy Score | 0.9928 |
| F1 Score | 0.9928 |
| FPR | 0.0073 |
| TN | 1,261,550 |
| FP | 9,300 |
| FN | 9,036 |
| TP | 1,261,877 |

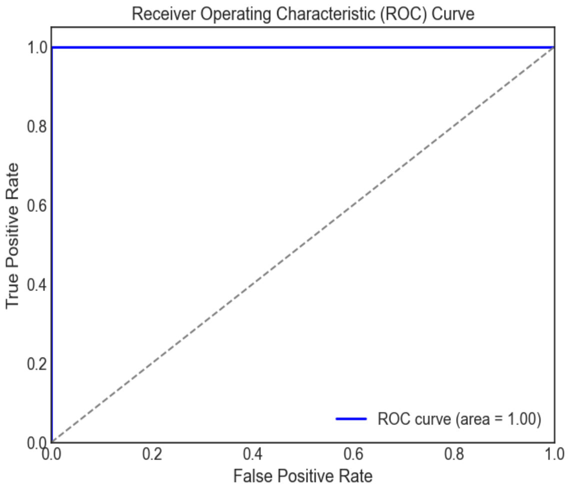


**Figure 9:** ROC curve of the CNN + BILSTM model with Area under the Curive (AUC)

### 3.6. Model Prediction

The model's performance was evaluated on the dataset for validation, tessting and training.. The research includes a wide variety of model success metrics for the model. Table 9 shows the classification report. Figure 10 shows the Confusion Matrix of the CNN + BILSTM model while Figure 11 depicts the bar graph of the CNN + BILSTM, with the value for each evaluation metric. Figure 12 shows the bar graph of the classification report. The CNN-

BiLSTM model is a great way to spot mobile money scams.It has a 99% success rate in class 1 and a 99% success rate in class 0. Because deals are legal or unfair, it is easy to distinguish

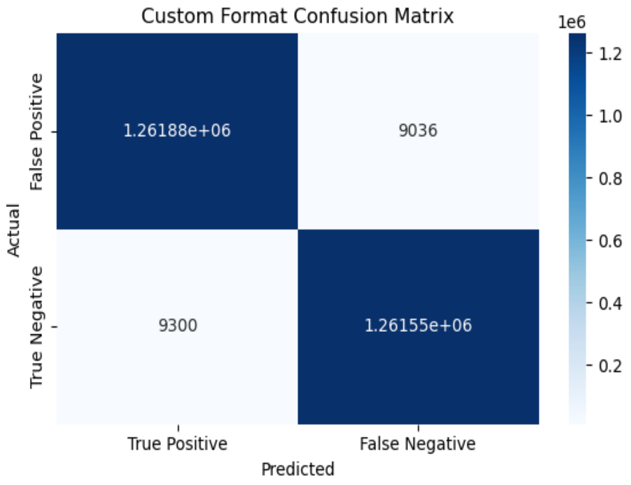


**Figure 10:** Confusion Matrix of the proposed CNN + BILSTM model

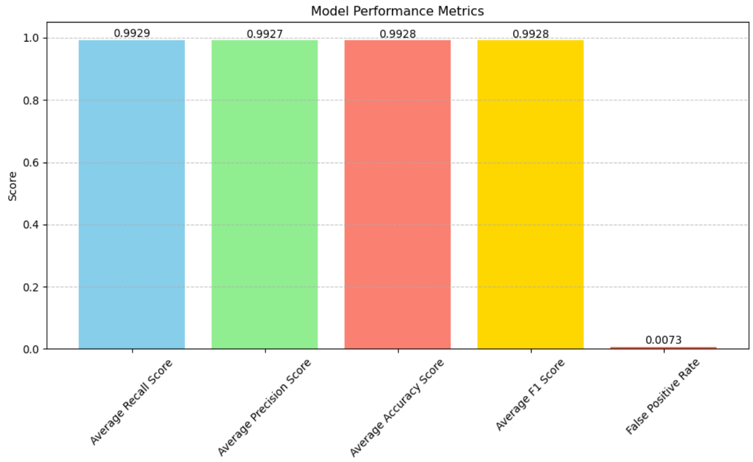


**Figure 11:** Bar graph showing the evaluation metric score for the proposed CNN + BILSTM model

**Table 9:** The classification report

| CLASS | PRECISION | RECALL | F1-SCORE | ACCURACY |
|---|---|---|---|---|
| IS NOT FRAUD (0) | 0.99 | 0.98 | 0.99 | 0.99 |
| IS FRAUD (1) | 0.98 | 0.99 | 0.99 | 0.99 |

between real and fake ones. Customers trust financial companies and their services more because this technology can easily find mobile money theft.

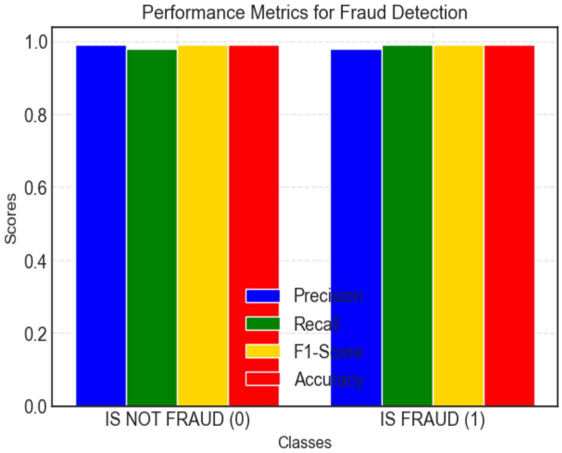


**Figure 12:** The classification report

The CNN-BiLSTM model effectively detects mobile money fraud with a 99% accuracy rate, distinguishing between authorized and fraudulent transactions. Its recall rate of 98% demonstrates its effectiveness in preventing false negatives. This system is particularly beneficial when prompt and accurate detection of fraudulent behaviour is crucial, as it boosts customer trust in financial institutions.

### 3.7.Model Comparison

Predicting fraud in mobile money transactions with machine learning: how sampling methods affect a sample that isn't fair (Botchey et al., 2021: 45-56). Revised: Detection of financial fraud in the healthcare sector via the use of machine learning and deep learning techniques has been retracted (Networks, 2023: 1-1). Methods for Detecting Fraudulent Digital Payments in the Digital Era and Various Industries (Chang et al., 2022: 1-31). Table 10 shows the Metric performance of the applied model with their respective datasets.

**Table 10:** Metric performance of the proposed applied model compared to other detection algorithms with their respective datasets.

| MODEL | ACCURACY | PRECISION | RECALL | F1-SCORE |
|---|---|---|---|---|
| RF (Networks, 2023: 1-1) | 98% | 97% | 97% | 97% |
| DT (Chang et al., 2022: 1-31) | 98% | 92% | 83% | 100% |

| | | | | |
|---|---|---|---|---|
| ADASYN (Botchey et al., 2021: 45-56) | 98% | 0.84% | 99% | 1.68% |
| CNN+ BiLSTM (proposed model) | 99% | 99% | 99% | 99% |

Using stochastic gradient descent with momentum for optimization, the suggested CNN-BiLSTM model showed good classification performance, attaining 99% accuracy, 99% precision, 99% recall, and 99% F1-score. These outcomes outperform related studies using the same dataset and assumption produced by other machine learning models. This model employs machine learning methods to identify fraudulent online payment transactions (Almazroi & Ayub, 2023: 137188-137203). Using machine learning techniques to predict fraudulent mobile money transactions (Lokanan, 2023: 1-24). Detecting fraud with the Kolmogorov-Arnold Network (Gislain & Yurievic, 2025: 1-14). Table 11 shows the Metric performance of the applied model.

**Table 11:** Metric performance of the proposed applied model compared to other detection algorithms with the proposed dataset.

| MODEL | ACCURACY | PRECISION | RECALL | F1-SCORE |
|---|---|---|---|---|
| RF (Lokanan, 2023: 1-24) | 0.89 | 0.96 | 0.80 | 0.87 |
| RXT-J (Almazroi & Ayub, 2023: 137188-137203) | 96% | 96% | 98% | 97% |
| KAN (Gislain & Yurievic, 2025: 1-14) | 97% | 97% | 97% | 97% |
| CNN+ BiLSTM (proposed model) | 99% | 99% | 99% | 99% |

With a 99% success rate, the CNN-BiLSTM model is better than most modern ways of finding mobile money schemes. Using convolutional neural networks and bidirectional long short-term memory (BiLSTM) layers, this model is better than all the others. These include Decision Tree (DT), ADASYN, RXT-J, Kolmogorov-Arnold Network (KAN) , and Random

Forest (RF). There are almost no fake wins or rejections produced by deep learning systems. These results show they might make banking institutions safer against mobile money theft.

For sophisticated mobile money fraud detection, the CNN-BiLSTM and Optimised SGD with Momentum model produced exceptional results for a number of reasons. Subtle irregularities in sequential data were captured by CNN layers, which successfully retrieved local transactional patterns. BiLSTM layers improved pattern identification for both fraudulent and genuine acts by learning intricate temporal relationships in both past and future contexts. By smoothing gradient updates, the Optimised SGD with Momentum sped up convergence and prevented local minima, resulting in accurate and steady learning. When combined, these elements yeielded a balanced precision of 0.9927, which decreased false alarms, and a high recall of 0.9929, which is essential for identifying fraud efforts. Strong overall performance is reflected in the outstanding F1 score of 0.9928. This architecture successfully handled imbalanced data issues, resulting in a low false positive rate (0.0073) and balanced classification accuracy for fraud and non-fraud classes, which makes it ideal for real-world mobile money fraud detection.

### **3.8.** Limitations

It's critical to recognize the PaySim dataset's limitations even though it offers a useful approximation of mobile money transactions. İt might not fully represent the intricacy, volatility, and variety of actual financial behavior because it is a simulated dataset. Because the are predicated on set assumptions, the fraudulent patterns produced might not accurately represent the changing tactics employed by actual fraudsters. Furthermore, simulated data is typically more organized and tidy than real data, which could cause model performance to be exaggerated. These elements imply that even though the outcomes are encouraging, additional verification utilizing actual mobile money data is required to validate the model's resilience and capacity for generalization.

## 4. CONCLUSION AND RECOMMENDATION

The study demonstrates the effectiveness of CNN-BiLSTM and optimized SGD + Momentum in detecting mobile money fraud. The model has a recall score of 0.9929, a Precision score of 0.9927, and an overall accuracy of 0.9928in distinguishing between legal and fraudulent transactions. It correctly recognized and rejected occurrences, indicating that advanced deep learning techniques like CNN-BiLSTM architecture and improved SGD + Momentum optimization approaches can mitigate mobile money fraud.

### 4.1. Conclusion

This study uses momentum-optimized stochastic gradient descent (SGD) and CNN-BILSM to detect mobile money theft. The CNN-BILSM model achieves high generalisability and convergence by identifying patterns in time and space and optimizing its momentum and learning rate components. The model outperforms older models like ADASYN, Decision Tree (DT), Random Forest (RF), Kolmogorov-Arnold Network (KAN) and RXT-J, with a precision of 0.9927 and a success rate of 0.9928. The model's effectiveness is evaluated using various methods.

### 4.2. Recommendation and Future Works

A new method for detecting mobile money fraud has been developed using CNN-BiLSTM and Optimised SGD plus momentum models. This method aims to enhance real-time fraud detection, improve system transparency, and improve feature engineering. Collaboration between police, IT firms, and institutions can efficiently tackle mobile money fraud. Future research will use innovative methodologies like CNN-BiLSTM and federated learning to enhance detection further in addition to Real-time transaction monitoring systems can also help identify and prevent fraudulent behaviour early.

## REFERENCES

Agarwal, A., Iqbal M., Mitra, B., Kumar, V., & Lal, N.( 2021). "Hybrid CNN-BILSTM-attention based identification and prevention system for banking transactions," *… Essent. OILS J. …*, vol. 8, no. 5, pp. 2552–2560, [Online]. Available: http://www.nveo.org/index.php/journal/article/view/809

Alghofaili, Y., Albattah, A., & Rassam, M. A. (2020). "A Financial Fraud Detection Model Based on LSTM Deep Learning Technique," *J. Appl. Secur. Res.*, vol. 15, no. 4, pp. 498–516, https://doi.org/10.1080/19361610.2020.1815491.

Almazroi, A. A., & Ayub, N. (2023) "Online Payment Fraud Detection Model Using Machine Learning Techniques," *IEEE Access*, vol. 11, no. November, pp. 137188–137203, https://doi.org/10.1109/ACCESS.2023.3339226.

Botchey, F. E., Qin, Z., Hughes-Lartey, K., & Ampomah, K. E.( 2021). "Predicting Fraud in Mobile Money Transactions using Machine Learning: The Effects of Sampling Techniques on the Imbalanced Dataset," *Inform.*, vol. 45, no. 7, pp. 45–56, https://doi.org/10.31449/inf.v45i7.3179.

Chatterjee, M., & Namin, A. S. (2019). "Detecting Phishing Websites through Deep Reinforcement Learning." *Proceedings - International Computer Software and Applications Conference* 2(1): 227–32.

Chang, V., Doan, L. M. T., Di Stefano, A., Sun, Z., & Fortino, G. (2022). "Digital payment fraud detection methods in digital ages and Industry 4.0," *Comput. Electr. Eng.*, vol. 100, pp. 1–31, https://doi.org/10.1016/j.compeleceng.2022.107734.

El Kafhali, S., Tayebi, M., & Sulimani, H.( 2024). "An Optimized Deep Learning Approach for Detecting Fraudulent Transactions," *Inf.*, vol. 15, no. 4, pp. 1–25, https://doi.org/10.3390/info15040227.

Gibson, S., Issac, B., Zhang, L., & Jacob, S. M. (2020) "Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms," *IEEE Access*, vol. 8, pp. 187914–187932, https://doi.org/10.1109/ACCESS.2020.3030751.

Gislain, Z. N. T., & Yurievich, K. E. (2025). Fraud detection using Kolmogorov-Arnold Network, pp. 1-14.

Hambali Moshood, A., "Comparative Analysis of Decision Tree Algorithms for Predicting Undergraduate Students' Performance in Computer Programming".

Igwesi, C. (2023). "Enhancing Authentication and Fraud Detection in Financial Technology," no. December, 2023.

Kaggle. (n.d.). Kaggle datasets: Paysim1. Retrieved [4th August, 2023], from https://www.kaggle.com/datasets/ealaxi/paysim1

Lokanan, M. E. (2023).“Predicting mobile money transaction fraud using machine learning algorithms,” *Appl. AI Lett.*, vol. 4, no. 2, pp. 1–24, https://doi.org/10.1002/ail2.85.

Mbunge, E., Makuyana, R., Chirara, N., & Chingosho, A. (2015). “Fraud Detection in E-Transactions using Deep Neural Networks-A Case of Financial Institutions in Zimbabwe,” *Int. J. Sci. Res.*, vol. 6, no. 9, pp. 2319–7064, https://doi.org/10.21275/ART20176804.

Networks, C. (2023). “Retracted: Financial Fraud Detection in Healthcare Using Machine Learning and Deep Learning Techniques,” *Secur. Commun. Networks*, vol. 2023, pp. 1–1, https://doi.org/10.1155/2023/9758612.

Sun, Q., Tang, T., Chai, H., Wu, J., & Chen, Y. (2021). “Boosting fraud detection in mobile payment with prior knowledge,” *Appl. Sci.*, vol. 11, no. 10, pp. 1–17, https://doi.org/10.3390/app11104347.

Yang, X., Zhang, C., Sun, Y., Pang, K., Jing, L., Wa, S., & Lv, C. (2023) “FinChain-BERT: A High-Accuracy Automatic Fraud Detection Model Based on NLP Methods for Financial Scenarios,” *Inf.*, vol. 14, no. 9, pp. 1–26, https://doi.org/10.3390/info14090499.

Zhang, Z., Chen, L., Liu, Q., & Wang, P. (2020). “A Fraud Detection Method for Low-Frequency Transaction,” *IEEE Access*, vol. 8, pp. 25210–25220, , https://doi.org/10.1109/ACCESS.2020.2970614.

Zhdanova, M., Repp, J., Rieke, R., Gaber, C., & Hemery, B. (2014).“No smurfs: Revealing fraud chains in mobile money transfers,” *Proc. - 9th Int. Conf. Availability, Reliab. Secur. ARES 2014*, pp. 11–20, https://doi.org/10.1109/ARES.2014.10.