*Digital Object Identifier*

# Transparent Credit Card Fraud Detection: Integrating Deep Learning with Kolmogorov-Arnold Networks

**SAKSHAM SHARMA[1], SHAURYA DEV PATHAK[1], SUSHANT GARGI[1], SUGANESHWARI G[1]**
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
Corresponding author: Suganeshwari G (e-mail: suganeshwari.g@vit.ac.in).

**ABSTRACT** Credit card fraud is a serious problem that requires sophisticated detection. Although deep learning can perform exceptionally well in fraud detection, the black-box nature of models is an issue for trust and effective human oversight. Conversely, although interpretable, traditional machine learning models may be unable to detect subtle fraud patterns in intricate transactional data. To overcome this, we introduce a new fraud detection framework based on the Linear Time Attention CNN (LTACNN). This black-box model is designed to capture anomalies and behavioural changes typical of fraudulent transactions over the short and long term. The LTACNN uses convolutional-based operations to identify local, immediate patterns within streams of transactions, effectively capturing sudden bursts of fraudulent transactions. Meanwhile, long short-term memory cells operate on the overall transactional history, detecting subtle, evolving fraud patterns. An attention mechanism then selectively integrates these short-term and long-term perspectives to generate a strong and informed fraud judgment. Our framework also incorporates a Kolmogorov-Arnold Network (KAN) to make interpretability open. The KAN makes the LTACNN's decision-making transparent, generating precise mathematical functions and feature importance scores. This explainable output provides human analysts with trust, enables informed intervention, and ultimately enhances the efficacy of fraud management. The code for our framework is available at: GitHub Repository

**INDEX TERMS** Attention Mechanism, Class Imbalance, Convolutional Neural Networks (CNN), Credit Card Fraud Detection, Kolmogorov-Arnold Networks (KANs), Long Short-Term Memory (LSTM), XAI.

## I. INTRODUCTION

CREDIT card fraud remains a widespread and insidious menace in the global financial environment, demanding increasingly advanced and adaptive detection methods. The sheer volume of daily transactions, measured in billions worldwide, and the ever-changing tactics of fraudsters make it a challenging and dynamic environment for financial institutions [13]. Although deep learning algorithms are exemplary for identifying sophisticated patterns typical of fraud, their very "black-box" nature is a serious deterrent to their broad use in high-risk financial settings [2]. In contrast, conventional machine learning models, typically preferred for their interpretability, may not fully capture the richness of modern frauds in large, high-dimensional transactional data [17]. This very real trade-off between high accuracy and human understanding requirements motivates the core research of this paper.

The demand from customers and regulators for presenting good reasons justifying why one specific transaction should be suspected to be fraudulent continues to increase. This is not just a satisfying curiosity; it is necessary to gain trust, make timely and efficient intervention possible, support compliance with fair lending legislation, and allow human analysts to make sound decision-making choices. Regulations such as the General Data Protection Regulation (GDPR) and other data protection laws strongly emphasise transparency and interpretability in automated decision-making systems. "Black box" systems, even with high predictive capability, tend not to possess such a needed interpretability level and thus become unsuitable for use in practical settings of real-life fraud management.

To meet this vital requirement, we present a new fraud detection system that synergistically integrates the strength of a highly effective but secure deep-learning model—the Linear Time Attention CNN (LTACNN)—and the intrinsic transparency of a Kolmogorov-Arnold Network (KAN). Our system can deliver accuracy for fraudulent transactions and transparent, human-interpretable explanations for such predictions. The foundation is our deep learning model, the LTACNN.

The LTACNN is tailored. To handle class imbalance, oversampling is performed using ADASYN, which has been described in [3]. Data for the model is specially prepared. The data is suitable for time-series analysis as it is a series of feature vectors. Feature scaling is done using RobustScaler. This precautionary step avoids features with higher magnitudes taking over the model.

The LTACNN exploits the strengths of Long Short-Term Memory networks (LSTMs) and Convolutional Feature extractors. The 1D convolutional layers are best suited for detecting local, short-term patterns in sequences of transactions [6]. The layers are feature detectors, identifying bursts of abnormal activity or anomalous transaction patterns indicative of fraud. Multiple convolutional layers are stacked upon one another to create a hierarchical representation, identifying increasingly complex patterns. The LSTM layers process the more extended transaction history, learning long-term patterns and identifying subtle, evolving fraud patterns that can emerge over long periods [15]. This sequential processing capability is critical for identifying sophisticated schemes involving gradual behaviour changes or innocuous-looking transactions that, when aggregated, reveal fraudulent intent. To further improve the model in highlighting the most significant information within the short-term and long-term settings, we introduce a LinearAttention layer developed inhouse. It applies scaled linear attention, inspired by developments in Transformer architecture but tuned to computational efficiency. Unlike quadratic-time attention, our developed LinearAttention layer is of complexity O(T) and may be scaled up to the sequence length of a transaction. Such is especially pertinent in real-life scenarios wherein transaction histories would be extended. Attention allows the calculation of the relative contribution of different parts of the transaction sequence toward making a dynamic prediction and for the model to highlight the most relevant features and interrelations.

To unlock interpretability and bridge the gap between the LTACNN's robust pattern recognition and the need for human interpretation, our system incorporates a Kolmogorov-Arnold Network (KAN). With the pykan library, the KAN is an "interpretability layer" explaining the LTACNN's decision-making. Specifically, the KAN is trained to approximate the LTACNN's decision function with an explicit, mathematical explanation of how the model makes its predictions. This is achieved by parameterising functions of the KAN as B-splines, allowing flexible and smooth function approximation consistent with the principles outlined in [27]. The KAN can "translate" LTACNN's complex, "non-linear" interactions into easy-to-understand mathematical functions. The output of the KAN, along with a prediction (legitimate or fraudulent), also provides feature importance scores indicating how much each input feature contributed to the final decision. The traditional "black-box" deep learning method is quite different from this.

This study has primarily three contributions:

- We propose a novel fraud detection framework by blending a powerful but black-box deep learning model (LTACNN) and an explanation-friendly model (KAN). This addresses the largest gap for most fraud detection systems of today. The KAN is initially optimised for the provided width, grid, and k parameters.
- We demonstrate, through the KAN structure and feature importance scores, how to explain and make transparent the LTACNN's decisions to human analysts. This is important in shedding light on factors influencing fraud predictions.
- The combination of CNN, LSTM, linear attention, and KAN modules in the same PyTorch framework allows our model to deliver high accuracy with interpretability. Direct usage of RobustScaler, ADASYN, and LinearAttention layer focuses on real-world, efficient performance and scalability.

The forthcoming sections of this work delve into the precise technical details of each component (LTACNN, KAN, and training process), present empirical findings demonstrating the framework's effectiveness, and a framework of the ramifications of this research for the broader community of interpretable AI in financial use. The training process utilises an appropriate loss function called binary cross-entropy loss, which is required for classification.

## II. RELATED WORKS

### A. TRADITIONAL AND DEEP LEARNING APPROACHES

The high growth rate of electronic payment systems and the consequent increase in the use of credit cards have made fraud detection a significant field of research. Early research mainly utilized machine learning techniques like Support Vector Machines (SVMs), Decision Trees, and Logistic Regression to distinguish between good and bad transactions [1]. While these conventional models are reasonably interpretable and computationally inexpensive, they are typically incapable of learning complex nonlinear relationships within financial data. Moreover, the extreme class imbalance in fraud detection—the fraudulent transactions are only a small subset of all transactions—magnifies the effect of even minor misclassification errors [2]. This challenge has prompted researchers to seek more robust approaches to identify complex patterns from large-scale, imbalanced data.

More recently, deep learning methods have been prominent in credit card fraud detection. Deep neural networks like Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) have been used to learn feature hierarchies directly from raw transactional data [3]. LSTM networks, for instance, have been used to tap into temporal relationships in sequential transaction data, and CNNs particularly excel at learning spatial and contextual features in transactional metadata [4]. While these deep models have exceptional detection performance, they have a "black-box" problem—limiting their interpretability and undermining trust and regulatory acceptance in high-risk financial environments [5].

### B. HYBRID ARCHITECTURES AND INTERPRETABILITY TECHNIQUES

To overcome such adversity, recent research has investigated hybrid deep learning architectures that leverage the strength of multiple methods. Various authors have designed hybrid LSTM-CNN architectures for fraud detection to detect temporal and spatial patterns in the transaction data concurrently [6], [7]. These models are more sophisticated than single-architecture models because they have sequential and local feature extraction. One of the shortcomings of most of the above studies is the absence of inherent interpretability. Regulators and finance practitioners require transparency in decision-making; therefore, post-hoc explanation techniques have been proposed. These are often used as add-ons and not as part of the core functionality [8].

Attention mechanisms have become a potent weapon in deep learning to tackle interpretability. Dynamically learning weights to various input features depending on their importance, attention layers enable the model to concentrate on the most critical regions of the input data. In fraud detection, attention layers have been employed to improve the extraction of essential patterns by removing noise and less informative features [9]. Nevertheless, most existing hybrid models apply attention superficially, neglecting to embed it deeply into the network architecture. This drawback has led to more research towards embedding attention modules deeply into the LSTM and CNN pipelines to enhance interpretability and performance simultaneously [10].

Another possible approach to model interpretability is through the use of approximation models. In particular, Kolmogorov-Arnold Networks (KAN) have been investigated to approximate the behaviour of complex deep learning models. KAN can approximate high-order nonlinear functions and be employed as an interpretability layer by providing a mathematical description of the decision boundaries learned by a model [11]. Although KAN has been applied in other domains, its use in fraud detection is new. Integrating the KAN module with a hybrid LSTM-CNN model may thus not only yield improved performance but also improved transparency in decision-making [12].

### C. DATA RESAMPLING AND ENSEMBLE METHODS

In addition to architectural advancements, class imbalance has also gained considerable attention. Traditional oversampling methods like SMOTE and ADASYN have been widely used to create synthetic samples of the minority class. Still, they add noise and cannot replicate the original data distribution [13]. To overcome these limitations, new resampling methods like SMOTE-ENN and K-means SMOTEENN have been implemented. K-means and SMOTEENN initially divide the data into clusters with the help of K-means clustering and maintain the minority class distribution locally, then use SMOTE to create synthetic samples and ENN to remove noisy majority instances [14], [15]. Empirical experiments have illustrated that these hybrid resampling strategies yield better performance metrics like improved F1-scores and ROC-AUC values compared to classical oversampling strategies [16].

Besides this, ensemble learning methods have also been an effective solution to fraud detection. Bagging, boosting, and stacking are methods employed to combine the output of many classifiers and make a prediction that takes advantage of the strengths of the different models [17]. Of these, stacking ensembles have been of the most significant potential since they provide the ability to combine heterogeneous models (e.g., SVM, Random Forest, XGBoost, and LightGBM) into a unified combined predictor. Stacking ensembles are proven to improve prediction accuracy and model complexity, which, in return, can constrain explainability—a consideration of most tremendous significance in finance applications [18]. To reduce this, our work incorporates explainable AI methods like Local Interpretable Model-Agnostic Explanations (LIME) into the stacking method to generate localized, interpretable explanations of predictions [19]. Past research has established that hybrid ensemble methods are highly effective in handling imbalanced data while maximizing detection accuracy [20], [21].

Our paper bridges this gap by introducing a deep hybrid model that exploits LSTM's temporal dependency capture and CNN's spatial feature extraction. Our model employs an attention mechanism—realized custom LinearAttention layer—to dynamically highlight only the most salient features. In addition, we present an approximation module built around a Kolmogorov–Arnold Network (KAN), which serves as an interpretability layer by providing transparent, mathematical explanations for the model's decision-making. Class imbalance is addressed via ADASYN oversampling. However, our system is highly flexible and can easily employ enhanced resampling techniques, e.g., K-means SMOTEENN [24], as and when the situation demands.

Additionally, our system's ensemble effect is achieved through a stacking function that aggregates predictions of multiple models to improve overall accuracy without compromising explainability. The explainable AI technique LIME is applied to local predictions to render the final output transparent and trustworthy. Our end-to-end system achieves state-of-the-art detection accuracy and meets the imperative demands of interpretability and conformity for real-time financial fraud detection systems.

### D. CONVOLUTIONAL NEURAL NETWORKS (CNNS)

Convolutional Neural Networks (CNNs) extract spatial and contextual features from structured transaction data. By treating transaction histories as matrices (or multi-channel signals), CNNs learn hierarchical representations that distinguish between fraudulent and genuine transactions.

### E. LINEAR ATTENTION MECHANISM

Attention mechanisms dynamically weight input features based on their importance. Our custom LinearAttention layer approximates traditional self-attention by using a feature map,

$$\phi(x) = \mathrm{ELU}(x) + 1, \tag{1}$$

which allows us to rewrite the attention mechanism as follows:

$$\mathrm{Attention}(Q, K, V) = \frac{\phi(Q_i)\left(\sum_{j=1}^{N} \phi(K_j)^T V_j\right)}{\phi(Q_i)\sum_{j=1}^{N} \phi(K_j)}, \tag{2}$$

reducing computational complexity from quadratic to linear in sequence length.

### F. KOLMOGOROV-ARNOLD NETWORKS (KANS)

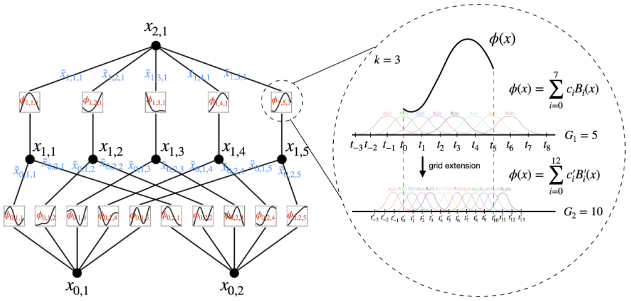
$x_{2,1}$
$\tilde{x}_{1,1,1}$ $\tilde{x}_{1,2,1}$ $\tilde{x}_{1,3,1}$ $\tilde{x}_{1,4,1}$ $\tilde{x}_{1,5,1}$
$\phi_{1,1,1}$ $\phi_{1,2,1}$ $\phi_{1,3,1}$ $\phi_{1,4,1}$ $\phi_{1,5,1}$
$x_{1,1}$ $x_{1,2}$ $x_{1,3}$ $x_{1,4}$ $x_{1,5}$
$\tilde{x}_{0,1,1}$ $\tilde{x}_{0,2,1}$ $\tilde{x}_{0,1,2}$ $\tilde{x}_{0,2,2}$ $\tilde{x}_{0,1,3}$ $\tilde{x}_{0,2,3}$ $\tilde{x}_{0,1,4}$ $\tilde{x}_{0,2,4}$ $\tilde{x}_{0,1,5}$ $\tilde{x}_{0,2,5}$
$\phi_{0,1,1}$ $\phi_{0,1,2}$ $\phi_{0,1,3}$ $\phi_{0,1,4}$ $\phi_{0,1,5}$ $\phi_{0,2,1}$ $\phi_{0,2,2}$ $\phi_{0,2,3}$ $\phi_{0,2,4}$ $\phi_{0,2,5}$
$x_{0,1}$ $x_{0,2}$
$k = 3$
$\phi(x)$
$\phi(x) = \sum_{i=0}^{7} c_i B_i(x)$
$G_1 = 5$
$t_{-3}$ $t_{-2}$ $t_{-1}$ $t_0$ $t_1$ $t_2$ $t_3$ $t_4$ $t_5$ $t_6$ $t_7$ $t_8$
grid extension
$\phi(x) = \sum_{i=0}^{12} c_i' B_i'(x)$
$G_2 = 10$
$t'_{-3}$ $t'_{-2}$ $t'_{-1}$ $t'_0$ $t'_1$ $t'_2$ $t'_3$ $t'_4$ $t'_5$ $t'_6$ $t'_7$ $t'_8$ $t'_9$ $t'_{10}$ $t'_{11}$ $t'_{12}$ $t'_{13}$

FIGURE 1: The image outlines a Kolmogorov-Arnold Network (KAN) with B-spline basis functions and learnable coefficients, demonstrating grid extension and hierarchical layers L1, L2, L3."Source": [34]

Kolmogorov-Arnold Networks (KANs) provide interpretability by approximating high-dimensional functions as a composition of univariate functions, typically parameterised. The KAN module generates transparent, mathematical explanations for the decision-making process in our framework. Figure 1 illustrates a KAN represented by learnable coefficients for B-spline basis functions.

### G. ADAPTIVE SYNTHETIC SAMPLING (ADASYN)

ADASYN is a data-level approach that alleviates class imbalance by adaptively generating synthetic samples in regions where the minority class is underrepresented. This improves the classifier's ability to learn the decision boundary between fraudulent and genuine transactions without over-replicating noisy instances.

## III. MODEL ARCHITECTURE: LINEAR TIME ATTENTION CNN FEATURE EXTRACTOR

The section below describes the architecture and theoretical rationale underlying the linear-time attention CNN feature extractor, which is used for preprocessing transactions before classification. The LTACNN effectively extracts both local and global temporal patterns computationally by integrating the strengths of convolutional neural networks and the attention mechanism within linear time.

### A. ARCHITECTURE

The LTACNN model includes the following main components:

**Input Representation:** Transaction information is represented as a sequence of feature vectors. Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the feature vector for the transaction at time step $t$, where $d$ is the feature dimension (e.g., amount, merchant category, location). The input sequence is $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$, with $T$ being the sequence length.

**1D Convolutional Layers:** There are multiple 1D convolutional layers on the input sequence. The layers utilise local temporal features. The $k$-th convolutional filter in layer $l$ is of size $K_l$ and produces a feature map $\mathbf{h}_k^{(l)}$. The convolution operation is as follows:

$$\mathbf{h}_{k,t}^{(l)} = \mathrm{ReLU}\left(\sum_{j=0}^{K_l-1} \mathbf{w}_{k,j}^{(l)} \cdot \mathbf{x}_{t+j} + b_k^{(l)}\right), \tag{3}$$

where $\mathbf{w}_{k,j}^{(l)} \in \mathbb{R}^d$ is the weight vector for the $j$-th component of the $k$-th filter in layer $l$, $b_k^{(l)}$ is the bias term, and $\mathrm{ReLU}(x) = \max(0, x)$ is the rectified linear unit activation function. Multiple layers are stacked to capture increasingly complex patterns and construct a hierarchical representation.

**Linear Time Attention Mechanism:** We employ a linear time attention mechanism to capture global dependencies in the sequence after the convolutional layers. Unlike standard quadratic-time attention (e.g., in Transformers), linear attention has $O(T)$, computational complexity and is thus well-suited for long sequences of transactions. We employ a simplified linear attention form, as discussed below.

Let $\mathbf{H}^{(L)} = [\mathbf{h}_1^{(L)}, \mathbf{h}_2^{(L)}, \ldots, \mathbf{h}_{T'}^{(L)}]$ be the last convolutional layer (potentially with sequence length $T'$ reduced due to pooling). The linear attention mechanism calculates a weighted sum of the hidden states:

$$\mathbf{c}_t = \sum_{i=1}^{T'} \alpha_{t,i} \mathbf{h}_i^{(L)}, \tag{4}$$

where weights $\alpha_{t,i}$ are calculated from a kernel function $\kappa(\cdot, \cdot)$ that calculates similarity between a query vector $\mathbf{q}_t$ and a key vector $\mathbf{k}_i$:

$$\alpha_{t,i} = \frac{\kappa(\mathbf{q}_t, \mathbf{k}_i)}{\sum_{j=1}^{T'} \kappa(\mathbf{q}_t, \mathbf{k}_j)}. \tag{5}$$

Key and query vectors generally originate from the latent states employing linear transformations:

$$\mathbf{q}_t = \mathbf{W}_q \mathbf{h}_{t-1}^{(L)}, \tag{6}$$

$$\mathbf{k}_i = \mathbf{W}_k \mathbf{h}_i^{(L)}, \tag{7}$$

where $\mathbf{W}_q$ and $\mathbf{W}_k$ are the learnable weight matrices. The most common kernel function is the exponential of the dot product:

$$\kappa(\mathbf{q}_t, \mathbf{k}_i) = \exp(\mathbf{q}_t^T \mathbf{k}_i). \tag{8}$$

Other kernel functions, such as cosine similarity-based ones, can also be used. The context vector $\mathbf{c}_t$ is a global representation of the sequence, weighted by what each time step contributes to the current time step $t$.

**Output Layer:** The context vectors $[\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T]$ (or an aggregated/pooled version thereof, e.g., via max-pooling or averaging) are fed into a terminal output layer (e.g., a fully connected layer or a recurrent layer) to produce the feature representation $\mathbf{z} \in \mathbb{R}^{d'}$, where $d'$ is the dimensionality of the extracted features. The end representation encodes both local and global temporal patterns.

### B. THEORETICAL JUSTIFICATIONS

The following theoretical considerations drive the LTACNN design decisions:

**Local Pattern Extraction:** 1D convolutional layers are particularly suited for local temporal pattern detection in sequences of transactions. This is due to the intuition that fraudulent behaviour is most likely to manifest as abnormal transaction patterns in a short time frame (e.g., a series of small followed by a significant transaction or transactions in unexpected locations). The convolution operation can identify these local patterns irrespective of their exact location in the sequence.

**Global Dependency Modeling:** The linear time attention mechanism allows the model to identify long-range dependencies in the transaction sequence. This is necessary to identify fraudulent patterns that last longer, e.g., a cumulative build-up of suspicious behaviour or repeated patterns with abnormal timing. The attention mechanism allows the model to assign the relative importance of different sequence elements to make a prediction.

**Feature Hierarchy:** Combining convolutional layers and attention results in a hierarchical feature representation. Convolutional layers learn the low-level features (e.g., transaction types, range of amounts) and combine them into a higher-level, context-sensitive representation considering the entire sequence.

**Efficiency:** Linear attention's $O(T)$ complexity versus the $O(T^2)$ complexity of the standard attention mechanism is essential in efficiently handling long transaction histories. This makes the LTACNN scalable to real fraud detection scenarios with long sequences.

**Translation Invariance (Convolution):** Convolutional layers have some translation invariance, i.e., the model will learn to identify similar patterns regardless of their location in the sequence. This is helpful in fraud detection, where fraud patterns are not always at the exact location in the transaction history.

## IV. MODEL FRAMEWORK: TRAINING AND OVERSIGHT

This section and Figure 2 collectively describe the fraud detection system's general training and deployment architecture. The architecture combines a "black box" high-accuracy classification model with an interpretable Kolmogorov-Arnold Network (KAN) to facilitate oversight. Anomaly detection triggers the KAN's call to explain suspicious transactions.

### A. TRAINING PROCEDURE

The training process has the following steps:

**Data Preprocessing:**

- Takes raw transaction data and transforms it into a stream of feature vectors in preparation for the LTACNN (described in Section III). Preprocessing may include feature scaling, one-hot encoding of categorical features, and imputing missing data.
- The imbalanced data is processed with ADASYN (Adaptive Synthetic Sampling). ADASYN produces synthetic samples from the minority class (fraud) and levels out the training set.

This strengthens the ability of the model to learn to detect fraud.

**Feature Extraction:** The preprocessed transaction sequences are fed into the LTACNN feature extractor. This produces feature vectors, denoted by $\mathbf{z}_i$, for each transaction $i$. These vectors capture functional temporal patterns.

**Black Box preprocessed:** The features extracted are employed to train a "black box" classification model. This may be a deep neural network, a gradient boosting machine, or another high-capacity classifier. The training data is a collection of pairs $\{\mathbf{z}_i, y_i\}$, with $\mathbf{z}_i$ being the feature vector and $y_i \in \{0, 1\}$ being the fraud label (0 for good, 1 for fraudulent). The model is trained to minimize a loss function appropriate for binary classification, such as cross-entropy loss. An optimisation like Adam or SGD is used.

**KAN Training (Parallel or Conditional):** A KAN is trained to acquire the black box model's decision function approximation. Training can be done in either of the two following ways:

- **Parallel Training:** The KAN is trained in parallel with the black box model using the same dataset (or an analogous one). This enables the KAN to learn a generalization of the black box model's behaviour.
- **Conditional Training:** The KAN is conditioned on the flagged data points only employing the oversight mechanism (described below). This is analogous to fine-tuning the KAN to the most challenging or questionable instances. This is less expensive, but the KAN may not learn the complete decision boundary well.

In either case, the input features are the KAN training data and the black box model's output probabilities (or logits).

### B. HUMAN/AUTONOMOUS OVERSIGHT SERVICE

The monitoring service tracks the black box model predictions and employs the KAN for explanations as and when required.

**Anomaly Detection in Predictions:** The black box model, trained using the dataset, produces predictions of the probability of fraud for novel, unseen transactions. Anomaly detection software monitors these predictions for potentially

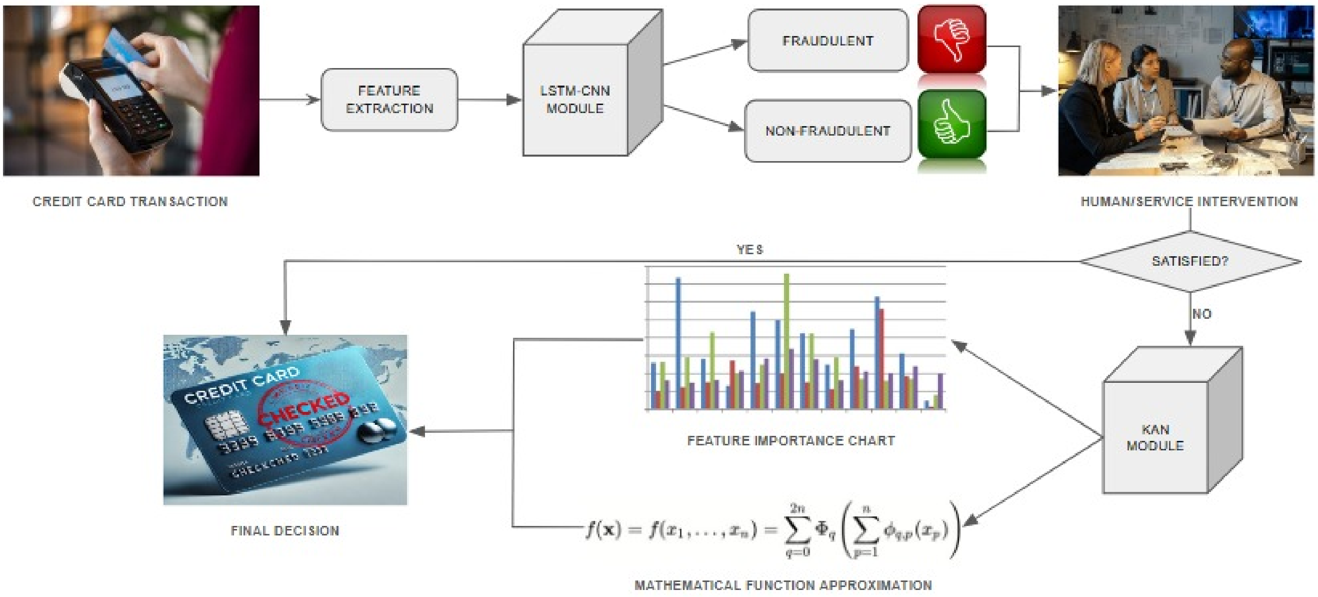

FIGURE 2: The image illustrates our credit card fraud detection system using an LSTM-CNN model and KAN, integrating feature importance analysis and human/service intervention for final decision-making.

suspicious or uncertain examples. Such a system could utilize one of the following methods:

- **Uncertainty Estimation:** Uncertainty can be estimated through methods like Monte Carlo dropout, ensembling, or temperature scaling. High uncertainty triggers the oversight mechanism.
- **Thresholding on Prediction Scores:** The predicted probability of fraud is thresholded. Predictions near the threshold (e.g., near 0.5 for binary classification) are flagged for investigation.
- **Rule-Based Systems:** Rules, drawn from expert knowledge, flag suspicious transactions (e.g., massive amounts, unusual locations).
- **Outlier Detection:** Statistical methods identify those substantially different from legitimate transactions.

**KAN Trigger and Feature Importance:** If the anomaly detection system detects a suspicious transaction, the following occurs:

- The KAN receives the transaction data (raw data or LTACNN features) as its input.
- The KAN approximates the decision function of the black box model for the specific transaction. KANs are more explainable (with univariate functions on edges), which is why the black box model predicted this.
- The geometry of KANs reveals feature importance. The magnitude of the learned univariate functions over edges emanating from input nodes reveals the importance of each feature. A plausible measure of feature importance is:

$$\text{Importance}(x_i) = \sum_{j \in \text{Outgoing}(i)} \left( \int |\phi_{i,j}(x)| dx \right), \quad (9)$$

Where $\text{Outgoing}(i)$ denotes the set of nodes that can be reached from input node $i$, and $\phi_{i,j}(x)$ denotes the univariate function along the edge from input node $i$ to node $j$. The integral, representing the "area under the curve," quantifies its impact. Other straightforward measures, such as the maximum value, can also be used.

**Human Review (Optional):** The KAN's output (approximated decision function and feature importances) can be manually reviewed by a human expert. The expert checks if the prediction is correct, reads the justification, and possibly discovers new fraud schemes.

Figure 3 provides a more visual understanding of the process using a data flow model.

### C. ITERATIVE REFINEMENT

The framework is iterative. Findings from the KAN and oversight service can improve both models:

- **Black Box Model Improvement:** KAN's knowledge can uncover the black box model's shortcomings (e.g., overreliance on insignificant features, miscalibrated decision boundaries). Training data, structure, or hyperparameter improvement of the black box model is facilitated.
- **KAN Refinement:** The KAN training can be supplemented with examples found by the monitoring mechanism to add, besides stress on complex cases.

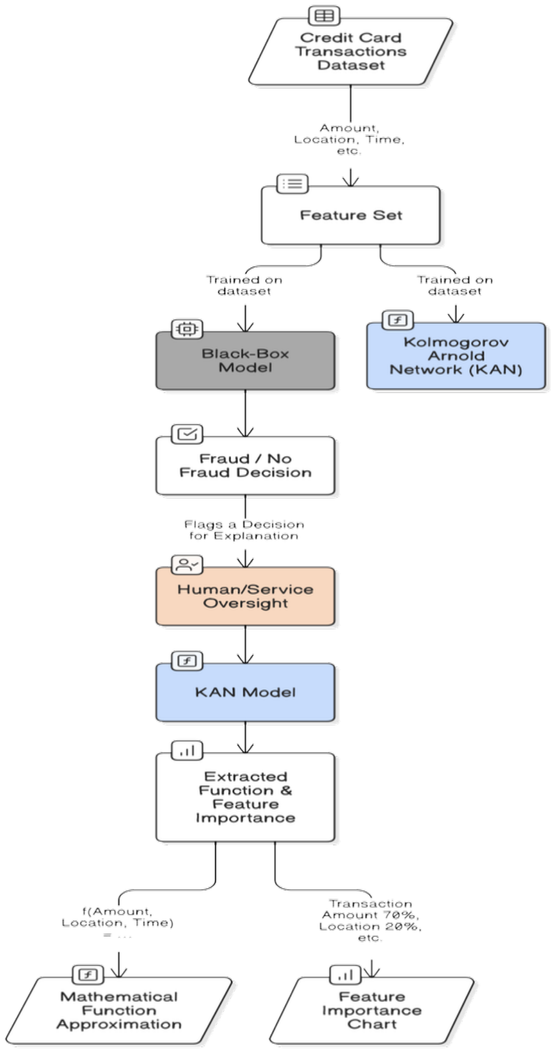
Credit Card Transactions Dataset
Amount, Location, Time, etc.
Feature Set
Trained on dataset
Trained on dataset
Black-Box Model
Kolmogorov Arnold Network (KAN)
Fraud / No Fraud Decision
Flags a Decision for Explanation
Human/Service Oversight
KAN Model
Extracted Function & Feature Importance
f(Amount, Location, Time) = ...
Transaction Amount 70%, Location 20%, etc.
Mathematical Function Approximation
Feature Importance Chart

FIGURE 3: Flowchart of Fraud Detection and Feature Importance Extraction from a Black-Box Model (like LSTM-CNN) and Kolmogorov Arnold Network

## V. DATASET DESCRIPTION

The information, which was sourced from www.kaggle.com, is the purchases of European consumers using credit cards over two days in September 2013. It holds 284,807 transactions, 492 of which are reported as fraudulent [31]. The dataset comprises 31 features, including the transaction date, transaction amount, and 28 anonymised labels from V1 to V28. The target variable, 'Class', is a binary indicator where '1' indicates a fraudulent transaction and '0' indicates a typical transaction.

To balance the uneven classes in the dataset, we will apply the ADASYN description of sampling to create more synthetic samples for the minority class. ADASYN adaptively modifies the distribution of synthetic samples based on the minority class instance density, creating more synthetic samples in areas with higher class imbalance. ADASYN will assist the model in learning to identify fraudulent transactions better by lessening the model's bias towards the majority class and enhancing overall classification performance. This model will learn more representative fraudulent patterns, resulting in a more stable and balanced fraud detection system.

## VI. EVALUATION STRATEGY

For credit card fraud detection, fraud detection systems' main goal is to avoid false positives and false negatives but with more focus on preventing false negatives because they significantly affect cost and inflict damage to customer trust. A false negative (FN) happens when a fraudulent transaction is wrongly classified as legitimate, which results in instant financial loss. A false positive (FP) is when a legitimate transaction is classified as fraudulent; a false positive (FP) will inconvenience customers but not cause economic damage.

We employ a confusion matrix to rigorously assess our proposed fraud detection model, differentiating the classification results. From this matrix, we will obtain crucial evaluation metrics, including Accuracy, Precision, and Recall (Sensitivity).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{10}$$

$$Precision = \frac{TP}{TP + FP}. \tag{11}$$

$$Recall = \frac{TP}{TP + FN}. \tag{12}$$

True Positives (TP) are the accurately detected fraudulent transactions, and True Negatives (TN) are valid transactions accurately detected as non-fraudulent. False Positives (FP) are valid transactions falsely detected as dishonest, and False Negatives (FN) are fraudulent transactions falsely detected as valid.

Since the cost of undetected fraud is so high, Recall (Sensitivity) will be a significant evaluation measure. It will capture the precision of the model in identifying suspicious transactions. We will also investigate the Precision measure to ensure that fraudulent transactions reported are indeed dishonest and do not unnecessarily inconvenience honest customers.

In addition to these traditional metrics, we also measure model performance using the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC), which give us insights into the model's ability to distinguish models from authentic transactions at various classification thresholds.

## VII. RESULTS AND DISCUSSION

The LSTM-CNN model in this research showed superior classification performance compared to traditional machine learning models and the LSTM-Attention model. Using LSTM and CNN in complementary integration successfully resolved temporal dependencies and spatial patterns in transaction data, leading to improved accuracy in fraud detection.

TABLE 1: **Results of models**

| **Algorithms** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|
| GRU [29] | – | 0.8626 | 0.7208 |
| LSTM [29] | – | 0.8575 | 0.7408 |
| SVM [30] | 0.9349 | 0.9743 | 0.8976 |
| KNN [30] | 0.9982 | 0.7142 | 0.0393 |
| ANN [30] | 0.9992 | 0.8115 | 0.7619 |
| LSTM-attention [28] | 0.9672 | 0.9885 | 0.9191 |
| LSTM-CNN-attention | 0.9434 | 0.9880 | 0.9782 |
| KAN-approx | 0.9234 | 0.9790 | 0.9701 |

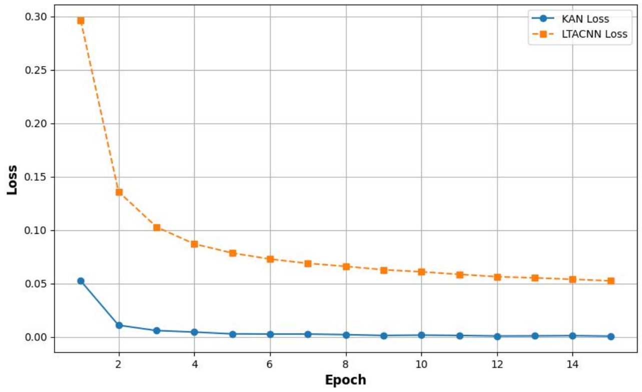

FIGURE 4: Graph of loss curves of models LTACNN and KAN.

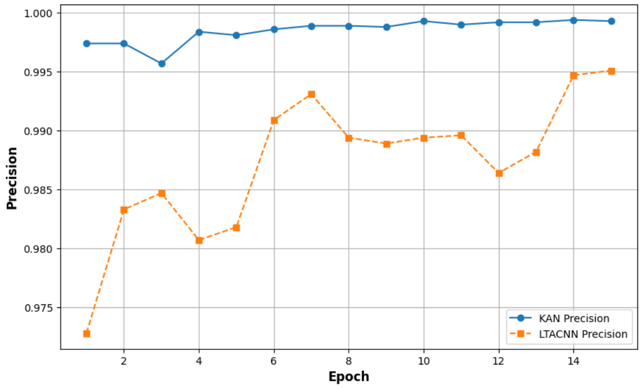

FIGURE 5: Comparison of KAN and LTACNN Precision over Epochs.

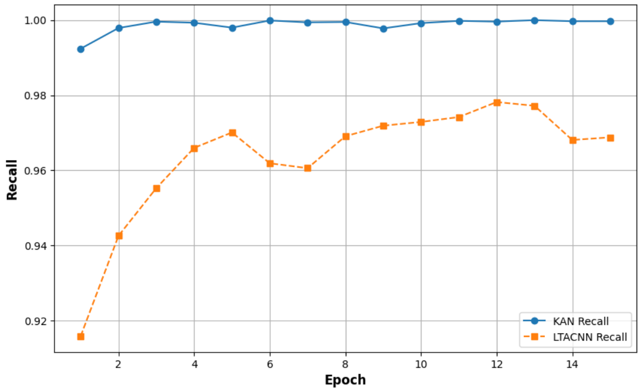

FIGURE 6: Comparison of KAN and LTACNN Recall over Epochs.

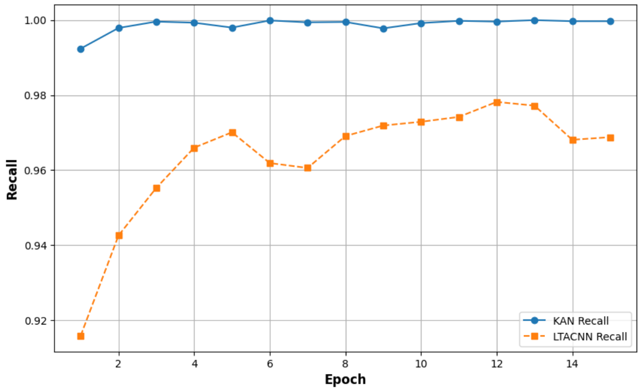

FIGURE 7: Comparison of KAN and LTACNN ROC over Epochs.

Despite the success of the LSTM-Attention model in focusing on significant features, it showed slightly poorer performance, which reflects that the use of CNN positively affects feature extraction processes. In addition, the Kolmogorov-Arnold Networks (KAN) model improved interpretability by approximating the decision boundaries of the LSTM-CNN model, thus providing valuable insights into determinants in fraud classification.

The y-axis in Figure 4 indicates the loss value, while the x-axis shows the epochs. Small loss values indicate better model convergence and optimization, and the figure depicts how the KAN Model achieves significantly lower loss than the LTACNN model. Both models start with high loss values, yet the KAN Model experiences a far greater loss reduction rate, which eventually tends close to zero in the initial few iterations. In comparison, the LTACNN model exhibits a slower loss decrease, reaching a higher equilibrium point, thus depicting slower convergence.

As illustrated in Figures 5, 6, 7 the KAN model consistently surpasses the LTACNN model on all critical performance indicators: ROC-AUC, recall, and precision. It attains an almost perfect ROC-AUC early in the training process and maintains this stability, which reflects its enhanced capability to differentiate between classes with considerable confidence effectively. Conversely, LTACNN exhibits a gradual yet less rapid enhancement in ROC-AUC, failing to achieve a comparable level of performance as the KAN model.

About recall, the KAN model achieves almost optimal scores in the initial stages of training so that there are virtually no false negatives. The LTACNN model keeps improving but is still lagging, indicating that it might leave out some positive samples. Likewise, the KAN model achieves exceptionally

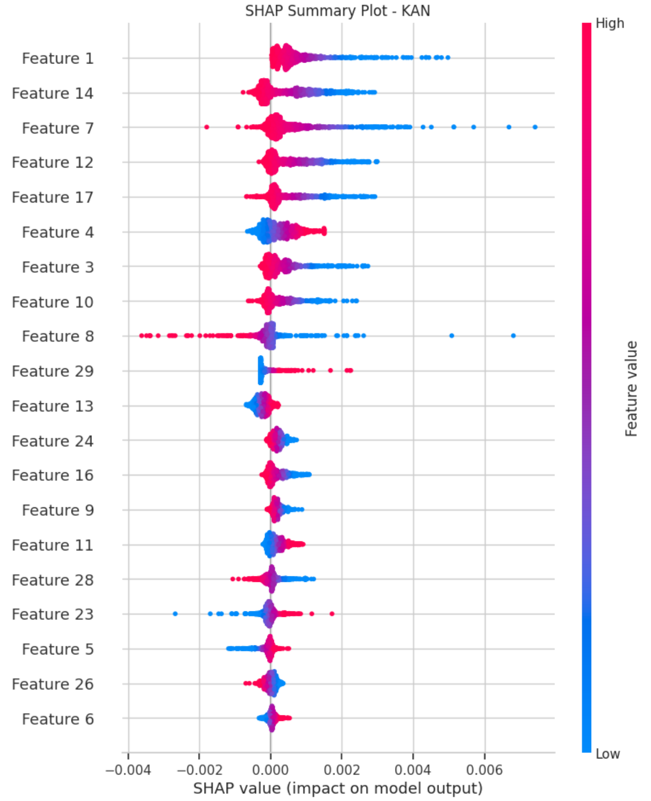

FIGURE 8: SHAP summary plot of KAN model.

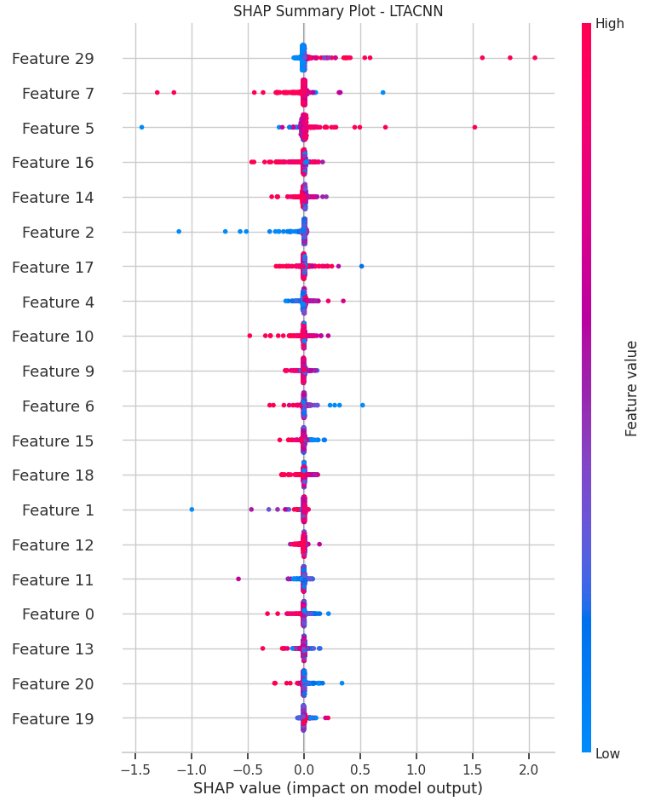

FIGURE 9: SHAP summary plot of LTACNN model.

high precision, i.e., fewer false positives, while LTACNN, though better, can't match its score. This indicates the KAN model's excellence in high-stakes classification problems where precision and recall are paramount.

SHAP summary plots of LTACNN Figure 9 and KAN Figure 8 reveal both models place the most significant emphasis on shared features when making predictions. Feature 1, Feature 7, Feature 14, Feature 17, and Feature 10 are notable here, with high SHAP values in both models as they represent the most significant effect on model predictions. Aside from similarities and differences in the magnitude of SHAP values between the models, the shared overlap in the significance of these determinative features suggests both architectures are making decisions based on shared information. Such convergence of feature importance is in line with the stability and replicability of the predictor factors across model architectures.

The Wilcoxon signed-rank test is a nonparametric test used to compare two related samples. Unlike the paired t-test, it does not assume a normal distribution and can be used for deep learning models when the data distribution is unknown or skewed. It ranks the paired differences in absolute value and tests whether the median difference significantly differs from zero.

In the results shown, the Wilcoxon test was employed to compare LTACNN with KAN based on both SHAP and LIME explanations. The p-value of the SHAP LTACNN vs. SHAP KAN test was 0.87, and that of the LIME LTACNN vs. LIME KAN test was 0.38. Since both p values are much more significant than the conventional significance value (0.05), we cannot reject the null hypothesis, i.e., we cannot conclude there is any statistically significant difference in LTACNN and KAN's feature attributions.

This result also validates the KAN's prior SHAP analysis. Both models make predictions based on similar feature importance distributions, affirming the stability and reliability of their decision-making mechanisms.

## VIII. CONCLUSION

In this paper, we proposed a hybrid deep learning model for credit card fraud detection that integrates LSTM, CNN, Attention and KAN mechanisms to enhance the accuracy of fraud classification and interpretability. The LSTM model effectively captures temporal dependencies among transaction sequences, and CNN captures spatial dependencies among the data. The KAN model provides functional approximation capability, enhancing the generalisation attention mechanism to ensure the model's attention on informative transaction patterns. The integrated framework absorbs the strengths of each approach and presents a more interpretable and robust fraud detection system.

To verify our approach, we conducted large-scale experiments on credit card fraud datasets and showed that our model performs better than conventional deep learning models.

SHAP analysis also verified that KAN and LTACNN (LSTM-CNN with Attention) rely on identical feature importance distributions, demonstrating the consistency of their decision-making. Statistical analysis with the Wilcoxon signed-rank test also showed no substantial difference between the two models' feature attributions, showing the consistency of our hybrid approach.

## REFERENCES

[1] C. Yu, Y. Xu, J. Cao, Y. Zhang, Y. Jin and M. Zhu, "Credit Card Fraud Detection Using Advanced Transformer Model," in *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*, Hong Kong, China, 2024, pp. 343–350, doi: 10.1109/MetaCom62920.2024.00064.

[2] Y. Tang and Z. Liu, "A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning," in *IEEE Access*, vol. 12, pp. 182547–182560, 2024, doi: 10.1109/ACCESS.2024.3491175.

[3] P. Singh, K. Singla, P. Piyush and B. Chugh, "Anomaly Detection Classifiers for Detecting Credit Card Fraudulent Transactions," in *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India, 2024, pp. 1–6, doi: 10.1109/ICAECT60202.2024.10469194.

[4] P. Kumari and S. Mittal, "Fraud Detection System for Financial System Using Machine Learning Techniques: A Review," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2024, pp. 1–6, doi: 10.1109/ICRITO61523.2024.10522197.

[5] A. Rawat, S. S. Aswal, S. Gupta, A. P. Singh, S. P. Singh and K. C. Purohit, "Performance Analysis of Algorithms for Credit Card Fraud Detection," in *2024 2nd International Conference on Disruptive Technologies (ICDT)*, Greater Noida, India, 2024, pp. 567–570, doi: 10.1109/ICDT61202.2024.10489771.

[6] M. Thilagavathi, R. Saranyadevi, N. Vijayakumar, K. Selvi, L. Anitha and K. Sudharson, "AI-Driven Fraud Detection in Financial Transactions with Graph Neural Networks and Anomaly Detection," in *2024 International Conference on Science Technology Engineering and Management (ICSTEM)*, Coimbatore, India, 2024, pp. 1–6, doi: 10.1109/ICSTEM61137.2024.10560838.

[7] V. R. Adhegaonkar, A. R. Thakur and N. Varghese, "Advancing Credit Card Fraud Detection Through Explainable Machine Learning Methods," in *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2024, pp. 792–796, doi: 10.1109/IDCIoT59759.2024.10467999.

[8] N. R. S. Jebaraj, J. Shekhawat and R. Gupta, "An Overview of Clustering Algorithms for Credit Card Fraud Detection," in *2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC)*, Debre Tabor, Ethiopia, 2024, pp. 1–6, doi: 10.1109/ICOCWC60930.2024.10470724.

[9] R. Raut, A. B. Chandanshive, P. N. Gadkar and E. Govardhan, "Credit Card Fraud Detection Using Ensemble Modeling," in *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, Raigarh, India, 2024, pp. 1–6, doi: 10.1109/OTCON60325.2024.10687633.

[10] J. G. Sherwin Akshay, T. Vinusha, R. Sharon Bianca, C. K. Sarath Krishna and G. Radhika, "Enhancing Credit Card Fraud Detection with Deep Learning and Graph Neural Networks," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICT)*, Kamand, India, 2024, pp. 1–6, doi: 10.1109/ICCCNT61001.2024.10725042.

[11] G. Bharath and P. S. Uma Priyadarsini, "A Novel Accuracy Analysis of Credit Card Fraud Detection Using ResNet50 Over Linear Regression," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, 2024, pp. 1–5, doi: 10.1109/ICCCNT61001.2024.10725812.

[12] S. Sukruth, M. Haripriya, S. Deepa, J. Jayapriya and M. Vinay, "Comparative Study on GANs and VAEs in Credit Card Fraud Detection," in *2024 First International Conference on Software, Systems and Information Technology (SSITCON)*, Tumkur, India, 2024, pp. 1–6, doi: 10.1109/SSITCON62437.2024.10796972.

[13] V. Suganthi and J. Jebathangam, "A Novel Approach for Credit Card Fraud Detection using Gated Recurrent Unit (GRU) Networks," in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, Nepal, 2024, pp. 1716–1721, doi: 10.1109/I-SMAC61858.2024.10714795.

[14] T. -T. -H. Le, Y. Hwang, H. Kang and H. Kim, "Robust Credit Card Fraud Detection Based on Efficient Kolmogorov-Arnold Network Models," in *IEEE Access*, vol. 12, pp. 157006–157020, 2024, doi: 10.1109/ACCESS.2024.3485200.

[15] S. Jhansi Ida, K. Balasubadra, S. R R and L. N. T, "Enhancing Credit Card Fraud Detection through LSTM-Based Sequential Analysis with Early Stopping," in *2024 2nd International Conference on Networking and Communications (ICNWC)*, Chennai, India, 2024, pp. 1–6, doi: 10.1109/ICNWC60771.2024.10537550.

[16] S. A, N. V and S. S. Pandi, "A Novel Approach for Credit Card Fraud Detection Using Deep Learning Algorithms," in *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*, Kollam, India, 2024, pp. 1870–1875, doi: 10.1109/ICCPCT61902.2024.10672824.

[17] K. G. Dastidar, O. Caelen and M. Granitzer, "Machine Learning Methods for Credit Card Fraud Detection: A Survey," in *IEEE Access*, vol. 12, pp. 158939–158965, 2024, doi: 10.1109/ACCESS.2024.3487298.

[18] E. Ileberi and Y. Sun, "A Hybrid Deep Learning Ensemble Model for Credit Card Fraud Detection," in *IEEE Access*, vol. 12, pp. 175829–175838, 2024, doi: 10.1109/ACCESS.2024.3502542.

[19] I. D. Mienye and N. Jere, "Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions," in *IEEE Access*, vol. 12, pp. 96893–96910, 2024, doi: 10.1109/ACCESS.2024.3426955.

[20] Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou and M. Li, "Learning Transactional Behavioral Representations for Credit Card Fraud Detection," in *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 5735–5748, Apr. 2024, doi: 10.1109/TNNLS.2022.3208967.

[21] Imran, O. Yakoob, A., "Leveraging LSTM and Attention for High-Accuracy Credit Card Fraud Detection," in *Fusion: Practice and Applications*, 2025, pp. 209–220, doi: 10.54216/FPA.170115.

[22] P. Pandey and K. K. Garg, "Credit Card Fraud Detection Using KNC, SVC, and Decision Tree Machine Learning Algorithms," in *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, Houston, TX, USA, 2025, pp. 1–3, doi: 10.1109/ICAIC63015.2025.10848573.

[23] B. Dharma and D. Latha, "Fraud Detection in Credit Card Transactional Data Using Hybrid Machine Learning Algorithm," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, Goathgaun, Nepal, 2025, pp. 213–218, doi: 10.1109/ICMCSI64620.2025.10883549.

[24] F. Khaled Alarfaj and S. Shahzadi, "Enhancing Fraud Detection in Banking With Deep Learning: Graph Neural Networks and Autoencoders for Real-Time Credit Card Fraud Prevention," in *IEEE Access*, vol. 13, pp. 20633–20646, 2025, doi: 10.1109/ACCESS.2024.3466288.

[25] N. Damanik and C. -M. Liu, "Advanced Fraud Detection: Leveraging K-SMOTEENN and Stacking Ensemble to Tackle Data Imbalance and Extract Insights," in *IEEE Access*, vol. 13, pp. 10356–10370, 2025, doi: 10.1109/ACCESS.2025.3528079.

[26] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention," in *Proc. of the 37th International Conference on Machine Learning (ICML)*, Online, PMLR 119: 5650–5659, 2020, doi: 10.48550/arXiv.2006.16236.

[27] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov–Arnold Networks," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2025, [Online]. Available: refhttps://arxiv.org/abs/2404.19756https://arxiv.org/abs/2404.19756.

[28] I. Benchaji, S. Douzi, B. El Ouahidi, and J. Jaafari, "Enhanced credit card fraud detection based on attention mechanism and LSTM deep model," *Journal of Big Data*, vol. 8, article 151, 2021, doi: 10.1186/s40537-021-00541-8.

[29] J. Forough and S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," *Appl. Soft Comput.*, vol. 99, 106883, 2021, doi: 10.1016/j.asoc.2020.106883.

[30] A. R. Asha and S. K. Kumar, "Credit card fraud detection using artificial neural network," *Glob. Transitions Proc.*, vol. 2, no. 1, pp. 35–41, 2021, doi: 10.1016/j.gltp.2021.01.006.

[31] A. Dal Pozzolo, O. Caelen, R. A. Johnson and G. Bontempi, "Credit Card Fraud Detection Dataset," ULB Machine Learning Group, 2015. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

[32] Wikipedia contributors, "Long short-term memory," Wikipedia, The Free Encyclopedia, Available: https://en.wikipedia.org/wiki/Long_short-term_memory.

[33] M.-I. Gheorghe, S. Mihalache and D. Burileanu, "Using Deep Neural Networks for Detecting Depression from Speech," in *2023 31st European Signal Processing Conference (EUSIPCO)*, Helsinki, Finland, 2023, pp. 411–415, doi: 10.23919/EUSIPCO58844.2023.10289973.

[34] Hesam Sheikh, "Understanding Kolmogorov–Arnold Networks (KAN)," *TDS Archive*, May 7, 2024. Available: https://medium.com/data-science/kolmogorov-arnold-networks-kan-e317b1b4d075.

**SAKSHAM SHARMA** is currently studying a B.Tech. in Computer Science and Engineering at Vellore Institute of Technology, Chennai (2021–2025). A great fan of video games, he is especially interested in using AI to enrich the gaming experience and make AI accessible to the masses. Apart from game mechanics analysis, he likes to study the thin line between algorithms and entertainment—sometimes reaching for codes to debug and dodging virtual monsters at other times. ORCID: 0009-0006-6177-6007

**SHAURYA DEV PATHAK** is currently pursuing a B.Tech. degree in Computer Science and Engineering from Vellore Institute of Technology, Chennai (2021–2025). From 2023 to 2024, he did his summer internship, where he developed an electronic property predictor for CTS thin films to accurately forecast electric properties based on atomic percentages of copper, tin, sulfur, and substrate temperature. He is the author of a publication in the International Journal of Alloys and Compounds via ScienceDirect and holds a patent related to the same research. His research interests include machine learning applications in materials science, electronic property prediction, and thin-film technology.

**SUSHANT GARGI** is currently pursuing B.Tech. degree in Computer Science and Engineering with a specialization in Intelligence and Robotics from Vellore Institute of Technology, Chennai (2021–2025). He is a tech enthusiast with knowledge of AI, machine learning, automation, and full-stack development. With knowledge of React, MongoDB, and cloud computing, he is inclined to develop smart decision-making and security systems. He is reflective and continues to search for breakthroughs in artificial intelligence, automation, and technology to solve intricate problems better.

**SUGANESHWARI G** received her PhD in Big Data Analytics from the Vellore Institute of Technology, India, and is currently a Senior Assistant Professor at VIT, Chennai. Her research focuses on big data analytics, recommendation systems, healthcare systems, and blockchain applications. She has published extensively in peer-reviewed journals, including IEEE Access, and has collaborated internationally on projects in federated learning, session-based recommendations, and deep learning for medical imaging. Dr Suganeshwari is an active member of ACM and CSTA, and she has reviewed leading journals and organised several workshops and conferences. ORCID: 0000-0002-0887-0196.