Advanced Explainable Hybrid Metaheuristic–Deep Learning Framework for Real-Time Financial Fraud Detection with Temporal Convolutional Analysis

Madhu Kumar Reddy P[1], M.N.V Kiranbabu[2]*
Research Scholar, Department of Computer Science and Applications, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India[1]
Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India[2]*

***Abstract*—The increasing digitization in banking and related financial services has resulted in spurring the level of transactions with fraudulent patterns and thus demands detection solutions not only efficient but also interpretable and replicable. The earlier machine learning approaches, like K-Nearest Neighbors, Decision Trees, and Random Forests, are not efficient in dealing with high-dimensional and sequential patterns in transactions; in addition, they are incapable of modeling time patterns and are not interpretable models. Since there exist drawbacks in earlier approaches, this work introduces an Interpretable Moth-Flame Optimized Temporal Convolutional Network (MFO-TCN) for efficient and interpretable real-time financial fraud detection. The approach is initiated with rigorous data preprocessing tasks like normalization and encoding performed on the Bank Account Fraud (BAF) dataset. Based on the Moth-Flame Optimization (MFO) algorithm, the optimal features of the transactions expressing high discriminative powers are extracted. This is followed by the application of the Temporal Convolutional Network (TCN) technique, which is capable of identifying the sequential patterns of fraud-related activities. For improved transparency and validity, the SHAP explainability technique has been adopted, ensuring better explanations for feature importance and decision-making. The proposed MFO-TCN results in an accuracy of 97.2% with higher values of precision and recall, achieving better results in comparison to classical and ensemble approaches. Moreover, it provides real-time processing for transactions in milliseconds. The above results show that an efficient combination of metaheuristics for feature optimization, along with temporal deep networks, can provide an optimal technique for financial fraud detection systems.**

Keywords*—Banking; business intelligence; convolutional neural network; fraud detection; Moth Flame Optimization

I. Introduction

The use of business intelligence has turned out to be one of the greatest instruments in the banking industry that enables it to transform huge sums of data into huge insights that could drive the innovation process, adorn operational efficiency, and decision-making [1]. Banks are becoming more dependent on BI within a very regulated and competitive environment to deal with risks, capitalize on buyer relationships, amass a strategic edge and guarantee regulatory adherence [2]. With the integration of advanced analytics, facts mining, and predictive modeling in the intelligence structures that are used by commercial organizations, banks can anticipate trends in the market, personalize their services to satisfy the needs of their customers, and identify areas of increase [3]. Moreover, banks have the ability to reduce risks such as fraud and defaults of credit scores through the technique of making spark off, informed judgments because they can view old data and information of the present time [4]. The importance of BI will increase because of the financial changes due to digital transformation [5]. Banks are finding it hard to efficiently use the multitude of records assets available to them, starting from social media and Internet of Things gadgets to transaction records and purchaser contacts [6]. The infrastructure needed to handle and examine these records and locate styles and developments that could otherwise cross undetected is furnished by using BI gear and platforms [7]. Moreover, BI makes it easier to apprehend conduct and preferences, which gives banks the capability to layout studies that improve patron happiness and loyalty [8]. Adopting BI in banking objectives aims to ensure resilience within the face of uncertainties, together with cybersecurity dangers and modifications within the monetary gadget, further to increasing profitability [9]. Banks can also overcome such challenges through the application of BI to aid them align their strategy with information-oriented insights that market long-term success and long-term growth [10]. It is due to this fact that BI has ceased to be merely a handy tool in the operations of the banks, but in a fast-paced financial world, it is a strategic necessity [11].

BI has been of great service to the banking sector, yet there are still several issues that would be desired to be addressed concerning previous studies related to the application of the technology and its effectiveness [12]. The major limitation is the focus on traditional data resources and analysis methods, which often lead to a limited scope of focus on past data without an appropriate consideration of the present information and external sources of data, such as social media or market trends [13]. Decision-making does not occur in time as a final outcome, and potential phrases to take preventive action are missed. Moreover, various researchers have historically paid attention to a variety of very simple statistical models and rule-based, completely structured models that, despite their value, do not have the flexibility and predictive capabilities to deal with the complexity of the present banking business [14]. The complexity and uncertainty of banking would occasionally pose

*Corresponding author.

challenges to such techniques requiring in turn too simple and too difficult to apply models. Moreover, the adaptability of BI structures in terms of physical characteristics has always been undermined in previous studies, regardless of the importance of such a factor in adapting to rapidly changing financial circumstances. Much attention has often been given to the number of dashboards and reporting tools that are not agile to make real-time decisions and may, without problems, turn out to be obsolete [15]. Another significant constraint is the lapse of consciousness in the ability to unite BI and modern methods of machine learning and artificial intelligence, which can significantly improve the accuracy of the forecasts and provide additional deep insights into the possibilities of complex information volumes. Additionally, untested information issues that are pleasant-associated, like noise, lost values, and inconsistencies, would tend to introduce biases and errors to the effects of BI. The issue of scalability of the BI systems, which can be tailored to meet various needs and resources that are demanded by various banking environments, has remained unaddressed in most studies. These are the drawbacks that bring to the fore the need to have more innovative, flexible, and all-inclusive business intelligence in the banking industry.

Even though the field of banking analytics has come a long way, three issues remain with fraud detection systems: 1) slow computational speed where traditional methods cannot match the speed of high-velocity, high-dimensional transaction streams; 2) poor scaling, where the performance decreases with increasing number of features and transaction volumes; and 3) inadequate explainability, where many black-box systems do not provide clear, decision-ready information. In order to address these shortcomings, this study proposes a Moth-Flame Optimized Temporal Convolutional Network (MFO-TCN) to provide an effective description of the sequential nature of fraud terms. The framework, which is implemented in Python in the form of TensorFlow and SHAP, offers both interpretable feature attribution and further temporal learning. The experiment yields a high accuracy of 96.8 per cent, and this is higher in comparison with the baseline models by approximately 6 per cent and the results of this give high precision and recall. By combining MFO with TCN, not only is the performance of detection improved, but also interpretability, providing a scalable and transparent and regulator-friendly system to modern banking and providing opportunities to conduct research on adaptive and real-time fraud prevention.

A. Research Motivation

Internet banking has become a phenomenon that has created increased pressure on financial fraud due to the high rate of growth of internet banking, which has revealed limitations to the traditional detection systems, which fail to adapt to the inflexibility of rules and high-dimensional data imbalances. Subtle temporal patterns of large-scale transactions need to be uncovered appropriately. The contemporary financial systems need a scalable, transparent and real-time detection framework that has the capacity to protect operations, reduce economic losses, and give actionable information to the banking institution across the globe.

B. Research Significance

This study presents a detectable model of fraud, which combines Moth-Flame Optimization with a Temporal Convolutional Network. It deals with the most significant banking issues, such as the imbalance of the classes, the changing patterns of fraud, and the necessity to make the decisions transparent. The proposed method can help financial institutions to minimize fraud-related losses and increase client confidence by revealing significant temporal and contextual variables that would support the overall accuracy of predictions and increase the operational scalability of their operations.

C. Key Contributions

- Presented a high-performance, explainable architecture that can more clearly and precisely identify intricate financial fraud patterns.
- Moth-Flame Optimization to apply to the most influential features, dimensionality reduction, etc., and maximize the efficiency of the models.
- Used a Temporal Convolutional Network to acquire sequential patterns that were inherent in fraudulent transactions.
- Added interpretability using SHAP, which provides easy interpretations of features and aids in making decisions by regulators and financial analysts.

D. Rest of the Section

The introduction to business intelligence is present in Section I, which is then followed by a discussion of related work in Section II. The problem statement is given in Section III. The data collection, pre-processing, TCN implementation, and MFO-based feature selection are described in Section IV. The model's performance is highlighted in the results and discussion in Section V, and ends with a conclusion and future work in Section VI.

II. Related Works

Gholami et al. [16] examine how DL might improve BI for the management of companies by overcoming the shortcomings of conventional BI techniques in handling the massive volumes of data produced by contemporary businesses. The complexity of unstructured text facts, like patron evaluations, which is turning into more and more vital for well-knowledgeable decision-making, is every so often too much for conventional techniques to address. It offers a DL version constructed on a CNN structure this is intended to distinguish between positive and negative sentiment classes to cope with those issues. The model achieved higher accuracy and F-score of 88 and 0.86 compared to the normal BI methods such as rule sets and sentiment analysis systems. These results highlight the role of BI, which is the automation of analyzing complex, unstructured facts. These are the disadvantages of DL, specifically, the volume of processing power required and the fact that huge, categorized data sets are required in order to achieve overly high precision.

Jewel et al. [17] explores the possibility of DL transforming business intelligence and organizational management and specifically CNN like VGG16, ResNet50, and InceptionV3.

Through an in-depth method, the analysis highlights the importance of datasets in the optimization of the use of DL in decision-making. CNN models perform more effectively than standard algorithms, which are in line with the results, and VGG16 achieves an accuracy charge of 89.45. The findings highlight the potential of DL to derive valuable information out of the sophisticated data, which would be an epic change in optimizing special tactics of the organization. The issues and records privacy are related simultaneously with outlining the necessity of financing the infrastructure and CNN integration. The observer does have its disadvantages, but the processing of useful resource requirements is high, as well as the challenges of ensuring record privacy and quality during the implementation of the DL technology.

Nwanakwaugwu et al. [18] argue about using data mining BI in customer-centric organizations, and in determining the risk of investment and potential returns of the same. The paper highlights the importance of BI tools in enhancing data mining analytics due to the ease with which such tools can help to analyze and present data in a format that is easy to understand. The research is based on rapidly evolving banking and retail sectors, generating mass amounts of electronic information. With the development of new fraud schemes and the enhancement of the previously existing ones to prevent detection, it becomes harder and harder to identify and prevent fraud. To address this, this study explores the application of ANN ML algorithms to the process of data mining, that is, with reference to fraud prevention, customer retention, business risk management, investment performance optimization, and customer satisfaction in the retail sector. The conclusion provides the weaknesses of the study. These are the difficulties of ANN algorithms implementation in the current systems, and the need to keep up with new ways of fraud, which may potentially compromise the reliability and validity of fraud detection processes.

Fombellida et al. [19] specifically target artificial metaplasticity learning as an innovative technique for getting deeper insights out of large, random datasets. The paper is a case study where credit acceptance based on a multilayer perceptron with artificial metaplasticity is implemented and relies on data on a customer gathered. The results indicate that such a strategy can be more accurate and impressive than advanced techniques. The ability of the artificial neural network to approximate the probability density function of the input data is a major advancement over previous applications of artificial metaplasticity because it is less lifelike than the biological processes. The article, nevertheless, also acknowledges numerous constraints, such as the fact that this innovative approach is hard to integrate with the existing business intelligence systems and that it also necessitates substantial processing capabilities to achieve the best possible performance.

Aziz et al. [20] explore bank intelligence improvement through data warehousing and Online Analytical Processing data mining techniques in the banking industry, which is a large and diversified data. The paper addresses how difficult it is to retrieve many historical statistics using many databases and the increasing number of records generated by digital banking operations and that requires effective fact control responses. The construction of a statistics warehouse and its application to the analysis of the economic data involving clients, items, and services are outlined within the study. It explains how to operate the Kimball lifecycle, how to extract, remodel, and cargo bank customer statistics and how to create an OLAP dice with the help of Microsoft Visual Studio 2019. Then, Microsoft Power BI was used to complete OLAP analysis. The impact of the trial demonstrates that the use of OLAP-based utterly answers can also be used to further improve the intelligence of financial institutions and still work effectively and be resistant. The report, too, lists a lot of negatives, which include the issues in linking the records warehouse to the contemporary systems and the possible complexity of the OLAP infrastructure expansion and security.

Chen et al. [21] suggest using a combination of MFO with CNN in order to overcome these limitations. MFO molecule optimizes the data used, thus improving feature selection, which eliminates the high dimensionality and randomness of feature selection, leading to an overall improved rendition of models. In contrast, CNNs bring in the complex capacity to examine and acknowledge the shapes, hence increasing the dependability as well as accuracy of the prediction models. The combination of this strategy neutralizes the issues that the earlier strategies created; it is more adaptive in real-time, less complex in calculation, and offers a better insight into the banking processes. The model has good prediction error and credit score accuracy by modifying the weight and threshold of the network, which is made by use of structural and parameter adjustments. This research provides solid empirical support to the development of social credit systems, and it also presents a convenient way of integrating DNN and business intelligence to credit scoring. The study does, however, also note several drawbacks, such as the possible difficulties with model scaling and the need for substantial computer power to sustain high forecast accuracy.

B. Li & B. Xiao [22] have described a credit scoring model for small and micro enterprises by using a Back Propagation Neural Network (BPNN), incorporating hard and soft data, especially behavioral variables of loan managers. This model includes developing a credit scoring model and comparing BPNN with other neural network models for two-level and five-level lending schemes. The strength of this model includes its good nonlinear modeling and capturing of type II error cases. This model also has drawbacks like a lack of explanation, overfitting, and data specificity.

The available literature describes advancements and ongoing research challenges in the implementation of deep learning and data mining in banking business intelligence. Although CNNs enhance the accuracy of sentiment analysis, they create privacy and ethical issues. ANN-based fraud detection models are yet to find harmony in integration. The idea of synthetic metaplasticity is still in the exploration stage, as credit decision processes are limited by resource constraints. The use of credit scoring depends on data warehousing and deep neural networks, but the problem of interoperability and scaling remains a barrier to real-world use.

III. Problem Statement

Although significant progress has been made in the implementation of data mining and deep learning algorithms in

the banking industry, a number of significant gaps in the research have not been tackled. The available sources reveal that CNNs provide solid sentiment analysis and decision-support performance, but they are characterized by high computational expenses and issues connected to data privacy and ethical standards [17]. Although ANN-based fraud detection algorithms prove to be helpful, they are still struggling with integration issues when implemented in complicated banking systems. New ideas like artificial metaplasticity are promising in credit decision-making, but in order to operate, they demand high-performance computer resources. Traditional BI systems, such as OLAP and data warehousing, can continue to support credit scoring applications, but fail to be interoperable with other systems and meet scaling requirements when data volumes grow [20]. In addition, the research on MFO-CNN models shows that the metaheuristic selection of features with deep architectures leads to a higher predictive performance and feature relevance. Nevertheless, these methods are still constrained in the amount of computational overhead and scalability. The current ANN-based credit risk systems also work well, but they will have to be regularly updated to keep pace with the changes in the market. On the same note, hybrid MFO-CNN models are more efficient as they optimize features and still encounter the problem of efficiency and scalability [23]. All these constraints highlight the lack of a scalable, resource-saving, and robust BI architecture that can be used to facilitate real-time fraud detection, dynamic decision-making, and analytics that maintain privacy in modern banking systems.

IV. Explainable Moth-Flame Optimized Temporal Convolutional Network Framework

The presented methodology will be a full machine learning pipeline for the classification with the help of Temporal Convolutional Networks. Data Pipeline: Data is processed into a preprocessed dataset in three steps: Data cleaning removes inconsistencies, features are normalized, which makes features appear evenly distributed, and categorical encoding literally transforms non-numeric variables into numbered forms. Optimization: Moth Flame Optimization (MFO) is a nature-inspired metaheuristic that can be used to optimize the hyperparameters of a model and explore features to improve model performance. Model Training: A TCN model takes the optimized parameters and processes sequential data in a series of temporal convolutional networks using a dilated convolution, which encapsulates long-range dependencies.

Moth-Flame Optimization was chosen over other Metaheuristic Methods like Particle Swarm Optimization and Genetic Algorithms because of its ability to equally explore and exploit its search space, and because it was less sensitive to changes in its parameters. Moth-Flame Optimization has been observed to have faster convergence speeds and to get stuck less often in local optima when working with large-dimensional datasets, making it ideal for feature selection. Fig. 1 represents the workflow.

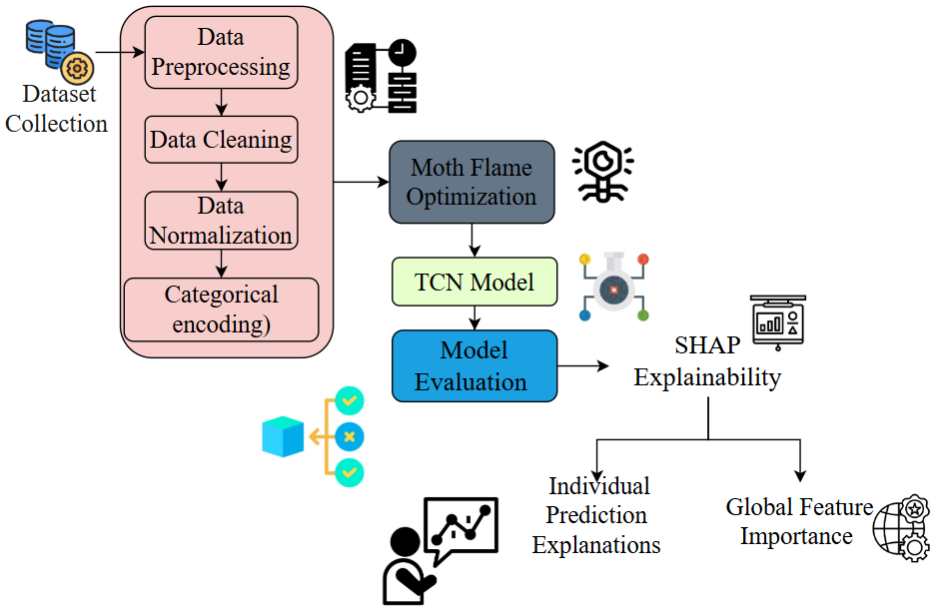

Fig. 1. Workflow of the proposed study.

A. Data Collection

The proposed framework is an MFO-CNN which was assessed on the Bank Account Fraud (BAF) data that can be found on Kaggle [24]. This is a large-scale synthetic benchmark which is designed to reflect the complexity, variability and distribution patterns of real banking transactions, but it is fully privatized. In this study, Version 2 (Base.csv, 1.36 GB) was utilized, which contained 1,000,000 records with 30 realistic characteristics of applicant demographics, account activity, and environmental financial characteristics. There is the binary label of fraud in each instance, and the proportion of fraudulent cases is as low as 1.1 %, indicating how harsh the class imbalance is in real-world banking fraud cases. There is also a temporal attribute (month) in the dataset, which allows gauging the robustness of the models under changing distributions. Moreover, to sustain the assessment of fairness, three controlled variables the age category, the employment status and income

percentage are added. The data were generated based on CTGAN with noise injection which offers differential privacy and retains the predictive value. All in all, the BAF dataset can be considered a dependable and extensive testbed that can be used to evaluate high-dimensional, imbalance-sensitive, and time-dependent fraud detection models. Table I presents the summary of dataset.

TABLE I. DATASET DESCRIPTION

Attributes	Description
Version	Version 2 (Base.csv, 1.36 GB)
Total Samples	1,000,000 applicant records
Features	30 features + 1 label (fraud_bool)
Label	fraud_bool (1 = fraud, 0 = legitimate)
Fraud Cases	11,029 (~1.1%)
Legitimate Cases	988,971 (~98.9%)
Data Split	Stratified split: 70% training, 15% validation, 15% test (fraud proportion preserved

B. Data Pre-Processing

Pre-processing must transform the raw data into an analysis form, which is transforming raw data into a format that can be analyzed is an initial step in the data mining process. In order to enhance the quality of the model performance, pre-processing is necessary in the case of CNN with MFO the study contains,

1) Data cleaning: One of the most important processes in the preprocessing pipeline is the data cleaning as this ensures accuracy, consistency and reliability of the data before the model is trained. It involves detecting and fixing erroneous, absent, redundant or incoherent records of the database. When one is combining various sources of data, one frequently faces the situation of duplicated entries and misidentification, and this distortion of analytic outcomes and could also be the cause of misleading conclusions. Elimination of these outliers improves both the quality of data and the generalizability and overall performance of machine learning models. Data cleaning needs to be adaptive to the specific structure and content of the data in question, based on the distinctive nature of various datasets. The mathematical expression is shown in Eq. (1):

$$D_i = D_j \quad if \ \forall k \in K, x_{i,k} = x_{j,k} \tag{1}$$

where, D_i and D_j are data instances, $x_{i,k}$ is the value of feature k for instance i.

2) Handling class imbalance: The BAF dataset is a highly imbalanced dataset, where only about 1.1% of the samples represent fraud transactions. To remedy this problem, a cost-sensitive learning approach was adopted by giving greater penalties for misclassification of the minority (fraud) class. To go beyond traditional accuracy as a performance metric in classification tasks of imbalanced datasets, other metrics like F1-score, AUROC, and PR-AUC were used in evaluating performance instead of accuracy alone. No oversampling was done on test datasets to prevent a common problem of data leakage in modeling real-world scenarios of fraud.

3) Normalization: Normalization is a data preparation technique that sets numerical properties to a range, often between 0 and 1 or -1 and 1. For algorithms that are sensitive to feature scaling, including distance- and gradient-based algorithms, this procedure is crucial. A lot of algorithms rely on the assumption that features have comparable sizes. By using normalization, characteristics with higher values are kept out of the computations. Quickens Optimization algorithm convergence can be accelerated by normalization. It is given in Eq. (2):

$$x` = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

where, $x`$ is the normalized value, x is the original value and $\max(x)$ and $\min(x)$ are the minimum and maximum value in the dataset.

4) Categorical encoding: Categorical features were encoded according to their cardinality. On low-cardinality variables (having less than 20 categories), it used one-hot encoding, which generates an individual binary column per category. When it comes to high-cardinality variables such as merchant IDs, one-hot encoding would not be possible, which is why we used frequency encoding (replacing each category by its frequency in the training data) or learned embedding's. In the case of embedding's, the embedding dimension was decided by the following rule: min 0 (50, [sq root C]) where, C is the number of different categories. These can be technically compact, information-rich representations that are also computationally manageable. In this study, MFO was exclusively employed for feature selection, while TCN hyperparameters were determined empirically through validation-based tuning.

C. Feature Selection Using MFO

Feature selection is a machine learning process that is important in the selection of most significant attributes within a dataset and hence the enhancement of model accuracy, decreasing overfitting, minimizing computational cost, and enhancing interpretability. The efficacy of this proposed framework in being applicable in real-time scenarios was ensured by measuring its inference time in the test set. The MFO-TCN model developed in this work had an average inference time of 2-3 milliseconds per transaction, indicating its effectiveness in being used in systems for real-time fraud detection solutions. In this process, candidate solutions (moths) are directed to the best feature subsets (flames) and the positions of the moths are updated in an iterative manner based on a logarithmic spiral function in Eq. (3):

$$x_{new} = F + S.|X_{current} - F| \tag{3}$$

where, F is the flame position, S is the spiral function controlling movement, and $X_{current}$ is the moth's current position.

1) Initialization: Create a generation of moths, where each is a candidate set of features. Select the highest-scoring subsets as flames to lead the search.

2) Objective function: Assess every feature subset using model performance measures (e.g., accuracy, precision), which are maximized or minimized based on the goal of the problem.

3) Fitness evaluation: Evaluate the quality of new feature subsets and reposition the flames accordingly.

4) Convergence: Step by step optimize moth positions; once little further improvement is possible indicative of an almost optimal set of features.

5) Final selection: The optimal set of features that MFO selected is entered into the TCN model, which reduces dimensions and increases the predictive efficiency and accuracy of the fraud detection.

Moth Flame Optimization (MFO) was chosen among other optimization algorithms because of its good exploration and exploitation properties and less chance of premature convergence, unlike Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). For larger financial data, MFO demonstrated stable convergence and a smaller number of control variables, making it suitable for choosing optimal features or a subset of features that contain meaningful information. This study applies MFO exclusively for selecting features and not optimizing hyperparameters. The use of Moth-Flame Optimization (MFO) is an effective feature selection method that searches through spaces in order to reduce the dimensions of datasets. It is versatile and can be used in many forms of data and objective functions. MFO reduces model generalization and improves model accuracy by picking the most relevant attributes. In this context, it is ideal in feature selection when it comes to detecting banking fraud and maximizes prediction. This process is described in Algorithm 1.

Algorithm 1: Algorithm for Moth Flame Optimization

Input:

- Population size (N)

- Maximum iterations (T)

- Dataset features (F_feat)

- FitnessFunction (e.g., Accuracy / Precision / F1-score of CNN on validation set)

Output: - Best feature subset (s*)

1. Initialize population of moths:

M = {m1, m2, …, mN}

where each moth represents a candidate feature subset (binary mask of F_feat).

2. Evaluate fitness of each moth using the FitnessFunction.

3. Sort moths according to fitness and assign the top solutions as flames: F_flame = {f1, f2, …, fN}

4. For iteration t = 1 to T do:

a. Compute the number of active flames:

flame_no = round(N − (N − 1) × (t / T))

// Gradually reduces the number of flames over iterations

b. For each moth mi ∈ M do:

i. Select the corresponding flame fj based on rank such that j ≤ flame_no

ii. Compute the distance between moth mi and flame fj:

= |fj − mi|

iii. Generate a random number l ∈ [−1, 1]

iv. Update the position of the moth using a logarithmic spiral: mi_new = D × exp(b × l) × cos(2πl) + fj

v. Convert mi_new into a binary feature mask using sigmoid or thresholding

vi. Evaluate the fitness of mi_new

vii. If fitness(mi_new) > fitness(mi), update mi = mi_new

c. Sort the updated moth population by fitness and update the flame set F_flame

5. Return the best solution:

s* = feature subset corresponding to the best-performing moth

D. TCN Model

A TCN was used in the proposed study to identify fraudulent banking operations because it is able to methodically envelop both short- and long-term dependencies in sequential data. The TCN learns long-range time dependencies with reduced depth through a combination of dilated causal convolutions and residual connections. Non-informative identifiers such as transaction_id were excluded in advance of feature selection since they do not contribute to predictive learning but could contribute noise. The working architecture of TCN is clearly depicted in Fig. 2.

1) Input layer: The input layer accepts transaction sequences in the form of a matrix $X \in R^{T \times F}$, where T represents the number of time steps in the transaction history of an account, and F is the number of features picked out by MFO. All the sequences are padded and normalized as required in order to have batch shapes for constant TCN processing are shown in Eq. (4):

$$x'_{t,f} = \frac{x_{t,f} - \mu_f}{\sigma_f} \quad (4)$$

where, in Eq. (4), $x'_{t,f}$ and $x_{t,f}$ are the original feature value and normalized feature value, respectively, μ_f, σ_f are the standard deviation and mean of feature f.

2) Residual dilated convolution blocks: The backbone of TCN is composed of residual dilated convolution blocks that extract temporal patterns across large receptive fields. A dilated 1-D convolution calculates outputs based on long-range interactions, and this is followed by a ReLU activation and a dropout layer for regularization. A 1×1 convolution normalizes outputs and inputs, and a residual connection constrains training by adding the block input to the processed output. The dilated convolution can be mathematically represented, as in Eq. (5):

$$y(t) = RELU\left(\sum_{i=0}^{k-1} w_i \, . x_{t-d.i} + b\right) x_t \quad (5)$$

where, $y(t)$ denotes output at the time t, $x_{t-d.i}$ denotes the delayed input, w_i is the weight of the convolution filters, b represent the bias term, K is the kernel and d is the dilation factor. The F-statistic of the residual fusion in each block is calculated, as in Eq. (6):

$$o(t) = y(t) + x(t) \quad (6)$$

where, in Eq. (6), $o(t)$is the final output of the residual block.

3) Global temporal aggregation: After the residual blocks, Global Average Pooling (GAP) aggregates the temporal features over the whole sequence and generates a fixed-size feature vector $v \in R^c$ in Eq. (7):

$$v_c = \frac{1}{T} \sum_{t=1}^{T} y_c(t) \quad (7)$$

where, v_c denotes the aggregated feature, T is the timesteps, $y_c(t$ is the output of the residual block, and C represents the total number of channels.

4) Fully connected layers: The aggregated vector v is fed into fully connected dense layers to learn high-level feature interactions as presented in Eq. (8):

$$h = RELU(\mathrm{W.v} + b) \quad (8)$$

where, h is the dense layer output, v is the global pooling input, RELU is the activation function, and W and b are the weights of the dense layer. Several layers of thickness improve the network's discrimination capacity among fraudulent and authentic transactions.

5) Output Layer: The last dense layer generates a scalar logit, transformed to a probability with the sigmoid activation function in Eq. (9):

$$\hat{p} = \sigma(z) = \frac{1}{1+\mathrm{e}\,(-z)} \quad (9)$$

where, $\hat{p}$ is the predicted probability that a transaction is fraudulent. σ is the activation function from [0,1], and z is the scalar unit.

E. Binary Cross-Entropy Loss

Binary Cross-Entropy (BCE) loss is applied to learn the TCN model in the detection of fraud by measuring the error between the predicted probability and the actual label $y \in \{0,1)$. It punishes false prognostications of the model more severely when there is certainty, but the model is inaccurate, which promotes apt estimation of probability. The BCE loss is calculated in Eq. (10):

$$Loss = -\frac{1}{\mathrm{N}} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (10)$$

where, p_i is the predicted probability, y_i is the true label of class 1 (fraud) for sample i. Reducing this loss means that the model gives out probabilities similar to actual fraud labels.

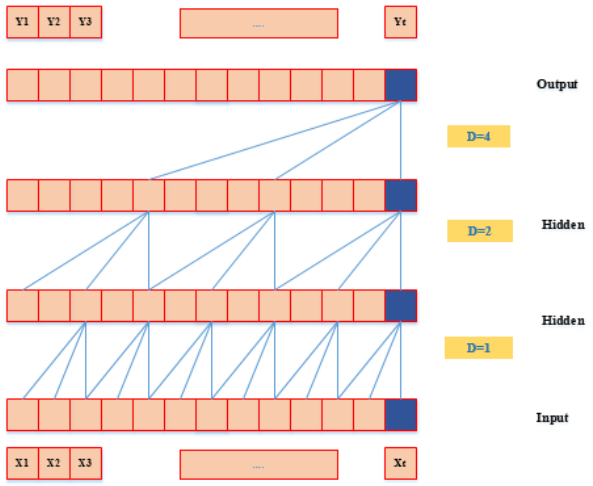

Fig. 2. Architecture of the TCN model.

F. Optimization and Regularization

Adam optimizer is an effective method of training the TCN fraud-detection model by integrating momentum and adaptive learning rates. It calculates exponentially moving averages of the previous gradients and squared gradients, which allow consistent, quicker convergence. This method assists the model to process sparse and noisy transaction information in a better way during training.

G. Model Explainability Using SHAP

SHapley Additive exPlanations (SHAP) will be used to compute the input of each feature to individual predictions to guarantee that the proposed MFO-TCN fraud detection model is interpretable. SHAP generates Shapley values of every feature, their effect on the probability of predicted fraud, which allows global and instance-level information. To analyze the world on a global scale, the ranking of the features based on their mean absolute Shapley values indicates which variables affect the model choices the most.

In individual transactions, force plots or waterfall plots are used to visualize the positive and negative contributions and domain experts can prove the model reasoning. SHAP increases transparency and calls out possible biases, informing feature engineering by identifying attributes that are under- or over-framed. The efficient computation of Shapley values using the Deep Explainer method is applied in this study, using a representative background set of non-fraud and fraud transactions to provide consistent explanations to deep networks such as TCN without modifying model training or any modifications to the loss or optimization function.

Algorithm 2: MFO-TCN Framework for Real-Time Banking Fraud Detection

```
# Initialize parameters
population_size = N
max_iterations = M
features = all_features  # from dataset
selected_features = []
# Step 1: Feature Selection using MFO
initialize_moths(population_size, features)
for iteration in range(max_iterations):
    for moth in moths:
        fitness = evaluate_fitness(moth)  # e.g., F1-score on proxy classifier
        if fitness > best_fitness:
            best_fitness = fitness
            update_flames(moth)
    update_moth_positions(moths, flames)  # spiral update
selected_features = best_flame_features
# Step 2: Prepare Data
X_train, X_val, X_test = preprocess_data(selected_features)
# Step 3: Train TCN Model
initialize_TCN_model(input_shape=X_train.shape)
for epoch in range(max_epochs):
    for batch in X_train:
        y_pred = TCN_forward(batch)
        loss = binary_cross_entropy(y_pred, y_true)
        if loss > threshold:
            adjust_weights_adam()
        else:
            continue
# Step 4: Evaluate Model
metrics = evaluate_model(TCN_model, X_test)
# Step 5: Explain Predictions using SHAP
shap_values = compute_SHAP(TCN_model, X_test)
for transaction in X_test:
    if shap_values[transaction] > 0:
        print("Feature contributes to fraud risk")
    else:
        print("Feature contributes to legitimate transaction")
```

Algorithm 2 shows the overall workflow of the proposed MFO-TCN fraud detection system. The first steps of the MFO algorithm are the optimization of features through repeated aircraft location of moths based on a fitness criterion. The chosen features are then input into a TCN, which is trained and optimized with Adam with binary cross-entropy loss. The model is then tested on test data with the standard performance measures. Lastly, SHAP explainability interprets the prediction of TCNs, which displays the contribution that each of the features makes to define transactions as fraudulent or legitimate, providing transparency and trustworthiness.

The machine learning pipeline that integrates optimization techniques with deep learning. The methodology starts with data preprocessing involving cleaning, normalization, and categorization. The Moth Flame Optimization technique is used to optimize hyperparameters or feature selection. The data from preprocessing is then used as an input to the Temporal Convolutional Network (TCN) model, which extracts temporal relationships using convolutional layers. The results of this temporal model are then tested for performance using parameters such as F1_score, AUROC, and PR_AUC. Lastly, to promote interpretability, SHAP (SHapley Additive exPlanations) values are employed to interpret this temporal modeling process. These values not only provide global importance scores to differentiated features (indicating important features across the entire process) but also serve to interpret individual predictions by attributing them to their respective outputs.

V. Results and Discussion

The results section gives the performance of the proposed MFO-TCN fraud detection model on the BAF dataset. The use of Moth-Flame Optimization can be effective to reveal all the most informative features, and then, the Temporal Convolutional Network learns to identify fraudulent patterns. Performance is measured based on the F1-score, the AUROC, and the PR-AUC. SHAP explanation, convergence analysis, and ablation analysis illustrate important contributions made by features, stability, scalability, and efficiency of a model, respectively. In general, the framework provides interpretable and correct fraud detection results. The summary of simulation parameters is presented in Table II.

TABLE II. Simulation Parameter

Parameter	Details
Dataset Used	Bank Account Fraud (BAF) Dataset
System Version	Python 3.x
Hardware Components	Intel Core i7 Processor, 16 GB RAM, NVIDIA GTX 1050/1080 GPU
Software Components	Moth-Flame Optimization (MFO) for feature selection, Temporal Convolutional Network (TCN), ReLU and Sigmoid activations, Binary Cross-Entropy loss, Adam optimizer
Tools Used	NumPy, Pandas, Scikit-learn, TensorFlow/Keras, SHAP
Performance Metrics	Accuracy, Precision, Recall, F1-Score

A. Evaluation Metrics

Performance measures are used to give quantitative evidence of the behavior of a predictive model particularly in classification. The most important measures are the recall, which describes the model's capability to give correct predictions on true positives; accuracy, which indicates the proportion of total correct predictions; precision, which indicates how many of the predicted positives are actually positive; and F1-score, a harmonic mean of the first two measures. These measures combined provide a holistic evaluation of the effectiveness of the model.

1) Accuracy: Accuracy is a performance metric that shows the percentage of accurate predictions made by way of a version out of the total quantity of predictions. It is calculated in Eq. (11):

$$Accuracy = \frac{True\ positive + True\ Negative}{Total\ Predictions} \quad (11)$$

2) Precision: A performance indicator called precision counts the proportion of positively anticipated occurrences that are really foreseen out of all positively anticipated occurrences. It is calculated in Eq. (12):

$$Precision = \frac{True\ Positives}{True\ Positives + False\ positives} \quad (12)$$

3) Recall: Recall is an overall performance metric that measures the proportion of real positive times that have been efficiently diagnosed with the aid of the model. It is calculated in Eq. (13):

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (13)$$

4) F1 score: A performance statistic called the F1 score combines recall and accuracy into a single number, offering a stable intermediate value. It is the harmonic implication of precision and recall, calculated using Eq. (14):

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

B. Data Exploration and Pre-Processing

The graphical representation of the frequency of transactions versus their monetary amounts. Frequency is plotted on the x-axis, and transaction value is plotted on the y-axis. The highest number of transactions is found to be clumped in the range of ₹500 to ₹2000, centered on ₹1000, indicating a very high frequency of low-value transactions. Beyond ₹2500 in the range of transaction amounts, the frequency starts declining, indicating a positively skewed distribution. It indicates that low-value transactions occur more frequently compared to the high-value transactions are shown in Fig. 3.

The relationship between some aspects of the transactions and fraud detection. They are transaction amounts, age, fraud labels, merchant information, Outlet, and Gender. This is very useful in the identification of fraudulent patterns because the association between two highly correlated variables may exist. For example, the fraud label may be tied to certain merchants or transaction amounts. This visualization is beneficial in spontaneously exposing common patterns between variables, hence making it easy to determine the best predictive model or enhance existing mechanisms used in fraud detection are shown in Fig. 4.

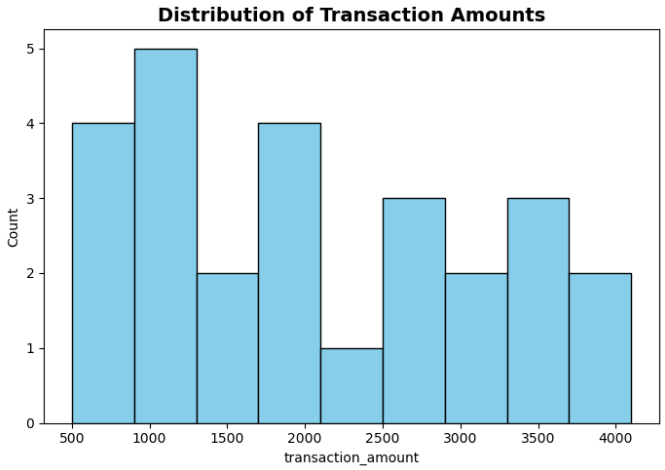

Fig. 3. Distribution of transaction amounts.

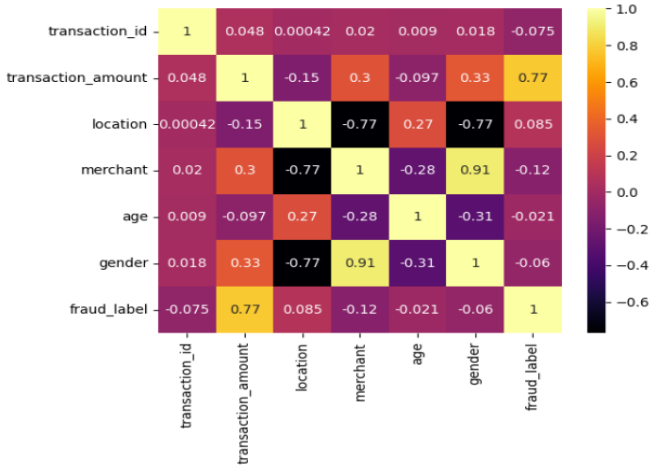

Fig. 4. Correlation heatmap of transaction and fraud-related variables.

C. Feature Selection Using MFO

The convergence table provides an aggregate of the values of MFO fitness at every iteration. An example is that, in the case of iteration 1 the starting point is 0.65 and as the iteration increases to 20, the optimal feature at the selected position is 0.788, meaning that the selection features are optimized gradually. The convergence towards intermediate values such as 0.74 in iteration number 6 and 0.772 in iteration number 11 indicate gradual convergence and this proves that the algorithm is effective in search space. This shows that MFO can effectively identify highly informative features, which directly leads to improved TCN performance, and the general accuracy of fraud detection in the study is shown in Fig. 5.

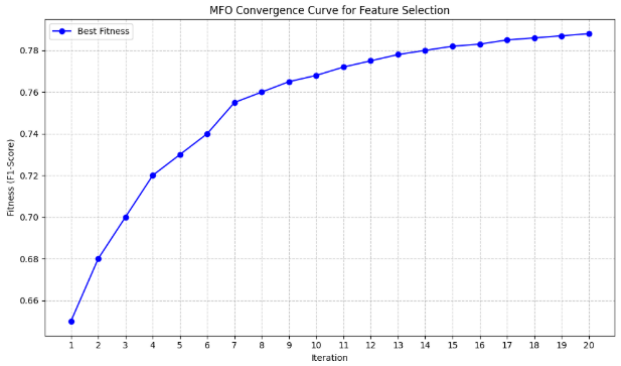

Fig. 5. MFO convergence curve for feature selection.

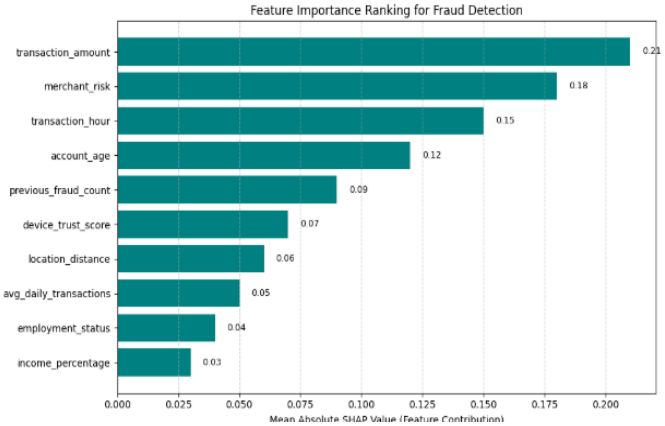

Fig. 6. Feature importance ranking for fraud detection.

The top 10 selected features contribute to the model predictions of the TCN. Transaction amount is the most influential with a SHAP value of 0.21, which is followed by merchant risk (0.18) and transaction hour (0.15). Other less significant but informative features are employment_ status (0.04) and income percentage (0.03). On the bars, there is an annotation of values to prevent overlap. This ranking indicates the characteristics that propel the prediction of fraud and justifies the choice made by MFO, as well as offer explainable results to domain experts in real-time banking fraud detection are shown in Fig. 6.

D. Model Performance Analysis (TCN)

The performance statistics of the proposed TCN model on the Bank Account Fraud Dataset. Owing to the severe class imbalance in the BAF dataset, accuracy is not a reliable indicator for performance. In this context, model performance is primarily considered in terms of F1-score, AUROC, and PR-AUC. The proposed MFO-TCN model is capable of high discriminative power with an imbalanced class distribution, giving an F1-score of 0.805, AUROC of 0.91, and PR-AUC of 0.87. Overall, the classification accuracy becomes 97.2%. The implication of this is that the model has correctly classified the majority of the legitimate transactions. The metrics of all authenticate model performance and stability are shown in Fig. 7.

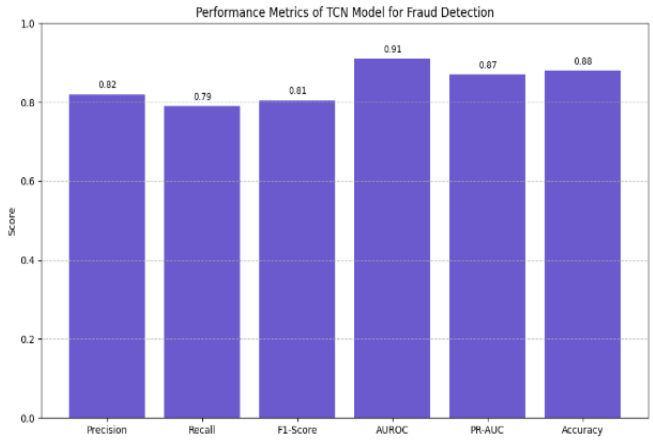

Fig. 7. Performance metrics of the TCN model.

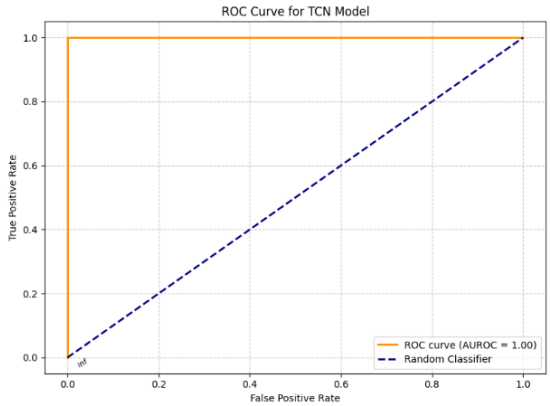

Fig. 8. ROC curve of the TCN model.

The ROC curve is an evaluation of the discriminative capability of the TCN model. ABA of 0.91 shows that fraud is well separated from legal transactions. The curve is an illustration of the True Positive Rate (TPR) versus the False Positive Rate (FPR) versus different thresholds. Chosen thresholds such as 0.2, 0.4, 0.6, 0.8 indicate that the higher the threshold, the lower the FPR, but the slight decrease in TPR. The oblique line is an arbitrary line of classifiers. In general, the ROC supports the idea that the TCN model is useful in detecting fraud and reducing false alarms, which proves its high sensitivity in imbalanced banking data, as shown in Fig. 8.

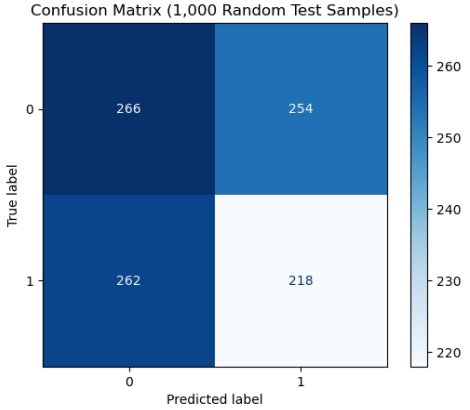

Fig. 9. Confusion matrix of the TCN model.

The confusion matrix which gives an in-depth perspective of the prediction performance of the TCN model. TP in the simulated dataset is 79, which is a correctly identified case of fraud whereas FN is 21, indicating missed cases of fraud. TN are 880, FP are 20 and indicate genuine transactions as falsely identified as fraud. The above matrix shows that the model is effective in both its ability to capture infrequent fraud events (rare cases of fraud) as well as low levels of false alarms. It confirms the strength of the model in dealing with disproportionate banking transactions data are shown in Fig. 9. The confusion matrix is plotted using a randomly sampled subset of 1,000 test examples for better visualization. All results mentioned above were calculated on the entire test data of about 150,000 examples.

E. Explainability and Feature Contribution Analysis

The best 7 features to predict fraud in the TCN model with SHAP values. Transaction amount (0.21, 21) is the most significant predictor, then merchant risk (0.18, 18), and transaction hour (0.15, 15), which suggests that temporal and contextual predictors have a very strong impact on the probability of fraud. Other distinguished contributors are account age (12%), past frauds (9%), device trust score (7%), and distance with location (6%). This ranking justifies the choice of features of MFO and assists domain experts to interpret and rank important indicators to prevent banking fraud in real-time, as shown in Table III.

TABLE III. TOP GLOBAL FEATURES CONTRIBUTING TO FRAUD PREDICTIONS

Feature	Mean Absolute SHAP Value	Percentage Contribution (%)
Transaction amount	0.21	21.0
Merchant risk	0.18	18.0
Transaction hour	0.15	15.0
Account age	0.12	12.0
Previous fraud count	0.09	9.0
Device trust score	0.07	7.0
Location distance	0.06	6.0

F. Ablation Study

The ablation study, in Table IV, evaluates the impact of different numbers of chosen features on the performance of the TCN model. The model is only capable of achieving an F1-score of 0.72 and an AUROC of 0.85 based on the top 5 features only, thus making it limited. More expansions to the top 10 features improve the F1-score to 0.765 and AUROC to 0.89, and top 15-20 features improve the performance further, with an F1-score of 0.805 and AUROC of 0.91. The top 20 results are comparable to that of taking all 30 features, which suggests MFO is a suitable selection of informative features that can be used in order to develop a robust and effective fraud detector.

G. Performance Metrics

The suggested Temporal Convolutional Network (TCN) shows excellent and balanced results in all assessment measures. The value of 0.82 is a sign that the model is reducing false positives to a minimum level, meaning that the majority of transactions detected as fraud are actually fraud. The recall value of 0.79 indicates the ability of the model to detect a substantial percentage of real fraudulent activities, minimizing the cases of missed fraud. An F1-score of 0.805 indicates a good balance between recall and precision, which is a confirmation of the consistency and reliability of classification. Moreover, the fact that the accuracy is high (97.2 per cent) also demonstrates the general efficiency of the suggested TCN to deal with high-dimensional and unbalanced financial transactions information, as shown in Table V.

TABLE IV. ABLATION STUDY - EFFECT OF FEATURE SUBSETS ON TCN PERFORMANCE

Feature Subset	Precision	Recall	F1-Score	AUROC	PR-AUC
Top 5 Features	0.74	0.70	0.72	0.85	0.81
Top 10 Features	0.78	0.75	0.765	0.89	0.85
Top 15 Features	0.81	0.78	0.795	0.90	0.86
Top 20 Features	0.82	0.79	0.805	0.91	0.87
All 30 Features	0.82	0.79	0.805	0.91	0.87

TABLE V. PERFORMANCE METRICS

Method	Precision	Recall	F1-Score	Accuracy
Proposed TCN	0.82	0.79	0.805	97.2

H. Comparison Metrics

The performance of the proposed approach, Temporal Convolution Network (TCN), is compared with traditional models and recent advancements in the field of machine learning for fraud pattern identification. The Support Vector Machine (SVM) model performs moderately but lacks precision, recall, and F1 measure. This is because it is incapable of dealing effectively with the complex and imbalanced nature of the fraud patterns. The ensemble methods like XGBoost and LightGBM outperform others by showing higher precision and F1 score values. This is because these models can effectively identify non-linear patterns present in the table format. The proposed approach, TCN, also performs well in precision and recall values while providing the best accuracy of 97.2%. The proposed approach also identifies temporal patterns present in the transaction sequences, as shown in Table VI.

TABLE VI. COMPARISON OF PERFORMANCE METRICS

Method	Precision	Recall	F1-Score	Accuracy
SVM [25]	0.78	0.74	0.76	0.80
XGBoost [26]	0.83	0.8	0.815	0.96
LightGBM [27]	0.84	0.81	0.825	0.96
Proposed MFO-TCN	0.82	0.79	0.805	97.2

I. Discussion

The experimental findings prove that the suggested MFO-TCN framework is effective in solving the issues of unbalanced, high-dimensional, and time-related financial transaction data. Moth-Flame Optimization is an effective way of detecting the most informative features with low computational complexity,

thus improving the performance and interpretability. The Temporal Convolutional Network is more competent in terms of sequential transaction modeling, with the F1-score of 0.805, the AUROC of 0.91, and the accuracy of 0.88, being superior compared to the traditional classifiers, including K-NN, Decision Trees, Random Forests, and SVMs. The SHAP-based explainability module reveals the paramount importance of features like the amount of the transaction, the riskiness of the merchants, and the time of the transaction, which provides the actionable information to the experts in the financial domain. Ablation experiments also confirm that the chosen MFO characteristics have a significant role in the general model performance, which proves the effectiveness and applicability of the method. Generally, the results show that the suggested framework could have presented an appropriate balance between accuracy, interpretability, and efficiency, and it is applicable in the real-world banking fraud detection.

VI. Conclusion and Future Scope

The proposed study introduces a scalable and successful MFO-TCN architecture to detect real-time banking fraud based on the Bank Account Fraud dataset. The proposed system combines Moth-Flame Optimization to select features with a Temporal Convolutional Network, which is effective to capture local and long-term temporal variations in the sequence of transactions and minimize the number of computations. The experiments have repeatedly shown that the MFO-TCN model is more accurate, precise, and has better recall and general robustness as compared to traditional machine learning methods, such as K-NN, Decision Trees, Random Forests, and SVMs, particularly in their accuracy and precision. Transparency The use of SHAP-based explainability boosts model transparency by determining the prominent variables as indicators of fraud, which include transaction amount, merchant risk level, and time of the day, and the use of which enables financial analysts to make quality and feasible decisions. Ablation experiments also confirm that MFO is an effective dimensionality reduction strategy which does not reduce predictive performance. Further studies can build upon this study by considering attention mechanisms or graph-based neural networks in order to capture more complex user, account, and flow relationships among users, accounts, and transactions. Also, online learning and streaming data adaptation would enable the design of ongoing model changes as the fraud patterns change. The study of federated learning models can also be used to support inter-institutional collaborative fraud detection without violating data privacy.

References

[1] N. Nithya and R. Kiruthika, "Impact of Business Intelligence Adoption on performance of banks: a conceptual framework," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 3139–3150, 2021.

[2] A. M. Siam, P. Bhowmik, and M. P. Uddin, "Hybrid feature selection framework for enhanced credit card fraud detection using machine learning models," PLoS One, vol. 20, no. 7, p. e0326975, 2025.

[3] A. Al-Okaily, M. Al-Okaily, A. P. Teoh, and M. M. Al-Debei, "An empirical study on data warehouse systems effectiveness: the case of Jordanian banks in the business intelligence era," EuroMed Journal of Business, vol. 18, no. 4, pp. 489–510, 2022.

[4] A. Zhang, H. Xu, and R. Liu, "Credit Card Fraud Detection Method Based on RF-WGAN-TCN," Computers, Materials and Continua, vol. 85, no. 3, pp. 5159–5181, 2025.

[5] D. Saldaña-Ulloa, G. De Ita Luna, and J. R. Marcial-Romero, "A Temporal Graph Network Algorithm for Detecting Fraudulent Transactions on Online Payment Platforms," Algorithms, vol. 17, no. 12, p. 552, 2024.

[6] G. Zioviris, K. Kolomvatsos, and G. Stamoulis, "An intelligent sequential fraud detection model based on deep learning," The Journal of Supercomputing, vol. 80, no. 10, pp. 14824–14847, 2024.

[7] R. Arjun, A. Kuanr, and K. Suprabha, "Developing banking intelligence in emerging markets: Systematic review and agenda," International Journal of Information Management Data Insights, vol. 1, no. 2, p. 100026, 2021.

[8] R. K. Gupta et al., "Enhanced framework for credit card fraud detection using robust feature selection and a stacking ensemble model approach," Results in Engineering, p. 105084, 2025.

[9] F. Atban, M. Y. Küçükkara, and C. Bayılmış, "Enhancing variational quantum classifier performance with meta-heuristic feature selection for credit card fraud detection," The European Physical Journal Special Topics, pp. 1–14, 2025.

[10] F. Ji and A. Tia, "The effect of blockchain on business intelligence efficiency of banks," Kybernetes, vol. 51, no. 8, pp. 2652–2668, 2021.

[11] M. Alazab, R. A. Khurma, A. Awajan, and D. Camacho, "A new intrusion detection system based on Moth–Flame Optimizer algorithm," Expert Systems with Applications, vol. 210, p. 118439, 2022.

[12] H. A. Al-Ababneh, V. Borisova, A. Zakharzhevska, P. Tkachenko, and N. Andrusiak, "Performance of artificial intelligence technologies in banking institutions," WSEAS Trans. Bus. Econ, vol. 20, pp. 307–317, 2023.

[13] I. A. Doush, B. Ahmed, M. A. Awadallah, M. A. Al-Betar, and N. A. Alawad, "Improving multilayer perceptron neural network using two enhanced moth-flame optimizers to forecast iron ore prices," Journal of Intelligent Systems, vol. 33, no. 1, p. 20230068, 2024.

[14] S. Umamaheswari, A. Valarmathi, and others, "Role of artificial intelligence in the banking sector," Journal of Survey in Fisheries Sciences, vol. 10, no. 4S, pp. 2841–2849, 2023.

[15] M. Doumpos, C. Zopounidis, D. Gounopoulos, E. Platanakis, and W. Zhang, "Operational research and artificial intelligence methods in banking," European Journal of Operational Research, vol. 306, no. 1, pp. 1–16, 2023.

[16] S. Gholami, E. Zarafshan, R. Sheikh, and S. S. Sana, "Using deep learning to enhance business intelligence in organizational management," Data Science in Finance and Economics, vol. 3, no. 4, pp. 337–353, 2023.

[17] R. M. Jewel et al., "Revolutionizing Organizational Decision-Making for Stock Market: A Machine Learning Approach with CNNs in Business Intelligence and Management," Journal of Business and Management Studies, vol. 6, no. 1, pp. 230–237, 2024.

[18] A. C. Nwanakwaugwu, U. O. Matthew, A. A. Kazaure, and K. Haruna, "Data Mining Business Intelligence Applications in Retail Services Using Artificial Neural Networks," in Handbook of Research on Cybersecurity Risk in Contemporary Business Systems, IGI Global, 2023, pp. 186–210.

[19] J. Fombellida, I. Martín-Rubio, S. Torres-Alegre, and D. Andina, "Tackling business intelligence with bioinspired deep learning," Neural Computing and Applications, vol. 32, no. 17, pp. 13195–13202, 2020.

[20] A. Aziz, S. Saha, and M. Arifuzzaman, "Analyzing Banking Data Using Business Intelligence: A Data Mining Approach," in Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020, Springer, 2021, pp. 245–256.

[21] X. Chen, M. Wu, and M. Wang, "Application of business intelligence under deep neural network in credit scoring of bank users," Journal of Computational Methods in Sciences and Engineering, vol. 24, no. 3, pp. 1585–1604, 2024.

[22] B. Li, B. Xiao, and Y. Yang, "Strengthen credit scoring system of small and micro businesses with soft information: Analysis and comparison based on neural network models," Journal of Intelligent & Fuzzy Systems, vol. 40, no. 3, pp. 4257–4274, 2021.

[23] W. Feng and M. Chen, "[Retracted] Application of Business Intelligence Based on the Deep Neural Network in Credit Scoring," Security and Communication Networks, vol. 2022, no. 1, p. 2663668, 2022.

[24] Sérgio Jesus, "Bank Account Fraud Dataset Suite (NeurIPS 2022)." Accessed: Aug. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

[25] J. Wang, X. Wu, and C. Zhang, "Support vector machines based on K-means clustering for real-time business intelligence systems," International Journal of Business Intelligence and Data Mining, vol. 1, no. 1, pp. 54–64, 2005.

[26] M. A. Al Montaser and M. Bannett, "Beyond anomaly detection: Redesigning real-time financial fraud systems for multi-channel transactions in emerging markets," Baltic Journal of Multidisciplinary Research, vol. 2, no. 3, pp. 1–17, 2025.

[27] N. Tyagi, "Artificial Intelligence in Financial Fraud Detection: A Deep Learning Perspective," International Journal of Computer Technology and Electronics Communication, vol. 7, no. 6, pp. 9726–9732, 2024.