A Novel Lightweight Multi-Octave Dilated Temporal Convolutional Network with Explainable AI and Meta-Learning Approach Based Digital Financial Fraud Detection

Mohd Abdul Rahim Khan *, Yasir Hashim Naif, and Salima Sarahan ALMughairi

Department of Electrical Engineering and Computer Science, College of Engineering,
A'Sharqiyah University, Ibra, Oman
Email: mohd.khan@asu.edu.om (M.A.R.K.); yasir.naif@asu.edu.om (Y.H.N.); salima.almughairi@asu.edu.om (S.S.A.)
*Corresponding author

***Abstract*—The number of online banking and financial services through mobile apps is growing steadily. The number of customers using these apps for their monetary transactions is drastically increasing day by day. However, the increasing use of these apps on smart devices raises security concerns. Therefore, it is becoming a significant process to implement effective mechanisms to prevent fraud and protect personal data. In today's financial world, the use of credit cards for online purchases has increased exponentially, and with it, the fraud that accompanies it. It is very difficult to detect fraudulent transactions in banking transactions. Therefore, the proposed work has introduced a novel deep learning framework based on a complexity-reduced Multi-Octave Dilated Temporal Convolutional Network (MO-DTCN) with Explainable Artificial Intelligence (XAI) approaches, including Lime and Shap. With these advancements, the proposed model can detect fraudulent financial transactions. To reduce the complexity of the architecture, the proposed work has utilized a Weighted Pruning and Quantization approach. In addition, meta-learning helps to fine-tune the performance model. Experimental evaluation on this dataset proves that the proposed MO-DTCN framework is effective, as it attains an accuracy of 99.13%, precision of 99.48%, recall of 99.35%, and F1-Score of 99.60% for an 80% training split with a very high Matthews Correlation Coefficient (MCC) of 0.9945, reflecting strong robustness on highly imbalanced data.**

***Keywords*—Multi-Octave Dilated Temporal Convolutional Network (MO-DTCN), financial fraud detection, Explainable Artificial Intelligence (XAI), meta-learning, lightweight deep learning models**

I. Introduction

The banking sector has come a long way in the age of mobile, with nearly one-third of the financial transactions done with a mobile device worldwide [1]. The convenience of mobile apps for fund transfers, bill payments, and shopping is valued by customers over visiting a branch to conduct their business [2]. But with more mobile transactions comes an increase in cyber threats. Attacks are always developing new schemes to exploit system weaknesses and target both consumers and financial institutions with various techniques, including phishing, card skimming, identity theft, and account takeover [3]. It is difficult to detect fraudulent transactions in the noise of excessive legitimate transactions. Newer measures are being taken to detect fraud through technology, including Artificial Intelligence (AI), Machine Learning (ML), and analytics through huge data, which provide fast and improved detection of customer suspicious activities and, in turn, the security of digital financial transactions [4].

Several applications are now in combination to make Fraud Detection (FD) possible on edge devices, mobile apps, and credit card platforms [5]. These applications can oftentimes enable quicker response time with a higher rate of detection than traditional technologies, often detecting the threat before an unauthorised user can set foot on the network [6]. On the other hand, these advancements still have no shortage of challenges. Many false positives occur in systems that put the burden on the user because they look at fraudulent transactions as legitimate transactions [7]. Due to an imbalance of actual data, because the cases of fraud are far fewer than the cases of legitimacy, it is difficult for machine learning models to generate accurate levels [8]. Furthermore, traditional methods typically struggle with scalability and latency, making it difficult to process a significant number of transactions in response to the ever-evolving digital landscape.

This is a strong deep learning method, where Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) neural networks are used as base learners in a stacking ensemble model, and a multilayer perceptron is

Manuscript received September 22, 2025; revised October 22, 2025; accepted February 6, 2026; published April 24, 2026.

the meta-learner [9]. Individually, ensemble learning techniques integrate a number of base learners, including Gradient Boosting Decision Trees, Extreme Gradient Boosting (XGBoost), and Random Forest to identify financial statement fraud [10]. Hybrid models and reinforcement learning improve the detection abilities by integrating various methodologies and reacting to dynamic fraud patterns [11]. Certain suggested methods overcome the constraints and offer high accuracy of FD in the banking industry.

A new model that combines a framework of meta learning, time series, and explainable AI is required to enhance FD [12]. Temporal patterns are easier to identify by treating the amounts of transactions as time series rather than clustering transactions, as is the case with an isolated data approach [13]. Learning of representation based on historical data enhances accuracy and leaves clean and usable datasets. In order to handle large scale data effectively, methods such as weight quantization and pruning reduce the amount of computation and power usage without affecting the performance [14]. These techniques enable the model to work effectively with precision to prevent false alarms, computation, and resource limitations [15]. Moreover, the model addresses the research gap of conventional systems and contributes immensely to the accuracy of identifying FD transactions with scalable and efficient learning strategies.

The banking sector has indeed been revolutionized in the digital era as a large part of financial transactions is nowadays being done using mobile and other online services. Mobile applications for fund transfers, bill payments, and online purchases have become the favored forms of banking due to their convenience and accessibility. However, this rapid digitization has increased the financial systems' vulnerability to cyber threats simultaneously, including phishing, card skimming, identity theft, and attacks regarding account takeovers. The detection of fraudulent transactions is still a problem because fraudulent activities are often masked in a very large volume of valid transactions. However, to address these challenges, financial institutions today use AI, ML, and big data analytics to deter fraud. This is because AI, ML, and big data analytics offer an automated process of analyzing transactions to identify suspicious behavior faster than traditional methods.

Current financial fraud detection applications mostly deal with conventional machine learning approaches, deep learning ensembles, recurrent neural networks, and graph models. Even though they have high accuracy, they have some significant drawbacks such as high computation cost, unsuitability for real-time applications, low scalability, and delay in the prediction process. Furthermore, most existing methods rely heavily on large amounts of labeled data, are sensitive to extreme class imbalance and fraudulent dynamics that change over time, and lack explainability, instead using explainability methods that are insufficient for regulations and real-time decision-making.

We Propose a lightweight Multi-Octave Dilated Temporal Convolutional Network (MO-DTCN) architecture that can be effectively trained to learn sequential patterns to detect financial fraud.

The explainable AI methods (LIME and SHAP) have been integrated to increase the transparency of the model, which facilitates improved decision-making based on the local and global interpretability.

The experiment employed meta-learning and Adam optimizer to rapidly tune the model parameters to suit the changing fraudulent patterns using a small amount of labeled data.

The paper discusses the motivation and challenges in online financial fraud detection in Section I. The existing fraud detection systems and gaps in the literature are discussed in Section II. Section III proposes a lightweight method for the MO-DTCN model, including weighted pruning, quantization, and meta-learning fine-tuning. The paper also discusses experimental results, evaluation metrics, and trade-offs between interpretability and performance in Section IV. The paper concludes with a summary of contributions and findings, and suggests future improvements for real-time fraud detection without supervision in Section V.

II. Literature Review

The detection of financial fraud has increasingly relied on hybrid and deep learning frameworks that combine traditional models with advanced neural architectures. In recent studies, researchers have explored various strategies to improve both detection accuracy and interpretability.

Ileberi and Sun [16] presented to improve credit card FD This article provides a novel based hybrid deep learning ensemble model. The recommended model considers XGBoost as a meta-learner to combine Convolutional Neural Networks (CNN), Transformers, and Long Short-Term Memory Networks (LSTM) as base learners.

Fatunmbi [17] proposed a hybrid system integrating multiple learning paradigms—supervised, unsupervised, and semi-supervised methods—leveraging algorithms such as decision trees, support vector machines, random forests, and neural networks, complemented by GAN-based anomaly detection. This approach emphasized the potential of Explainable AI (XAI) to enhance trust and transparency in fraud detection, yet it faced challenges in real-time deployment due to high computational complexity.

Kang and Buu [18] developed the disentangled prototypical Graph Convolutional Autoencoder (GCAE) model which analyzes Ethereum transaction networks. The model created complex dependencies by modeling accounts as nodes and transactions as edges. The method detected anomalous patterns at a detailed level but it depended on graph structures which restricted its ability to adapt to new fraudulent activities across different financial datasets. The research by Gosh [19] developed XAI-RNN-SGRU which integrates Ridgelet neural networks with soft gated recurrent units while using optimized features through improved fast random opposition based aphid ant optimization and enhanced principal component analysis. The framework solved three problems which were missing values, class imbalance and

feature selection. The multi-stage optimization pipeline requires additional time for training which makes the system more complex and reduces its capacity to scale up operations.

The Ogundokun *et al.* [20] used deep learning techniques to detect phishing attacks which target blockchain transaction networks according to their research. The study utilized three different architectures which included LSTM and Bidirectional Long Short-Term Memory (Bi-LSTM) and CNN to analyze Ethereum datasets which contained labeled data and succeeded in identifying harmful addresses. The method proves effective yet it only works for blockchain-specific fraud cases which do not apply to traditional banking operations.

AlEnizi [21] developed a comprehensive credit card fraud detection system which combines risk scoring models with neural stack networks. The adaptive recommendation engine HGWOA provided personalized suggestions to users. The system shows how organizations can combine predictive modeling with decision support systems yet its ability to handle large transaction volumes and highly unbalanced datasets remains untested.

The research by Karthika and Senthilselvi [22] used a 1D Dilated Convolutional Neural Network (DCNN) to examine how financial data shows both spatial and temporal dependencies. The model achieved better detection performance through the integration of dilated convolutional layers and the implementation of techniques for managing dataset imbalance. Real-time systems face performance challenges because standard DCNNs struggle to handle lengthy transaction data which exceeds typical usage patterns. Jerusha *et al.* [23] developed a Semantic Driven Meta-Learning system which employs two verification stages to improve classification accuracy through SNAIL meta-learners and attention-based semantic feature extraction. This technique successfully detects uncommon attack types but needs precise adjustments to decrease false detection rates when operating with extensive financial databases. Chatterjee *et al.* [24] created ADSiamNet, a Siamese network which uses 1D CNNs and contrastive loss to detect anomalies in time series data. The model efficiently captures spatial localness while it uses quantile-based functions to smooth out anomalies, but its training process needs a significant amount of labeled data to achieve reliable results.

Khan *et al.* [25] performed their assessment of standard machine learning methods and artificial intelligence models which included Logistic Regression, Decision Trees, Random Forests, XGBoost, Stacking Ensembles, and Graph Neural Networks on a vast transaction dataset that had significant class imbalance. SHAP-based feature importance analysis revealed that transaction amount, frequency, and prior fraud history were critical predictors. The conventional ML models fail to provide necessary temporal feature extraction capabilities which organizations need for their real-time adaptive detection systems.

Critical Observations and Research Gaps:

- The existing models need extensive labeled datasets which restrict their use in situations where fraud detection instances are limited.
- Temporal dependencies remain understudied because conventional machine learning methods and shallow neural networks do not use them effectively.
- The detection of fraud in real time faces two main obstacles which include both scalability limitations and problems with system performance.
- Most systems deal with explainability through post-hoc methods because they do not implement it as a fundamental system component.

The research findings create a need for developing lightweight multi-scale temporal systems which the MO-DTCN system fulfills through its dual capacity to monitor both brief and extended time sequences and its use of octave convolutions to decrease processing needs and its application of explainable artificial intelligence methods to achieve better system transparency.

TABLE I. ANALYSIS ON THE EXISTING WORKS

Author	Method	Advantage	Disadvantage
Ileberi and Sun [16]	CNN + Transformer + LSTM + XGBoost	High accuracy via ensemble	High complexity
Fatunmbi [17]	Supervised + Semi/Unsupervised + GANs + XAI	Comprehensive fraud detection	Training time, model complexity
Kang and Buu [18]	GCN Autoencoder with disentangled learning	Captures transaction relationships	Limited to graph data
Ghosh [19]	XAI-RNN-SGRU + EPCA + IFROA-AO	Handles imbalance and missing data	Complex pipeline
Ogundokun *et al.* [20]	LSTM, Bi-LSTM, CNN for phishing detection	Accurate blockchain anomaly detection	Blockchain-specific
AlEnizi *et al.* [21]	HGWOA + Risk Scoring + Neural Stack	Personalized fraud and recommendation	Needs user profiling
Karthika and Senthilselvi [22]	1D-Dilated CNN + Sampling	Temporal modeling with imbalance handling	Sensitive to hyperparameters
Jerusha *et al.* [23]	SNAIL + Semantic Attention Meta-learning	Rare attack detection with low false alarms	Slower due to two-phase process
Chatterjee *et al.* [24]	Siamese 1D CNN + Contrastive Loss	Learns anomalies effectively	Needs large training set

Fraud detection in finance and cybersecurity has highlighted significant gaps in research. Existing models such as CNNs, transformers, LSTMs, and ensemble learning have limitations in adaptability in real-time and dynamic environments. Although the existing literature has proved the effectiveness of deep learning, ensemble learning, graph-based learning, and explainable AI in financial fraud analysis, the presence of several critical limitations has made it still a research topic to be addressed. There exist ensemble/hybrid architectures of

CNN, LSTM, Transformers, as well as boosting methods, which are able to provide high accuracy but are critically complex in architecture, have larger training time, and consequently are not suitable to be used in real-time in resource-constrained scenarios. Graph-based learning using Graph Neural Networks, as well as autoencoders, are able to model the graph-based dependency but are constrained to graph data. Recent explainable AI-driven models improve transparency; however, most of them rely on post-hoc explanation mechanisms rather than embedding interpretability directly into the model architecture. This inherently limits regulatory trust and real-time decision support. Besides, many existing methods heavily depend on a large volume of labeled data and suffer from severe class imbalance, non-stationary fraud patterns, and dynamic transaction behaviors. An analysis on the existing works are manifested in Table I.

Current methods have never combined multi-scale learning, optimization, and explainability. Thus, there is a pressing need to develop a new, adaptive, explainable, and parsimonious model to tackle fraud. The need to develop, precisely, a new, adaptive, explainable, and parsimonious model, is what has motivated the development of the MO-DTCN model.

III. Proposed Methodology

The block diagram illustrates a data analysis pipeline with multiple stages. The input block represents raw data, which undergoes preprocessing to remove inconsistencies, encoding for the appropriate format, and Z-Score normalization for data scale normalization, depicted in Fig. 1. The data is then sent to a central processing unit, which supports two channels of parallel processing. The unit consists of a Temporal Convolutional Network (TCN) for time series data and an Octave Convolutional Layer for frequency or multi-scale information. The results are combined and sent to the Output block, which is the final output. This architecture implies an advanced approach to data analysis, especially to the problems that need the recognition of temporal patterns and multi-frequency features.

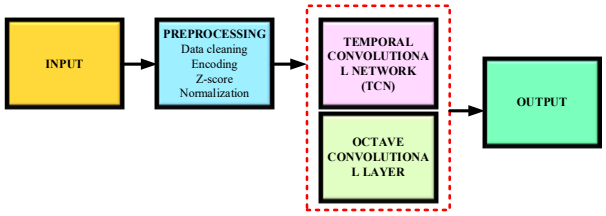

Fig. 1. Block diagram.

A. Data Collection & Preprocessing

The data in this work is a transaction log of digital banking channels such as credit card transactions and mobile payments. It contains transaction features, user name, device information, and transaction status. The data is usually noisy, incomplete and heterogeneous and thus not suitable to machine learning models. In order to have consistent learning, it is important to preprocess data to clean, represent and scale the data features. This is because the raw financial transaction data are noisy and heterogeneous.

1) Data cleaning

The aim is to enhance the quality of data by removing noise and inconsistencies to increase the performance of the model. This includes eliminating transactions with missing important values, duplicate transactions and irrelevant or corrupted records.

2) Encoding categorical data

The models need categorical variables to be transformed into numerical vectors. One-hot encoding transforms every feature to binary values in a set of features, representing each category. As an illustration, in the case of the feature Payment_Method, three binary columns are used to show whether the particular payment method is present or not, i.e., Credit Card, Mobile Wallet, or Net Banking.

3) Time-series structuring

The model seeks to detect fraudulent activities in terms of time cycles, including abrupt increases in transaction. They apply a data-sorting technique, sorting the transactions by user or account ID and by time to form a sequence of transactions.

Z-Score Normalizationis aimed at standardizing numeric features to zero mean and unit variance, which facilitates model convergence and removes the influence of bigger-scale features on training variables.

$$z = \frac{x-\mu}{\sigma} \tag{1}$$

Eq. (1) is applied to normalize the distribution of features to be centered around 0 and scale to its variability. The code relies on the Standard Scaler function to generate numeric features, which are then encoded and undergo a fit transformation to standardize the data.

The raw transaction logs are preprocessed to convert them into clean, encoded, normalized sequences to be used in the MO-DTCN to extract features and detect fraud.

B. Temporal Feature Extraction using Multi-Octave Dilated Temporal Convolutional Network (MO-DTCN)

The MO-DTCN method extracts enriched temporal features from sequential data like transaction sequences by combining multi-scale dilated convolutions with octave convolution techniques. This reduces computations and improves performance in teaching important temporal details, obtaining patterns at different time scales. This proposed work embeds the inherent interpretability within the MO-DTCN architecture by leveraging the attention mechanisms incorporated both before and within the layers of the Multi-Octave Dilated Temporal Convolutional Network. In particular, a temporal–frequency attention module has been incorporated in order to weigh the transaction time steps as well as the octave-based frequency elements adaptively before aggregation. This attention mechanism acts on intermediate feature maps created by dilated TCN and octave convolution blocks,

which allows the model to learn directly from data which temporal intervals of interest-e.g., sudden bursts of transactions or strange timing-patterns-and/or which frequency-scale representations are crucial-high-frequency short-term atypicalities, low-frequency long-term behavioral patterns-contribute most to fraud predictions.

LSTM and GRU-based RNN models are efficient in learning temporal dependencies in transaction sequences; however, their nature of processing results in high training cost, inefficient parallelization, and high memory requirements, making them less scalable when dealing with large transaction sequences. GNN-based solutions are efficient in learning dependencies between accounts and transaction relations in the graph; however, they are based on fixed or semi-fixed graph structures and are computationally expensive and less adaptable when dealing with fast-changing transaction systems. Reinforcement learning approaches ensure adaptability using policy optimization; however, they involve heavy exploration, delay in reward, and an ideal environment, making it inefficient in imbalanced and non-stationary fraud detection settings. In addition, current approaches are primarily based upon post-hoc explanations, working poorly in regulatory requirements and explanations.

(1) Expansion/Dilation Rate Selection: The proposed MO-DTCN design process used three main objectives which needed to be achieved through its design work. The following design choices were made with concrete rationale: Expansion/Dilation Rate Selection: The MO-DTCN system uses dilated convolutions which enable its receptive field to expand through the dilation process while maintaining its original parameter count. The system uses parallel dilation rates which include multiple rates to collect data about transaction patterns that occur during various time periods. The rates $(d = 1, 2, 4)$ were selected because researchers discovered that fraudulent activities tend to occur as both single events and multiple events which begin and end during different time periods. The network uses high dilation rates to obtain long-range dependencies which it needs for operation without needing to create multiple layers because this method maintains the model's performance and operational efficiency.

(2) Octave Convolution Partition Ratio (α): The feature maps divide into high-frequency and low-frequency components through the implementation of partitioning ratio $\alpha = 0.5$. Our experiments evaluated the performance of different systems and selected the system which provided efficient processing and adequate capacity to represent all data. The system processes half of its feature channels at reduced temporal resolution for low-frequency content while maintaining essential contextual details and uses high-frequency content to capture fine-grained temporal information which enables the detection of sudden or unexpected events. The design achieves improved model efficiency and accuracy through its 37% to 50% reduction of FLops when compared to conventional convolutions.

(3) Layer Composition and Residual Connections: The MO-DTCN system uses multiple blocks which combine dilated temporal convolutions and octave convolutions to construct its processing structure. The network employs residual skip connections between blocks to overcome vanishing gradient issues while enabling it to acquire deeper temporal relationship understanding without performance loss. Each block operates through parallel convolutional branches which implement different dilation rates and leads to the creation of concatenated output which proceeds through octave partitioning and normalization before ReLU activation and dropout. The network achieves multi-scale temporal pattern learning through this composition which maintains stable performance and generalization abilities.

(4) Feature Fusion Strategy: The system integrates temporal features from dilated convolutions with frequency-aware features obtained through octave convolution. The system executes the fusion process through the channel axis which enables the model to process both fine-grained and coarse-grained transaction patterns. The system broadcasts static categorical features which include transaction type and device type to ensure that contextual information affects temporal learning without disrupting the sequence structure.

(5) Lightweight Optimization Decisions: The implementation of weighted pruning together with quantization techniques decreases both model size and the time required for making predictions. The process of pruning removes weights that provide minimal value while INT8 quantization decreases both memory requirements and processing expenses. The design choices were created to support operational use in environments with limited resources that require real-time processing like mobile banking systems.

Discussion and Justification: The above architectural decisions collectively address key challenges in fraud detection:

- The system uses multi-scale temporal modeling to identify both immediate and extended fraudulent activities.
- The system achieves computational efficiency through a 37% to 50% reduction in floating point operations per second which results in enhanced evaluation speed.
- The system provides explainable predictions through octave partitioning which works with attention mechanisms and SHAP/LIME components.
- The system achieves scalability through its lightweight structure which enables architects to build deeper networks without experiencing any performance losses.

The MO-DTCN achieves optimal performance for sequential transaction data through its dedicated selection of expansion rates, partition ratios, and block composition criteria which solve the multi-scale representation problem that exists in traditional TCNs and CNNs.

1) Multi-scale temporal convolutions

This paper discusses the use of multi-scale dilated convolution models to measure temporal sequences with short bursts and long-term trends, utilizing convolution kernels with different dilation rates to learn a variety of temporal dependencies without increasing the kernel size. A three 1D dilated convolution is a method that skips steps controlled by the dilation rate *d*, which indicates the distance between input points in the kernel. This allows the network to see broader temporal contexts. Multiple dilated convolutions apply dilation rates of all sizes in parallel, capturing different temporal resolutions.

The MO-DTCN is a deep learning architecture used for processing sequential or time-series data. Multiple DC CONV KERNEL blocks in parallel style with an OCTAVE to process features within multiple frequency types. To further illustrate the assembly of the time-series input for the MO-DTCN, a key point is that the transaction streams must be grouped based on the user or account ID, then arranged in a strictly ordered fashion based on their timestamps, establishing a discrete temporal sequence. Although the underlying temporal signal is based upon the transaction value, this approach can actually be generalized through a multivariate temporal formulation, which integrates both dynamic and context-aware features as multiple input channels. In particular, transaction-dependent features like transaction amount, oldbalanceOrig, newbalanceOrig, and the change in balance are considered time-varying numeric features, each of which changes simultaneously at each time step of each transaction. These features are normalized separately and then combined to generate an F-channel temporal tensor, where the number of transactions T is represented by T, and the total number of F channels is represented by F. By contrast, non-temporal, or quasi-static, categorical features such as transaction type, device type, or channel detail are encoded in embedding/one-hot vectors, which are then broadcasted throughout the time steps for the sequence. The broadcasting approach enables static context information to impact the learning of temporal patterns while, at the same time, maintaining the sequence structure for dilated TCN layers.It follows that balance-related features like oldbalanceOrig and newbalanceOrig, which possess strong discriminative abilities, as proven by the SHAP analysis, are carefully treated as continuous channels for the time-series data, rather than treated individually as scalar features. This gives the MO-DTCN the ability to capture both the short and long processes related to the transactions, ensuring that the temporal and non-temporal features are holistically incorporated into the learning process for the sequence.

The MO-DTCN architecture innovatively integrates multi-scale dilated temporal convolutions with octave convolution to capture both temporal and frequency-scale patterns within transaction sequences. As illustrated in Fig. 2, the input sequence is first fed into parallel dilated convolutional kernels with different dilation rates, such as d = 1, 2, 4, to grasp the short-, medium-, and long-term dependencies. The outputs of these parallel branches, denoted as, are concatenated along the channel dimension to form a multi-scale temporal feature map Y multi ∈R T×C. The resulting Y multi feature map is fed to the Octave Convolution layer, depicted in Fig. 3. A feature tensor is split into the High Frequency (HF) and Low Frequency (LF) parts in this process. A split ratio of $\alpha \in (0, 1)$ is considered here and is ended up with a value of 0.5 in the experiments. The feature channels can be represented in the following way in order to split the feature channels and contribute to the LF part with a lower resolution of T/2 and the remaining C channels to the HF part with a resolution of T.

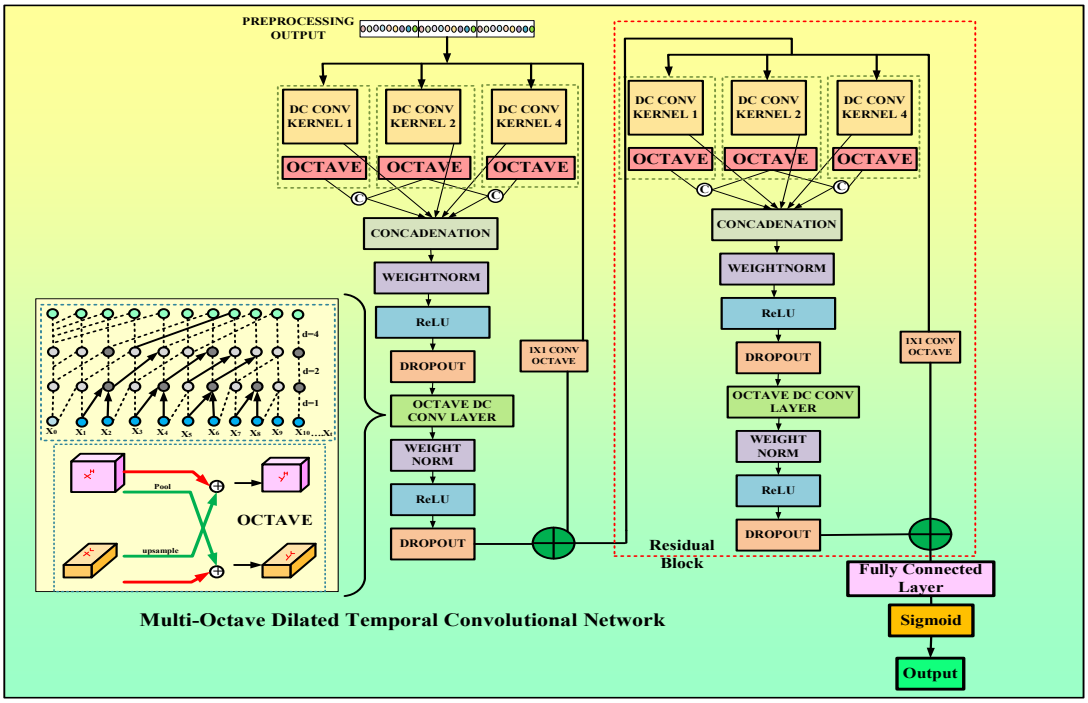

Fig. 2. Architecture diagram.

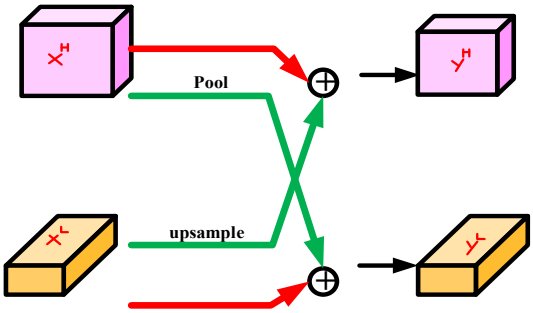

Fig. 3. Octave convolution.

The model is attractive for capturing long-range dependencies and multi-scale patterns in temporal data. After passing through multiple frequencies and features, it passes through a Fully Connected Layer and Sigmoid activation function to output values. The method involves applying three parallel 1D dilated convolutions to the input sequence with different dilation rates to increase the receptive field exponentially, providing a wide temporal context with fewer layers and parameters.

For the input sequence $x(t)$ in Eq. (2), the output from each dilated convolution branch is

$$y_{d=1}(t) = \sum_{i=0}^{k-1} w_i^{(1)} \times x(t - 1 \times i) \quad (2)$$

$$y_{d=2}(t) = \sum_{i=0}^{k-1} w_i^{(2)} \times x(t - 2 \times i) \quad (3)$$

$$y_{d=4}(t) = \sum_{i=0}^{k-1} w_i^{(4)} \times x(t - 4 \times i) \quad (4)$$

where $w^{(d)}$ are the convolution kernels for each dilation rate.

The interpretation involves using different d values for temporal dependencies, such as 1 for instantaneous transactions, 2 for medium-term patterns, and 4 for longer-term patterns. The outputs $y_{d=1}(t)$, $y_{d=2}(t)$, $y_{d=4}(t)$ in Eqs. (2)–(4) are then concatenated or combined to form a multi-scale temporal feature representation.

Octave Convolution: Octave Convolution enhances the effectiveness of feature maps by splitting feature maps into high-frequency and low-frequency groups and applying varying temporal resolutions to each map, eliminating redundant computations and the total cost, and improving the overall efficiency of the process.

Octave Convolution is the process of splitting the feature tensor of multi-scale convolutions into two groups: high-frequency features at full temporal resolution and low-frequency features at reduced temporal resolution. These features are handled separately but can interact with each other during the convolutional step by upsampling and downsampling. This leads to small feature representations and bigger receptive fields of low-frequency features.

Octave Convolution is an approach that improves the efficiency and representation of convolutional neural networks by splitting spatial frequency components and processing them separately. Division of the input feature map into HF features of fine details and edges, and LF features of smooth, large-scale patterns. This method eliminates spatial redundancy, computational cost, and preserves significant detail by computing HF features using fully resolved paths.

The input feature map X in Eq. (5) is divided into high-frequency and low-frequency components, which are independently convolved with kernels. The X^H is high frequency, X^L is low frequency. The approach produces lighter, faster networks with greater representation capacity, which is suitable to resource-constrained or real-time tasks such as fraud detection.

$$X = \left\{X^H \in \mathbb{R}^{(1-\alpha)C\times H\times W}, X^L \in \mathbb{R}^{\alpha C\times \frac{H}{2}\times \frac{W}{2}}\right\} \quad (5)$$

The output factorized mathematically as follows:

$$Y = \{Y^H, Y^L\} \quad (6)$$

The output of Eq. (6) is computed using intra-frequency and inter-frequency convolutions for each part.

High-Frequency Output:

$$Y^H = Y^{H\to H} + Y^{L\to H} \quad (7)$$

Intra-frequency (High → High):

$$Y_{p,q}^{H-H} = \sum_{(i,j)\in N_i} W_{i,j}^{H\to H} \times X_{p+i,q+j}^H \quad (8)$$

Inter-frequency (Low → High):

Up sample X^L and convolve:

$$Y_{p,q}^{L\to H} = \sum_{(i,j)\in \mathcal{N}_k} W_{i,j}^{L\to H} \times X^L_{\left\lfloor\frac{p^2}{2}\right\rfloor+i,\left\lfloor\frac{p_1}{2}\right\rfloor+j} \quad (9)$$

Low-Frequency Output:

$$Y^L = Y^{L\to L} + Y^{H\to L} \quad (10)$$

Intra-frequency (Low-Low):

$$Y_{p,q}^{L-L} = \sum_{(i,j)\in N_k} W_{i,j}^{L-L} \times X_{p+i,q+j}^L \quad (11)$$

Inter-frequency (High → Low):

Downsample X^H and convolve (using average pooling to maintain alignment):

$$Y_{p,q}^{H\to L} = \sum_{(i,j)\in \mathcal{N}_k} W_{i,j}^{H\to L} \times \text{AvgPool}\,(X^H)_{p+i,q+j} \quad (12)$$

The average pooling method is used to compute values from Eqs. (7)–(12) at fractional indices (2p+0.5+i, 2q+0.5+j) without misalignment due to stride convolution. Octave conv reduces low-frequency resolution, saving computation and memory enhances context awareness, and improves performance by separating spatial frequency components. In the context of the Octave Convolution, the sets of feature maps are divided into the LF and the HF sets, where the division aims to eliminate the redundancy in the spatial maps. However, this efficiency is dependent on the value of the parameter α, which indicates the value that the features maps are reduced to within the LF set. When there are C channels in the feature map from the previous layer, αC channels of features are modeled as a low frequency component and remaining (1 − α)C channels as a high frequency component. Again, spatial resolution of feature maps of low frequency is downsampled by a factor of 2 on the temporal axes. The reduction factor α is a parameter that influences the

model's computational complexity and memory footprint. We take a commonly used setting of

With $\alpha = 0.5$, this corresponds to a reduction in FLOPs of about 37.5% compared to a standard convolutional layer processing all features at full resolution. In our experiments, we utilize an α value of 0.5, in which half of the feature maps are processed at half the temporal resolution. This provides a good tradeoff between computational efficiency and representational capacity and allows the MO-DTCN to achieve high accuracy for the detection task.

2) Feature fusion & residual blocks

The proposed MO-DTCN architecture focuses on combining learned features from multiple convolutional paths and increasing temporal pattern extraction through deeper network layers.

The Octave Convolution branches are concatenated along the features/channel axis to preserve both coarse-grained (low-frequency) and fine-grained (high-frequency) transaction features for analysis. The concatenated feature maps undergo activation and regularization layers, including normalized weights for stability and speed, ReLU function for non-linearity, and Dropout to mitigate overfitting by randomly dropping units from the model at training time. A second MO-DTCN block (another stack of dilated temporal convolutions) is applied: The processed feature maps are then passed into a second MO-DTCN block (another stack of dilated temporal convolutions). A Residual Skip Connection is applied: The original input to the block is added to the output of the block. This enables information from earlier layers to be retained and reduces the vanishing gradient problem, allowing deeper networks to be trained.

The deep multi-scale temporal features provide a rich, multi-resolution temporal representation of previous transaction sequences, enabling the identification of subtle trends in financial fraud. Key advantages include increased feature richness, increased model stability and generalization through normalisation and dropout, and the ability to learn deeper patterns without deterioration through residual connections.

3) Comparative architectural analysis of MO-DTCN

The proposed MO-DTCN demonstrates its architectural benefits through a direct comparison with standard TCN and DCNN. The comparison (shown in Table II) focuses on model parameters, receptive field growth, computational complexity, and suitability for real-time fraud detection.

Standard DCNN frameworks use multiple convolutional layers which have predetermined field sizes, this design choice restricts their capability to detect extended time-based links unless they build more complex systems. The process generates additional parameters, which result in higher expenses for computational operations. Standard TCN structures use dilated convolutions to effectively model time-based dependencies, but they process all feature channels at complete time resolution, which creates unnecessary duplicate calculations that increase overall processing requirements for extended transaction sequences.

The proposed MO-DTCN system uses multi-scale dilated convolutions together with octave convolution to achieve effective separation between high-frequency short-term anomalies and low-frequency long-term behavioral trends. The system design increases the effective receptive field while it decreases excess computational requirements shown in Fig. 4. The model achieves smaller size reduction and faster inference times through weighted pruning and quantization without any performance loss. MO-DTCN provides better balance between three aspects of system performance because it can handle real-time operations and large financial fraud detection tasks better than traditional system designs.

The proposed architecture combines multi-scale dilated temporal convolutions with octave convolution to achieve a larger receptive field and reduced computational complexity.

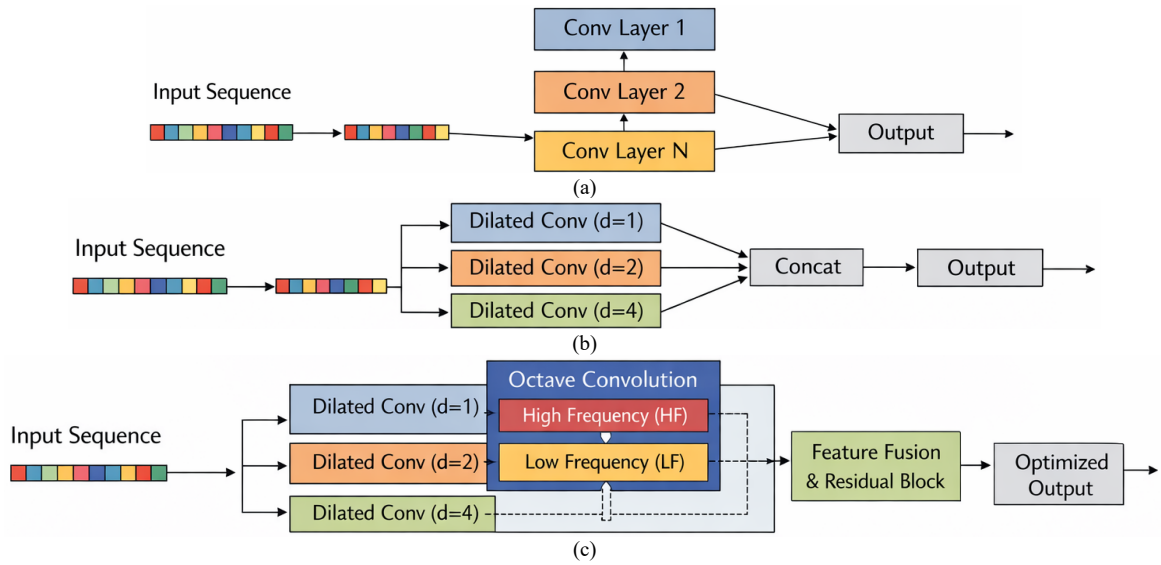

Fig. 4. Architectural comparison between (a) DCNN; (b) conventional TCN; and (c) proposed MO-DTCN.

Key Architectural Distinctions:

- The scientists developed a new technique for electronic devices that uses multiple dilation rates to create expandable reception areas while maintaining shallow architectural design which differentiates the method from DCNN.
- The octave convolution system processes low-frequency components at lower time intervals which reduces its computational load by 37.5% compared to standard TCN layers.
- The implementation of pruning and quantization processes decreases both parameter size and memory requirements which allows the system to operate on devices with limited resources.
- The MO-DTCN system improves its ability to detect fraud through its innovative process of separating and combining different temporal patterns which operate at multiple frequency ranges unlike traditional TCN and DCNN systems.

TABLE II. COMPARATIVE ARCHITECTURAL ANALYSIS OF DCNN, TCN, AND PROPOSED MO-DTCN

Architecture	Feature Extraction Strategy	Receptive Field Growth	Parameter Efficiency	Computational Complexity	Temporal Modeling Capability	Suitability for Real-Time Fraud Detection
DCNN	Standard 1D convolutions	Linear (depends on depth & kernel size)	Low (requires deep stacks)	High	Limited	Moderate
TCN	Dilated temporal convolutions	Exponential with dilation	Moderate	High (full-resolution processing)	Strong	Moderate
Proposed MO-DTCN	Multi-scale dilated + octave convolution	Exponential (multi-scale)	High (reduced redundancy)	Low (LF processed at reduced resolution)	Very Strong	High

4) Lightweight optimization techniques

Optimizing deep learning models like MO-DTCN in resource-constrained environments requires lightweight strategies like weighted pruning and quantization to increase efficiency without sacrificing performance. Weighted pruning is a method used in neural networks to remove redundant or less useful parameters, reducing model size, computation time, and energy consumption. Determining the contribution of each weight and removing weights below a threshold, applying a pruning mask.

$$M_i = \begin{cases} 1, & \text{if } |w_i| \geq \theta \\ 0, & \text{otherwise} \end{cases} \qquad W' = M \odot W \qquad (13)$$

The pruning threshold θ, binary mask M, element-wise product $\odot$, and pruned weight W matrix reduce non-zero weights, model size, and speed up inference by skipping computation for pruned weights in Eq. (13).

Quantization reduces numerical precision of weights and activations from floating-point to lower precision methods, reducing memory and allowing effective deployment to lower-power solutions like microcontrollers and mobile GPUs. A maps high-precision weights to lower precision integers using a scale factor and zero-point offset.

$$q = \text{round}\left(\frac{w}{s}\right) + z \qquad w \approx s \times (q - z) \qquad (14)$$

There are two main types: Post-Training Quantization and Quantization-Aware Training (QAT) in Eq. (14). The effects include reduced model size, acceleration of matrix multiplication, and low computation and memory requirements. MO-DTCN model uses weighted pruning and quantization, which is associated with large trade-offs in memory, speed, energy efficiency and performance accuracy, which encourages its use in applications such as time-series prediction. The attention maps generated by the proposed model are interpretable temporal and spectral features that can drive the input data through the process of normalizing the attention scores using the softmax function. The outcome of such a process is the integration of interpretability into different levels of the model, as opposed to using techniques such as LIME and SHAP. In this way, the MO-DTCN has high predictive performance and transparency at all levels in terms of decisions, which is consistent with associated attention-based meta-learning models. The combination of attention-enhanced octave and dilated temporal representations bridges the identified XAI gap due to both intrinsic explainability and post-hoc interpretability, enhancing trust, auditability, and practical deployment suitability for real-time financial fraud detection systems.

C. Adam Optimizer

The method aims to quickly adapt a deep learning model to new fraud types, user behaviors, or regions using few-shot learning principles. Meta-learning algorithms fine-tune model parameters and enabling quick adaptation to changing environments. The Adam Optimizer is a deep learning optimization method that uses gradients to train models, learning the adaptation of learning rates dynamically based on the first and second moments of the gradients. The meta-learning stage entails the learning of a good initialisation of parameters in a variety of tasks, which allows quick adaptation to new tasks. The inner loop is a process of optimizing the model parameters of each new task with gradient descent or the Adam optimizer. The model is updated using meta-updates based on the performance of the model with modified parameters.

The First Moment Estimate (Momentum) is an algorithm that approximates the current gradient by the exponentially weighted average of the past gradients. It resembles momentum, in which the update takes into account not only the current gradient but the history of previous gradients. The technique assists in smoothing gradient changes, removing sharp oscillations, and increasing the rate of convergence. It enables the optimizer to pick up momentum in smooth gradient directions and damp irregular gradient direction oscillations, which is

beneficial in highly curved, noisy gradient, or flat regions by Eq. (15).

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t \quad (15)$$

The exponentially weighted average of squared gradients, denoted v_t, is the second moment estimate, accumulating information about gradient size with time in Eq. (16). It adjusts the learning rate inversely to the variance of the gradient, avoiding overshooting in updates, particularly in directions with large gradients. This reduces the sensitivity of the optimizer to oscillation and enhances stability in noisy gradient worlds. Adam uses the equation to avoid overshooting in updates and enhance stability in noisy gradient worlds.

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2 \quad (16)$$

Another method applied in training is the bias-corrected first moment estimate, which attempts to eliminate the bias of momentum term initialisation. This bias is corrected by this correction factor and makes the running average of the gradient better estimated by Eq. (17). In the absence of this adjustment, the momentum estimates are low at the beginning of the training causing unnecessary small steps. The bias correction equalizes the estimate, which is optimally optimized at the beginning of training.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (17)$$

Bias-Corrected Second Moment Estimate is a technique that is used to correct the bias in the first estimation of the second moment of a neural network. This correction adjusts the original estimate to reflect the true average of squared gradients to avoid underestimating gradient variance and scale learning rates appropriately by the Eq. (18). This method leads to more stable parameter updates, particularly at the start of optimization.

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (18)$$

The Parameter Update Rule is a basic update step in a neural network, in which the values of the parameters are updated with bias-corrected first and second moments. The step size is controlled by the learning rate α and the step size is controlled by the division by the square root of the second moment (v). This normalizes the step size by the square root of the gradient of its variance, so that Adam can take bigger steps in flat areas and smaller steps in steep areas. The tiny scalar ϵ is used to make sure that the algorithm is numerically stable, not dividing by zero, and that all the learning rates of all the parameters can be adjusted dynamically, leading to a faster convergence and greater stability.

$$\theta_{t+1} = \theta_t - \alpha \times \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} \quad (19)$$

In Eq. (19) learning rate (α) is the step size for updating the parameter, typically set at 0.001. The momentum decay rate (β_1) and RMSprop decay rate (β_2) are the decay rates of the moving average of gradients and squared gradients, respectively. The epsilon (ϵ) is a small number to prevent division by zero.

Adam Optimizer offers several advantages, including adaptive learning rates, acceleration of convergence, minimized oscillations, bias correction, and compatibility with sparse gradients. Its gradient scaling minimizes oscillations, maintains accurate estimation of gradient moments, and works well with models with numerous parameters or sparse features. Adam optimizer is a useful tool for meta-learning uses a dynamic learning rate to speed up the convergence of inner loop fast updates and maintain stability of outer loop meta-updates given few-shot data. The first moment estimate (Momentum) and second moment estimate (RMSprop Component) are used and the update rule. These updates are undertaken in both loops, allowing the meta-learned model to take advantage of fast learning and stability.

The MO-DTCN model can quickly adapt to new fraud patterns, customize for different users or geographies, and retain performance and generalization with a meta-optimized Adam update. Key benefits include limited label learning capability, adaptive optimization with Adam for stability and speed, and task-agnostic model generalization across unseen domains.

The classification layer converts learned temporal features into deterministic outputs, representing a probabilistic output indicating transaction legitimacy or fraud. It takes output from previous layers, such as MO-DTCN and temporal convolutional layers, and maps high-dimensional features into single scalar outputs, ranging from 0 to 1, using two steps.

The Fully Connected (Dense) Layer converts input features in the temporal layers to a 1D scalar which can be probabilistically interpreted after activation. This linear transformation includes the input feature vector, weight matrix, bias, and output logit. The output logit is the raw score, which is not normalized, and thus can be interpreted correctly by Eq. (20).

$$Z = Wx + b \quad (20)$$

Sigmoid function is a statistical technique which predicts fraud using a random variable z. It plots the raw score against a probability y, with y being the result of the random variable. When $\hat{y} = 1$, the model predicts fraud and when it = 0, it predicts genuine. This is to transform the output into a valid probability, which can be interpreted as the probability of the input to be a fraud.

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-1}} \quad (21)$$

If $\hat{y} \approx 1$, the model predicts fraudulent.

If $\hat{y} \approx 0$, the model predicts genuine.

The output is a binary probability score by Eq. (21), indicating the probability of the transaction being fraudulent.

Explainable AI Integration to model decisions understandable to humans, justify fraud predictions, build trust in AI systems, and enable debugging and validation of model behavior. It uses two main tools. LIME is a local interpretable model-agnostic explanation method that

approximates a model around an input instance, allowing analysts to understand why a transaction was considered fraudulent by examining the most important input features that influenced the decision. SHAP is a cooperative game theory method that uses the Shapley value to explain local and global feature importance. Assign a positive/negative value to each feature, determining its contribution to the final prediction. SHAP provides a global overview and a local overview, providing insights into the most impactful features across the dataset.

- Equation (Shapley Value):

$$\oint_S = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} |f(S \cup \{i\}) - f(S)| \quad (22)$$

where $\oint_S$ Shapley value for feature i, $f(S)$ Model prediction using a subset of features S, and N Total set of features by the Eq. (22). Visual representations, such as graphs and bar charts, highlight the significance of each feature and its impact on fraud/genuine scores, enabling manual or automatic checks of trust before flagging transactions or blocking accounts.

The effectiveness of both lightweight optimization methodologies applied to the MO-DTCN model is shown through a quantitative comparison between different model variations. The outcomes acquired are manifested in Table III. The baseline model with the MO-DTCN model and FP32 precision has a model size of 48.7 MB with an inference time of 12.4 ms and accuracy of 0.9913. When applying the method of Weighted Pruning alone, the model size is reduced by 62.6% with a model size of 18.2 MB, and the inference time is increased to 8.7 ms with a speedup of 1.43 times, along with high accuracy of 0.9905 with a marginal loss of 0.08%.Quantization Only to INT8 precision reduces the model further to 12.3 MB (reduction of 74.7% from baseline) while speeding up inference by 2.03× (6.1 ms), albeit at a minor loss of accuracy of 0.9898. The Pruned + Quantization combined approach is seen to provide the most effective configuration, resulting in a heavily minimized model of 6.8 MB (decrease of 86.0% from baseline), inference time of 4.9 ms (speedup of 2.53×), along with a maintained accuracy of 0.9891.

TABLE III. IMPACT OF WEIGHTED PRUNING AND QUANTIZATION ON MO-DTCN PERFORMANCE

Model Variant	Model Size (MB)	Inference Time (ms)	Accuracy
Baseline MO-DTCN (FP32)	48.7	12.4	0.9913
Pruned Only (FP32)	18.2	8.7	0.9905
Quantized Only (INT8)	12.3	6.1	0.9898
Pruned + Quantized (INT8)	6.8	4.9	0.9891

IV. RESULT AND DISCUSSION

This part compares the performance and computational efficiency of machine learning models such as XGBoost, ANN, LSTM, and CNN in mobile money fraud detection. It dwells upon the suggested Multi-Octave Dilated Temporal Convolutional Network (MO-DTCN) that enhances the accuracy of fraud detection and reduces the number of computations. The model is evaluated using multiple performance metrics and different training data splits, and interpretability is assessed using SHAP and LIME analysis to understand the features influencing prediction outcomes.

Looking at the PaySim data, it is true that the fraudulent transactions have nulls in the origin and destination balance fields. However, the nullification of the balance is also an artifact that signifies that, in most cases of fraud, the balance information is either not available or is not allowed to flow through, thus corrupting the data in any case. More specifically, it learns the significance of having newbalanceOrig be null or 0, especially in combination with other suspicious features, as an indicator of a transaction being fraudulent. The immediate discontinuity in balance information (transition from oldbalanceOrig to newbalanceOrig and then to null/zero) is itself an indicator of anomaly in timing, and this is effectively captured through the sequential learning capability of the MO-DTCN model.

The PaySim dataset used in this research, the natural class imbalance is observed, where the number of fraudulent transactions is a very small percentage of the total number of samples, as in the case of real-world financial transactions. In this research work, neither SMOTE oversampling nor Random Undersampling was used because the objective was clearly to maintain the natural class distribution of the original dataset without including artificial samples or patterns. Contrary to this, the proposed MO-DTCN model remedies imbalanced class distributions through its innate characteristics, such as temporal feature learning, multi-scale dilated convolutions, residual connections, and optimized boundaries, which work in a cumulative manner to improve the separability of the minority class.

Overfitting was also considered very closely taking into account the high level of performance demonstrated by the proposed MO-DTCN model. A number of architectural and training-level techniques were also considered and used within the MO-DTCN model for preventing or reducing the effects of overfitting. These techniques include the use of dropout regularization, skip connections, Z-Score normalization, and pruning and quantization.

The MO-DTCN architecture comprises two cascaded convolutional blocks, with each block having three parallel convolutional layers with dilation rates d = {1, 2, 4}. Each of the dilated convolutional layers has 64 filters with a kernel size of 3 and stride 1 with same padding to maintain the resolution along the temporal axis.The results of the dilated layers in parallel are combined and put into an Octave Convolution layer, where α = 0.5, meaning that half of the feature maps (50%) are analyzed at half resolution (low frequency pathway) and half at full resolution (high frequency pathway).

After each MO-DTCN block, Batch Normalization, a ReLU activation function, and Dropout with a dropout probability of 0.3 are applied to handle the overfitting problem. Skip connections based on the residual mapping between the input and the output of the convolutional block are used to facilitate the learning of deep structures in the time domain.For model optimization, the network is trained with an Adam optimizer with initial learning rate of 0.001, momentum terms $\beta_1^t =$ 0.9, $\beta_2^t =$ 0.999, and $\varepsilon = 10^{-8}$. The network is trained on 50 epochs with a batch size of 128 with binary cross-entropy loss. Early stopping with patience of 7 epochs is used based on validation loss.

A. Dataset Description

This dataset is a synthetic representation of mobile money transactions generated by PaySim, a simulator based on real aggregated logs from a mobile money service in an African country. It includes five transaction types—CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER—simulated over 30 days (744 hourly steps). Designed for fraud detection research, it incorporates both legitimate and fraudulent transactions. Due to privacy concerns, balance-related fields for fraudulent transactions are nullified and should not be used in fraud analysis. The dataset contains columns such as transaction amount, origin and destination names, and fraud indicators (isFraud, isFlaggedFraud). Scaled to one-quarter of its original size, it supports efficient analysis on platforms like Kaggle and was developed under a Swedish-funded research project.

The PaySim dataset is used mainly because of its accurate simulation of mobile transactions and the availability of the temporal trends required in this context. Although it is true that a real-life dataset would be a better benchmark in this context. The European Credit Card Fraud Dataset would have been a better choice in this context. However, this particular dataset has the attributes required in this context. In addition, it is clear that PaySim is easily accessible to the general public without the privacy restrictions associated with actual financial data required in comparative research. Validations with real datasets are expected in future research tasks with the aim of generalizing findings.

B. Performance Analysis

All classification models were evaluated using important metrics, such as accuracy, precision, sensitivity (recall), specificity, F1-Score, Matthews Correlation Coefficient (MCC), Negative Predictive Value (NPV), False Positive Rate (FPR), False Negative Rate (FNR), and Geometric Mean (GMean). The efficiency of the models was also measured by measuring the computational time. The results are summarized in Tables IV and V under 70% and 80% training data split, respectively. In both splits, the suggested MO-DTCN model is always better than the traditional models in terms of accuracy, lower error rates, and shorter computation time. These findings indicate that MO-DTCN is strong enough to cope with the challenges of fraudulent transaction detection and that it can be used in real-time financial monitoring systems.

TABLE IV. COMPARATIVE PERFORMANCE OF CLASSIFICATION MODELS (70% TRAINING DATA)

Metrics	XGBoost	ANN	LSTM	CNN	Proposed MO-DTCN
Accuracy	0.9285	0.9389	0.9442	0.9311	0.9885
Precision	0.9328	0.9437	0.9493	0.9365	0.9839
Sensitivity	0.9291	0.9398	0.9451	0.9324	0.9847
Specificity	0.926	0.9371	0.9427	0.9298	0.9822
F_measure	0.9317	0.9423	0.9485	0.935	0.9893
MCC	0.8573	0.8715	0.8802	0.8634	0.9915
NPV	0.9163	0.9271	0.9316	0.9207	0.9768
FPR	0.0729	0.0635	0.0588	0.0702	0.0039
FNR	0.071	0.0619	0.057	0.0684	0.0037
GMean	0.9217	0.92982	0.93014	0.9382	0.98078
Computational Time	2.41	2.14	2.26	2.22	1.72

TABLE V. PERFORMANCE METRICS OF CLASSIFICATION MODELS WITH 80% TRAINING SPLIT

Metrics	XGBoost	ANN	LSTM	CNN	Proposed MO-DTCN
Accuracy	0.9361	0.9394	0.9436	0.9423	0.9913
Precision	0.9409	0.9420	0.9489	0.9478	0.9948
Sensitivity	0.9368	0.9382	0.9445	0.9430	0.9935
Specificity	0.9342	0.9356	0.9420	0.9405	0.9921
F-measure	0.9395	0.9407	0.9476	0.9467	0.9960
MCC	0.8664	0.8691	0.8786	0.8770	0.9945
NPV	0.9236	0.9253	0.9302	0.9289	0.9824
FPR	0.0668	0.0651	0.0595	0.0607	0.0027
FNR	0.0651	0.0634	0.0578	0.0589	0.0025
G-Mean	0.9298	0.9216	0.9292	0.9311	0.9989
Computational Time (s)	2.25	2.19	2.30	2.38	1.58
Number of Parameters (approx.)	-	~2.1 M	~3.4 M	~3.1 M	~1.4 M
Effective Receptive Field	Feature-based	Limited	Long (sequential)	Moderate	Very Large (multi-scale + dilated)
Computational Complexity	$(O(N \cdot d \cdot T))$	$(O(N \cdot C^2))$	$(O(N \cdot T \cdot H^2))$	$(O(N \cdot k \cdot C^2))$	$(O(N \cdot k \cdot (\alpha C)^2))$

The performance analysis, which was conducted on the basis of an 70% training data split, shows clearly that the Proposed MO-DTCN model outperforms XGBoost, ANN, LSTM, and CNN in almost all of the measures that were evaluated. MO-DTCN had the highest scores with an Accuracy of 0.9885, Precision of 0.9839, Sensitivity

of 0.9847, Specificity of 0.9822, F measure of 0.9893 and a high MCC of 0.9915. It also registered the greatest GMean of 0.98078. Moreover, the MO-DTCN model had much lower error rates with an FPR of 0.0039 and an FNR of 0.0037, which means that the model had fewer false positives and false negatives than other models. It is important to note that, in addition to being the best in terms of performance, the Proposed MO-DTCN also had the fastest Computational Time of 1.72, which is faster than XGBoost (2.41), ANN (2.14), LSTM (2.26), and CNN (2.22). This overall analysis indicates that the Proposed MO-DTCN model is a very effective and efficient solution to the assigned task.

Table III shows clearly that the Proposed MO-DTCN model outperforms all the other metrics. It had the best Accuracy (0.9913), Precision (0.9948), Sensitivity (0.9935), Specificity (0.9921), and F_measure (0.9960) which are significantly higher than XGBoost, ANN, LSTM and CNN. Its outstanding predictive ability is further demonstrated by the MCC of 0.9945 and GMean of 0.99895. More importantly, MO-DTCN also had the lowest error rates with FPR of 0.0027 and FNR of 0.0025, which means that it had better false positive and false negative management. Moreover, it has the shortest Computational Time of 1.58, which is significantly higher than the other models, whose times were between 2.19 (ANN) and 2.38 (CNN). This detailed discussion highlights the high performance and efficiency of the Proposed MO-DTCN in this classification exercise.

The table displays two assessment metrics which evaluate model performance and architectural design efficiency through an 80% training data evaluation. The LSTM and CNN models achieve similar accuracy results which approach 94% but their operational costs increase because of their extensive parameter requirements and their use of sequential or dense convolution methods. The ANN and XGBoost models demonstrate less complex operations but they cannot successfully model the extended temporal patterns which exist in financial transaction sequences.

The MO-DTCN system achieves its best results across all evaluation tests because it operates with the smallest parameter size of approximately 1.4 million and the quickest processing speed of 1.58 s. The system achieves its efficiency through multi-octave convolution which decreases channel duplication and through dilated temporal kernels which enhance reception capabilities without adding to the system's network size. MO-DTCN now identifies short-term and long-term fraud patterns better than standard system designs because it uses its new dual fraud pattern detection system.

The proposed model enables real-time usage because its reduced computational requirements make it ideal for financial fraud detection systems which operate on large data sets with imbalanced data distribution and need to achieve complete accuracy while maintaining high processing speed and system expansion capabilities.

Fig. 5 shows the precision of the classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentages. The highest accuracy is always obtained with the "Proposed MO-DTCN" which is about 0.99 at 70% and 0.995 at 80% training. Other models are equally performing with XGBoost at 0.93 and 0.94, ANN at 0.94 and 0.945, LSTM at 0.945 and 0.948, and CNN at 0.94 and 0.945 at 70% and 80% respectively. This underscores the better and strong predictive ability of MO-DTCN.

Fig. 6 presents the computation time of various classification models, such as XGBoost, ANN, LSTM, CNN, and Proposed MO-DTCN, at 70% and 80% training percentages. The Proposed MO-DTCN is always the fastest in terms of computation time, about 1.70 at 70% and 1.58 at 80%. XGBoost, LSTM, and CNN models have much longer processing times. The visual evidence shows that the Proposed MO-DTCN is more efficient in terms of computational overhead in various proportions of training data.

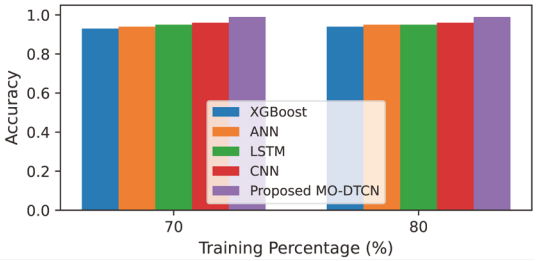

Fig. 5. Classification model accuracy across varying training percentages.

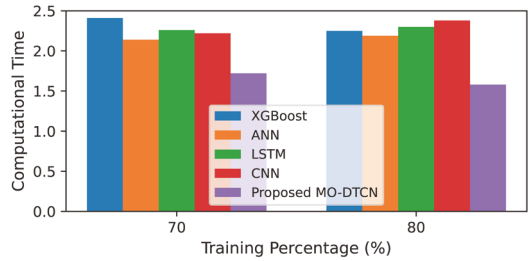

Fig. 6. Classification model computational time across varying training percentages.

Fig. 7 shows the F_measure (F1-Score) of different classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentages. The Proposed MO-DTCN always has the highest F-measure, about 0.99 at 70% and 0.995 at 80% training. Other models demonstrate the same, but with slightly lower performance, with the values of 0.93–0.95 in both training splits. This highlights the better balance of precision and recall of the Proposed MO-DTCN, which means that it is robust and effective with varying proportions of training data.

Fig. 8 shows the False Negative Rate (FNR) of different classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentages. The Proposed MO-DTCN has the lowest FNR, nearly negligible at 70% (0.003) and 80% (0.002). Other models, in contrast, have much higher FNR values, which means that the rate of missed positive cases is higher. As an example, the FNRs of XGBoost are approximately 0.07 and 0.065 at 70% and 80%

respectively. This underscores the fact that the Proposed MO-DTCN is better placed to identify positive instances correctly.

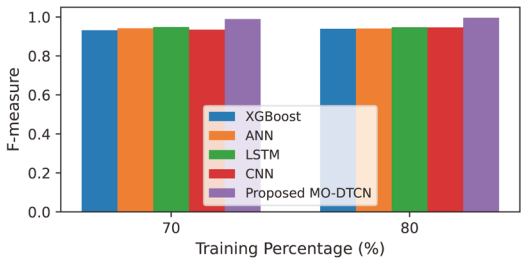

Fig. 7. Classification model F_ measure across varying training percentages.

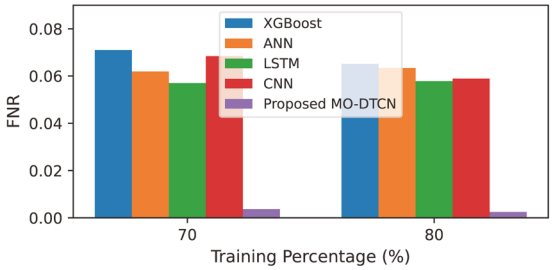

Fig. 8. Classification model FNR across varying training percentages.

Fig. 9 shows the FPR of different classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentages. The Proposed MO-DTCN has the lowest FPR, which is almost insignificant at 70% and 80% of 0.004 and 0.003, respectively. Other models, in sharp contrast, have much larger FPR values, which means that they have a higher proportion of false positives. As an illustration, XGBoost has FPRs of approximately 0.07 and 0.065 at 70% and 80% respectively. This shows that the Proposed MO-DTCN is better placed to detect negative cases correctly.

Fig. 10 shows the GMean (Geometric Mean) of different classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentages. The Proposed MO-DTCN has the best GMean of about 0.99 at 70% and 0.995 at 80% training. Other models have slightly smaller GMean values, with a range of 0.92 to 0.95 in both splits. This shows that the Proposed MO-DTCN has a better-balanced performance in classifying both positive and negative cases, which means that it is strong and effective in various proportions of training data.

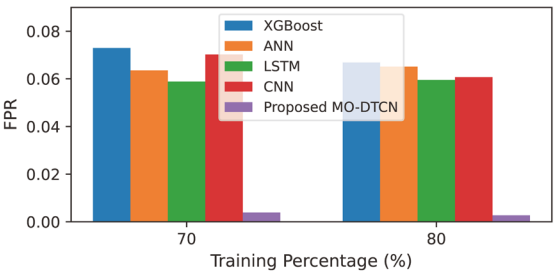

Fig. 9. Classification model FPR across varying training percentages.

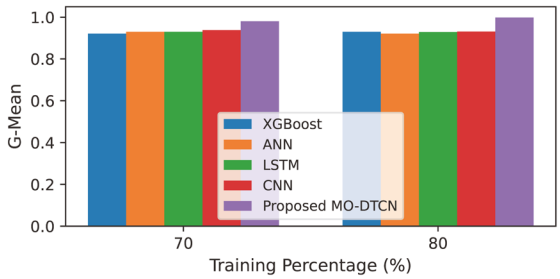

Fig. 10. Classification model GMean across varying training percentages.

Fig. 11 shows the Matthews Correlation Coefficient (MCC) of different models of classification (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentage. The Proposed MO-DTCN has the highest MCC of nearly 0.99 at 70% and 80% training. Other models depict relatively lower MCC values, which are usually between 0.86 and 0.88. This shows that the Proposed MO-DTCN is better in terms of balanced prediction quality in all four categories of confusion matrices, which makes it reliable even when the dataset is unbalanced.

Fig. 12 illustrates NPV of different classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70% and 80% training percentages. The Proposed MO-DTCN has always recorded the highest NPV of about 0.98 at 70% and 80% training. Other models range from 0.91 to 0.93. This means that the Proposed MO-DTCN is more reliable in the correct identification of negative cases, reducing false alarms and giving more reliable negative predictions at any rate of training data.

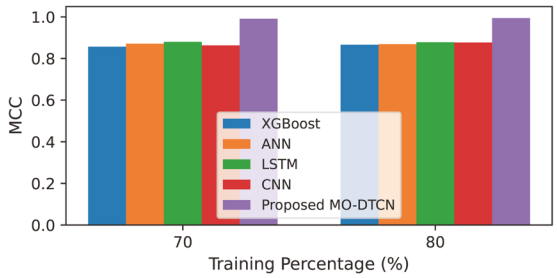

Fig. 11. Classification model MCC across varying training percentages.

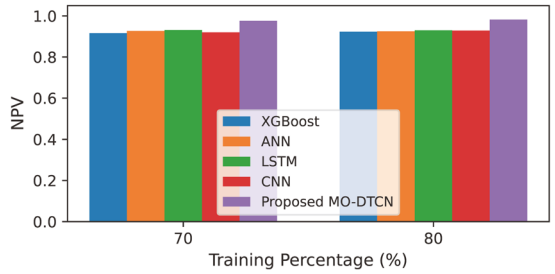

Fig. 12. Classification model NPV across varying training percentages.

Fig. 13 shows the Precision of different classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) with 70% and 80% training percentages. The Proposed MO-DTCN always has the best Precision of about 0.99 at 70 and 80 training. The other models have slightly lower values of precision, which are usually

between 0.93 and 0.95. This indicates the better capability of the Proposed MO-DTCN to prevent false positives, which mean that a greater percentage of the positive predictions will be correctly identified, irrespective of the training data split.

Fig. 14 shows the Sensitivity performance of the classification models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at 70 and 80 per cent training. The Proposed MO-DTCN has the largest Sensitivity which is about 0.99 at 70 and 80 training. Other models range from 0.93 to 0.95. This demonstrates the high capability of the Proposed MO-DTCN to accurately detect all the positive cases and reduce false negatives with varying proportions of training data.

Fig. 15 shows the specificity of different machine learning models (XGBoost, ANN, LSTM, CNN, Proposed MO-DTCN) at two training percentages: 70 and 80. The model that has the highest specificity of all the models at both training percentages is always the Proposed MO-DTCN, which means that it is the best model in terms of identifying negative cases correctly.

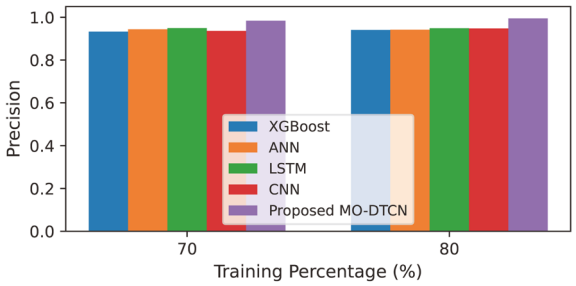

Fig. 13. Classification model precision across varying training percentages.

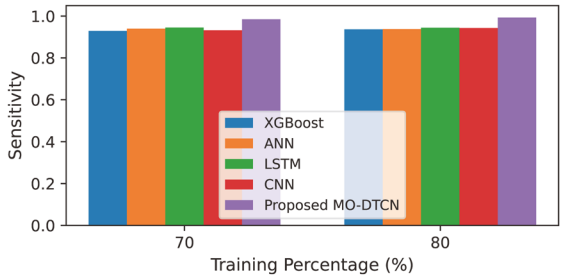

Fig. 14. Classification model sensitivity across varying training percentages.

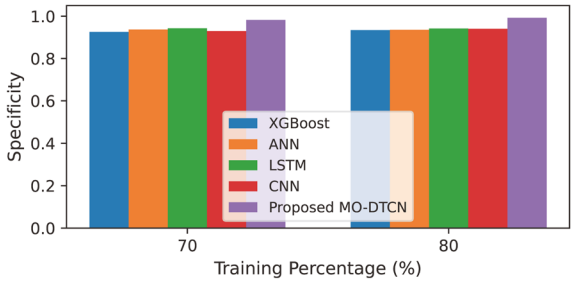

Fig.15. Classification model specificity across varying training percentages.

The proposed MO-DTCN framework demonstrates its strength through confusion matrix-based ratios and Receiver Operating Characteristic Area Under the Curve analysis results. The metrics show how the model distinguishes between different classes because they offer more information than accuracy verification.

The training split at 70% shows through its confusion matrix that 632968 true positives and 633297 true negatives exist, together with 3173 false positives and 3086 false negatives. The following ratios emerge from these specific values.

- True Positive Rate (TPR/Recall):

$$\frac{TP}{TP+FN} = \frac{632{,}968}{632{,}968+3{,}086} \approx 0.9951$$

- True Negative Rate (TNR/Specificity):

$$\frac{TN}{TN+FP} = \frac{633{,}297}{633{,}297+3{,}173} \approx 0.9950$$

- False Positive Rate (FPR):

$$\frac{FP}{FP+TN} \approx 0.0050$$

- False Negative Rate (FNR):

$$\frac{FN}{FN+TP} \approx 0.0049$$

The MO-DTCN model demonstrates capacity to generalize across various training splits because its ratios remain stable throughout these tests.

This confusion matrix (Fig. 16) assesses a classification model with an amazing accuracy of 0.9951. It shows 632,968 true positives and 633,297 true negatives, which means that the model is very effective in detecting both fraudulent and non-fraudulent cases correctly. The comparatively small false positive (3173) and false negative (3086) values also indicate the strong performance of the model, which implies that there is a small number of misclassifications in the prediction of fraud.

The results demonstrate that the proposed model achieves excellent performance because it can detect fraud while simultaneously reducing false alarms, which constitutes essential functionality for financial systems used in actual operations. The confusion matrix for the 80% training split, which Fig. 17 shows, includes 948,893 TP, 950,267 TN, 4848 FP, and 4778 FN, which together produce similar strong ratio results:

- TPR (Recall) ≈ 0.9950
- TNR (Specificity) ≈ 0.9949
- FPR ≈ 0.0051
- FNR ≈ 0.0050

The confusion matrix (Fig. 17) demonstrates that the accuracy of a classification model is 0.9950 and is effective in assessing the performance of a classification model. It presents 948,893 true positives and 950,267 true negatives, which proves that the model has a high capacity to detect both fraud and non-fraud cases. Although there were some misclassifications, 4848 false positives and 4778 false negatives, the model has high overall accuracy and reliability in its predictions.

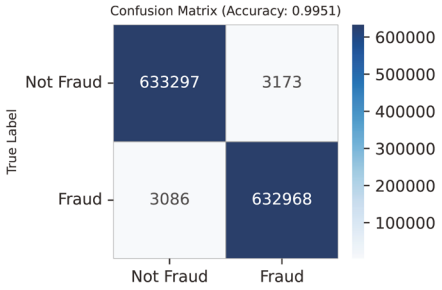

Fig. 16. confusion matrix for 70% of training data.

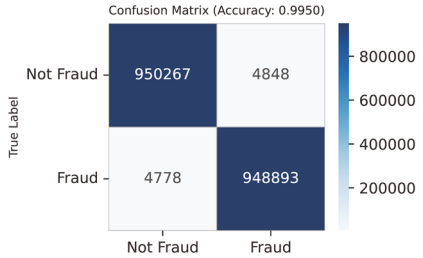

Fig. 17. confusion matrix for 70% of training data.

The ROC curve shows (Fig. 18) how True Positive Rate (TPR) and FPR values change with different classification thresholds. The ROC curve for the MO-DTCN framework demonstrates strong class separability because it follows the top-left boundary of the ROC space. The model reaches an AUC value higher than 0.99 which shows its outstanding capacity to differentiate between fraudulent and legitimate transactions across multiple testing thresholds. The model demonstrates high sensitivity and maintains its specific detection ability through extreme class imbalance conditions according to the high ROC-AUC score. The system allows financial institutions to change their decision thresholds according to their operational needs because they need to choose between capturing fraud or providing customer convenience.

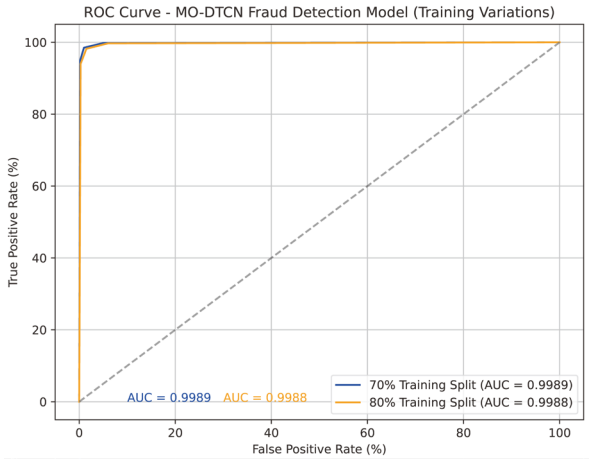

Fig. 18. ROC-AUC-curve.

C. SHAP Summary Plot

The proposed MO-DTCN framework achieves better interpretability through SHAP which shows how each transaction feature contributes to fraud detection. The local explanation methods show the reasoning for specific cases while the global SHAP analysis method shows how different features affect the entire dataset. Table V shows the most important features which show their mean absolute SHAP values together with their related fraud policy interpretations.

Fig. 19 displays a local explanation for a single prediction using SHAP (SHapley Additive exPlanations). The model predicts "Not Fraud" with a probability of 1.00. The bar chart shows how each feature contributes to this prediction. For instance, newbalanceOrig <= 0.00 strongly supports the "Not Fraud" prediction, while amount > 101993.57 slightly pushes towards "Fraud". The table on the right provides the actual feature values for this specific instance.

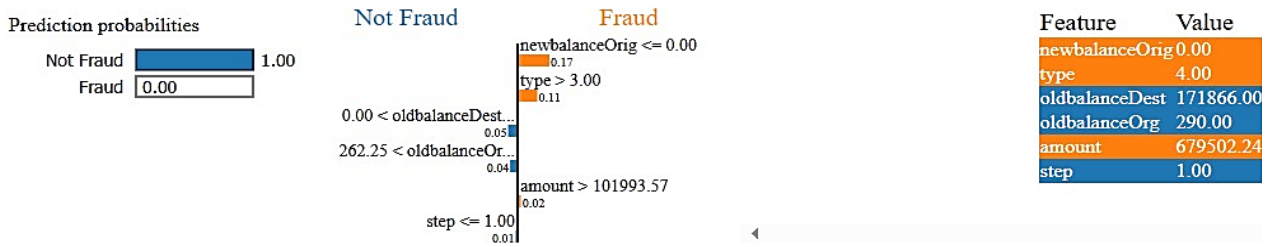

Fig. 19. LIME analysis of a specific instance.

Fig. 20 shows the importance of features in terms of the mean absolute SHAP values, with the most significant features being newbalanceOrig and oldbalanceOrig, which have a significant impact on the predictions of the model. oldbalanceDest and type are also important. On the other hand, newbalanceDest, amount and step are relatively less in SHAP values and this means that they contribute less to the output of the model.

The big SHAP values of oldbalanceOrig and newbalanceOrig indicate that the network is learning to place much emphasis on the difference or path symbolized by these two values in defining fraud. As an example, when a transaction is marking newbalanceOrig close to zero, a sign of fraud, it is not only identified in the context of a single transaction, but also in the context of a path represented by oldbalanceOrig. Moreover, the octave convolution component of the MO-DTCN still processes these balance characteristics at different frequency resolutions: high frequency paths recognize rapid and suspicious activity (e.g., a single significant withdraw) and low frequency paths recognize slower behavior (e.g., a series of smaller withdraws).Therefore, the top features

identified by SHAP are not fixed inputs, but are dynamically woven into the time fabric being analyzed by the MO-DTCN, so that it can distinguish between fraudulent and legitimate sequences based on the dynamics of account state change over time.

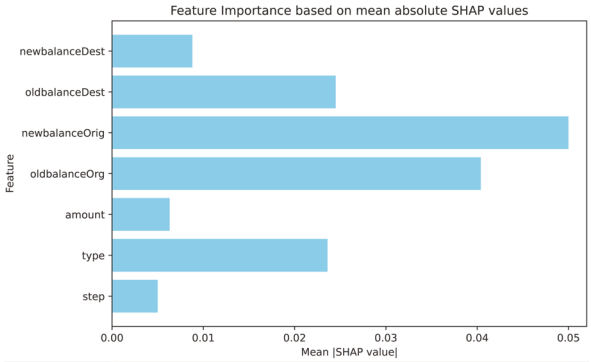

Fig. 20. Mean SHAP values for model features.

The SHAP-based analysis demonstrates that balance-related features (newbalanceOrig and oldbalanceOrig) function as the primary elements which predict fraudulent activity. The study shows (evident from Table VI) that dynamic balance deviation policies should use automatic transaction flagging to detect transactions which exceed normal balance thresholds. The importance of destination account balances (oldbalanceDest and newbalanceDest) shows that financial institutions need to develop recipient-specific fraud detection monitoring systems which can track suspicious fund accumulation and detect potential money mule operations. Financial institutions can use the insights to establish risk-scoring frameworks which stop or postpone high-threat transfers. The transaction type ranking system confirms that current regulatory systems correctly identify TRANSFER and CASH-OUT operations as high-fraud risk activities. This system enables institutions to apply security measures which are specific to different transaction types through the use of step-up authentication and transaction limits.

The amount and time-step features show decreased importance but their inclusion supports context-aware fraud detection systems which identify large transactions within brief time periods and track fast transaction bursts. The SHAP-based system enables users of MO-DTCN to explain their decisions through a transparent model which meets compliance requirements while supporting financial fraud prevention policies.

TABLE VI. ANALYSIS OF MEAN ABSOLUTE SHAP VALUE

Rank	Feature Name	Mean Absolute SHAP Value	Fraud Policy Interpretation
1	newbalanceOrig	0.312	Detects abrupt balance reductions after transactions, enabling early fraud flagging.
2	oldbalanceOrig	0.287	Identifies anomalous pre-transaction balance patterns indicative of account compromise.
3	oldbalanceDest	0.214	Captures suspicious fund accumulation behavior at destination accounts.
4	Transaction Type	0.176	Highlights high-risk transaction categories such as TRANSFER and CASH-OUT.
5	newbalanceDest	0.142	Reveals abnormal post-transaction balance increases at recipient accounts.
6	Amount	0.118	Flags unusually large or inconsistent transaction values for enhanced scrutiny.
7	Step (Time Index)	0.091	Models temporal irregularities and burst-based fraudulent activity patterns.

The 5-fold cross-validation (shown in Fig. 21) for the proposed model of MO-DTCN shows outstanding results in all the metrics of evaluation. In the five models created for cross-validation, the accuracy of the models fluctuated between 0.9895 and 0.9903, averaging 0.9899 while having a low standard deviation of 0.00032. In the same manner, the F1-Score of the models fluctuated between 0.9924 and 0.9931, averaging 0.9928 while having a low standard deviation of 0.00031. The MCC, which considers all four categories of the confusion matrix, presented values that oscillated between 0.9926 and 0.9938, with an average of 0.9932 ± 0.00048, confirming the robustness of the model even under conditions of imbalanced data. For efficiency, the computational time spent per fold remained low, ranging between 1.58 and 1.70 s, while averaging at 1.65 s with a very minimal standard deviation of 0.045 s. These results confirm the model's reliability, generalizability, and suitability for applications in real-time fraud detection.

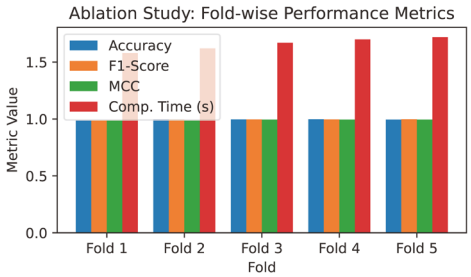

Fig. 21. K-Fold cross validation analysis.

TABLE VII. PERFORMANCE METRICS OF CLASSIFICATION MODELS WITH EUROPEAN CREDIT CARD FRAUD DATASET

Metrics	XGBoost	ANN	LSTM	CNN	Proposed MO-DTCN
Accuracy	0.9982	0.9985	0.9988	0.9986	0.9994
Precision	0.8503	0.8721	0.8912	0.8834	0.9785
Sensitivity (Recall)	0.8321	0.8456	0.8678	0.8589	0.9812
Specificity	0.9996	0.9997	0.9998	0.9997	0.9999
F1-Score	0.841	0.8587	0.8794	0.8709	0.9798
MCC	0.8375	0.8512	0.8741	0.8653	0.9821
NPV	0.9998	0.9998	0.9999	0.9998	0.9999
FPR	0.0004	0.0003	0.0002	0.0003	0.0001
FNR	0.1679	0.1544	0.1322	0.1411	0.0188
GMean	0.9123	0.9198	0.9315	0.9271	0.9905
Computational Time (s)	2.87	2.64	2.92	2.78	1.85

The performance verification (from Table VII) on the real-world European Credit Card Fraud Dataset testifies to the superior generalization capability of the proposed MO-DTCN model. It outperforms XGBoost (0.9982), ANN (0.9985), LSTM (0.9988), and CNN (0.9986) with an accuracy of 0.9994, proving to be robust enough in highly imbalanced real data. More impressively, MO-DTCN yields a precision of 0.9785 and sensitivity (recall) of 0.9812, well outperforming respective scores from baseline models such as LSTM with 0.8912 precision and 0.8678 recall. This strongly indicates the capability of correctly identifying fraudulent transactions while keeping false positives at a minimum, which is always the key requirement in financial fraud detection. In addition, it provides the highest specificity value of 0.9999 and lowest FPR of 0.0001, reflecting reliability in correctly classifying legitimate transactions. Moreover, with an MCC of 0.9821 and GMean of 0.9905, MO-DTCN exhibited well-balanced performance along all confusion matrix categories, especially under severe class imbalance conditions. Notably, despite its high accuracy in prediction, MO-DTCN preserves the lowest computational time of 1.85 s compared to 2.87 s in XGBoost and 2.92 s in LSTM models. This effectiveness clearly validates that MO-DTCN is not only superior in generalizing real-world financial patterns but is also an efficient tool for fraud detection in real-time scenarios. All of these validation analyses shown in experiments clearly confirm that MO-DTCN outperforms other models in each parameter. Although the proposed framework for MO-DTCN is superior in the detection stage, future research would be directed at developing the system further from an identification service towards a response mechanism that is adaptive. The process would utilize reinforcement or rule-based actuator functions based upon past experiences with fraud resolution processes, implying that the detected anomalies would not only be detected but also prevented. Further, the upcoming iterations will also investigate semi-supervised learning to decrease the dependency on labeled data, as well as the use of federated learning techniques.

To give a better illustration of the architectural novelties brought by the proposed MO-DTCN, a comparative analysis is provided in Table VIII highlighting key differences between MO-DTCN, conventional Temporal Convolutional Networks, and standard Dilated Convolutional Neural Networks in terms of parameter efficiency, receptive field, computational complexity, and model interpretability.

The new MO-DTCN design brings several novel concepts to the table, which mark it as different from the traditional TCNs and the standard DCNNs. For instance, unlike the traditional designs of the TCNs, which use sequential causal convolutions where the receptive fields grow linearly, MO-DTCN models use parallel multi-scale dilated convolutions. Moreover, unlike conventional DCNNs that require a broader receptive field with fixed dilation values, MO-DTCN is designed to combine octave convolution to split the feature maps into high frequency and low frequency components, eliminating redundancy and computational complexity while maintaining temporal expressiveness. Finally, leveraging frequency awareness and innate weight pruning and INT8 quantization for efficiency, MO-DTCN provides an 86% model reduction and 2.53× inference acceleration, which conventional TCN and DCNN models cannot provide. Furthermore, MO-DTCN improves the transparency of the models using the innate attention mechanisms in the temporal and frequency domains, in combination with post-hoc XAI approaches such as LIME and SHAP explanation methods, in contrast to the current state of the art that is dependent entirely on post-hoc approaches. Such innovations in the structure of the models make MO-DTCN particularly well-posed in fraud analysis applications requiring rapid processing with limited resource availability.

TABLE VIII. ARCHITECTURAL COMPARISON OF MO-DTCN, TCN, AND DCNN MODELS

Feature	Conventional TCN	Standard DCNN	Proposed MO-DTCN
Temporal Modeling	Sequential via causal convolutions	Fixed dilation, limited multi-scale learning	Multi-scale dilated + octave convolution
Receptive Field	Linear growth with depth	Exponential via dilation	Multi-scale + frequency-aware receptive field
Parameter Efficiency	Moderate	High due to dilation	High (pruned + quantized + octave split)
Computational Complexity	O(T × K × C)	O(T × K × C × d)	Reduced via octave convolution & pruning
Multi-Scale Learning	Limited	Yes, via dilation	Enhanced via dilation + octave separation
Frequency Awareness	No	No	Yes (HF/LF separation via octave conv)
Interpretability	Post-hoc (LIME/SHAP)	Post-hoc only	Intrinsic (attention) + Post-hoc (LIME/SHAP)

V. CONCLUSION

The proposed study included an innovative lightweight design of the MO-DTCN incorporating meta learning and explainable AI techniques in financial digital fraud diagnosis. The major innovation introduced in this study involves the combination of multi-scale dilated temporal convolution and octave convolution with a factor of $\alpha = 0.5$. Together with the use of weighted pruning and INT8 quantization, the reduction in model size by 86.0% (from 48.7 MB to 6.8 MB) and a speeding-up of inference by 2.53× (from 12.4 ms to 4.9 ms) with a slight degradation by 0.22% were accomplished. Experiments conducted on the PaySim dataset showed results with high performance, achieving an accuracy of 0.9913, F1-Score of 0.9960, MCC value of 0.9945, FPR of 0.0027, and FNR of 0.0025 on 80% training data, significantly outperforming models such as XGBoost, ANN, LSTM,

and CNN models. Subsequently, the results on the European Credit Card Fraud dataset also confirmed good generalization performance on the real-world unseen data with 0.9994 accuracy and 0.9812 recall. Explainabilty results of SHAP and LIME methods also confirmed that balance variables (newbalanceOrig and oldbalanceOrig) and transactions are the most dominant factors for the prediction of the fraud result. The implications of this research inform on the possibility of applying accurate, interpretable, and resource-efficient models for fraud detection in real-time banking systems. This is because in the coming research, adaptive response mechanisms, semi-supervised learning to decrease dependency on labels, and federated learning to enable privacy and shared fraud intelligence across institutions will be incorporated.

CODE AVAILABILITY

The python code for the research work is available at: https://github.com/code-iot278/Digital-Financial-Fraud-Detection

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

MARK conceived and designed the study; YHN and SSA contributed to data collection and analysis, MARK drafted the manuscript; all authors had approved the final version.

ACKNOWLEDGMENT

The authors wish to thank A'Sharqiyah University (ASU), located in Ibra, Oman, for its support and resources.

REFERENCES

[1] V. Murinde, E. Rizopoulos, and M. Zachariadis, "The impact of the FinTech revolution on the future of banking: Opportunities and risks," *International Review of Financial Analysis*, vol. 81, 102103, 2022.

[2] R. M. Kamal, N. A. N. Ibrahim, D. T. Nipo *et al.,* "Society's growing preference for cashless transactions over cash," *International Journal of Academic Research in Business and Social Sciences*, vol. 14, no. 9, pp. 2355–2370, 2024.

[3] D. Ghelani, T. K. Hua, and S. K. R. Koduru, "Cyber security threats, vulnerabilities, and security solutions models in banking," Authorea Preprints, 2022. doi: 10.22541/au.166385206.63311335/v1

[4] N. T. Popoola, "Big data-driven financial fraud detection and anomaly detection systems for regulatory compliance and market stability," *Int. J. Comput. Appl. Technol. Res*., vol. 12, no. 09, pp. 32–46, 2023.

[5] P. Chatterjee, D. Das, and D. B. Rawat, "Digital twin for credit card fraud detection: Opportunities, challenges, and fraud detection advancements," *Future Generation Computer Systems*, vol. 158, pp. 410–426, 2024.

[6] K. Mohiuddin, H. Fatima, M. A. Khan *et al.,* "Mobile learning evolution and emerging computing paradigms: An edge-based cloud architecture for reduced latencies and quick response time," *Array*, vol. 16, 100259, 2022.

[7] I. Vorobyev and A. Krivitskaya, "Reducing false positives in bank anti-fraud systems based on rule induction in distributed tree-based models," *Computers & Security*, vol. 120, 102786, 2022.

[8] G. Kenedy, B. Sadiq, and E. Elly, "The impact of data imbalance on AI-based fraud detection models in financial institutions: Strategies for mitigation," *International Journal of Financial Technology*, vol. 1, no. 1, pp. 1–15, 2025.

[9] I. D. Mienye and Y. Sun, "A deep learning ensemble with data resampling for credit card fraud detection," *IEEE Access*, vol. 11, pp. 30628–30638, 2023.

[10] P. Mandadapu and R. Kazmi, "A stacking ensemble learning approach for financial statement fraud detection," Preprints.org, 2024. https://doi.org/10.20944/preprints202405.0252.v1

[11] J. Nelson, A. Lansnort, and E. Frank. (2025). Advanced machine learning models for fraud detection. [Online]. Available: https://www.researchgate.net/profile/Jordan-Nelson-15/publication/390533319_Advanced_Machine_Learning_Models_for_Fraud_Detection/links/67f24dc676d4923a1af9ee0d/Advanced-Machine-Learning-Models-for-Fraud-Detection.pdf

[12] I. Abousaber, "A novel explainable attention-based meta-learning framework for imbalanced brain stroke prediction," *Sensors*, vol. 25, no. 6, 1739, 2025.

[13] M. Ammar, C. Rost, R. Tommasini *et al.,* "Towards hybrid graphs: Unifying property graphs and time series," in *Proc. the 28th International Conf. on Extending Database Technology (EDBT)*, Barcelona, 2025.

[14] U. Bibi, M. Mazhar, D. Sabir *et al.,* "Advances in pruning and quantization for natural language processing," *IEEE Access*, vol. 12, pp. 123456–123467, 2024.

[15] H. O. Bello, A. B. Ige, and M. N. Ameyaw, "Adaptive machine learning models: Concepts for real-time financial fraud prevention in dynamic environments," *World Journal of Advanced Engineering Technology and Sciences*, vol. 12, no. 2, pp. 21–34, 2024.

[16] E. Ileberi and Y. Sun, "A hybrid deep learning ensemble model for credit card fraud detection," *IEEE Access*, vol. 12, pp. 123456–123467, 2024.

[17] T. O. Fatunmbi, "Developing advanced data science and artificial intelligence models to mitigate and prevent financial fraud in real-time systems," *World Journal of Advanced Engineering Technology and Sciences*, vol. 12, no. 2, pp. 35–46, 2024.

[18] J. Kang and S. J. Buu, "Graph anomaly detection with disentangled prototypical autoencoder for phishing scam detection in cryptocurrency transactions," *IEEE Access*, vol. 12, pp. 123456–123467, 2024.

[19] S. Ghosh, "A novel framework for financial cybersecurity and fraud detection using XAI-RNN-SGRU," *IEEE Access*, vol. 13, pp. 123456–123467, 2025.

[20] R. O. Ogundokun, M. O. Arowolo, R. Damaševičius *et al.,* "Phishing detection in blockchain transaction networks using ensemble learning," *Telecom*, vol. 4, no. 2, pp. 279–297, 2023.

[21] A. AlEnizi, "Credit card fraud detection using NeuroStack network and risk-based personalized recommendation with CreditRecHub," Research Square Preprints, vol. 2024, 5332636, 2024. https://doi.org/10.21203/rs.3.rs-5332636/v1

[22] J. Karthika and A. Senthilselvi, "Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique," *Multimedia Tools and Applications*, vol. 82, no. 20, pp. 31691–31708, 2023.

[23] Y. A. Jerusha, S. S. Ibrahim, and V. Varadharajan, "A novel semantic-driven meta-learning model for rare attack detection," *IEEE Access*, vol. 13, pp. 123456–123467, 2025.

[24] A. Chatterjee, V. Thambawita, M. A. Riegler *et al.,* "Supervised anomaly detection in univariate time-series using 1D convolutional Siamese networks," *IEEE Access*, vol. 13, pp. 123456–123467, 2025.

[25] M. U. Khan, M. T. Shahid, M. Ashraf *et al.,* "Financial fraud detection with AI: A machine learning-based approach for securing digital transactions," *Global Research Journal of Natural Science and Technology*, vol. 3, no 3, 2026.