IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 September 2024, accepted 14 October 2024, date of publication 23 October 2024, date of current version 4 November 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3485200*

RESEARCH ARTICLE

# Robust Credit Card Fraud Detection Based on Efficient Kolmogorov-Arnold Network Models

**THI-THU-HUONG LE[1,2], YEONJEONG HWANG[3], HYOEUN KANG[4], AND HOWON KIM[3], (Member, IEEE)**

[1]Blockchain Platform Technology Center, Pusan National University, Busan 46241, South Korea
[2]IoT Research Center, Pusan National University, Busan 46241, South Korea
[3]School of Computer Science and Engineering, Pusan National University, Busan 46241, South Korea
[4]SmartM2M, Busan 48058, South Korea

Corresponding authors: Thi-Thu-Huong Le (lehuong7885@gmail.com) and Howon Kim (howonkim@pusan.ac.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00431, Development of open service platform and creation technology of federated intelligent digital twin, 100%).

**ABSTRACT** Credit card fraud detection remains a significant challenge in the financial industry, necessitating advanced models to identify fraudulent activities while minimizing false positives accurately. Traditional machine learning approaches, such as Multilayer Perceptrons (MLP), have been widely used but often struggle with interpretability and parameter optimization issues. Kolmogorov-Arnold Networks (KAN) present a promising alternative by addressing these limitations through their inherent structure, which allows for more interpretable and potentially more accurate models. This paper explores the application of KAN in the context of credit card fraud detection, motivated by the need for more effective and interpretable solutions. We implement and evaluate three MLP, KAN, and efficient KAN models using two publicly available credit card fraud datasets. Our experimental results demonstrate that both KAN and efficient KAN significantly outperform the traditional MLP model in terms of detection accuracy while reducing the number of parameters compared to MLP. The findings underscore the potential of KAN and its efficient variant as superior alternatives for credit card fraud detection, offering enhanced accuracy and interpretability. This study provides valuable insights into model performance and parameter efficiency, guiding future research and practical applications in fraud detection systems.

**INDEX TERMS** Credit card fraud detection, financial security, Kolmogorov-Arnold networks, multilayer perceptron.

## I. INTRODUCTION

Credit card fraud has become pervasive and complex in recent years, posing significant challenges to financial institutions, businesses, and consumers. While the digital era has brought unprecedented convenience to financial transactions, it has simultaneously facilitated the emergence of increasingly sophisticated fraudulent activities. The December 2022 issue of the Nilson Report [1] projects that global card fraud losses across issuers, merchants, and acquirers will reach a staggering $397.40 billion over the next decade. This alarming forecast underscores the critical need for robust and efficient fraud detection systems to safeguard financial ecosystems and protect stakeholders.

Traditional approaches to credit card fraud detection primarily rely on machine learning models, with Multilayer Perceptrons (MLPs) being a popular choice due to their pattern recognition capabilities. However, despite their effectiveness in identifying fraudulent transactions, MLPs often suffer from inherent complexity, resulting in opaque decision-making processes. This lack of transparency poses significant challenges for stakeholders attempting to interpret model outputs and integrate fraud detection systems into operational workflows [2]. Furthermore, as highlighted by [3], the black-box nature of MLPs can make it difficult to explain decisions to regulatory bodies and customers, potentially

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

eroding trust in financial institutions and their fraud detection mechanisms.

There is growing interest in adopting interpretable models for fraud detection to address these challenges. Kolmogorov-Arnold Networks (KAN) offer a promising solution by combining the predictive power of neural networks with interpretability akin to decision trees [4]. KANs are structured to decompose complex decision processes into interpretable steps, enhancing transparency and providing insights into the drivers of fraud detection. This approach aligns with the increasing demand for explainable AI in financial services, as highlighted by [11].

This paper investigates the application of KAN alongside traditional MLP models for credit card fraud detection. By leveraging the hierarchical structure of KAN, we aim to enhance detection accuracy and interpretability in fraud detection systems. We conduct experiments using established datasets widely employed in fraud detection research, comparing the performance of MLP, KAN, and an efficient variant of KAN. Our analysis evaluates accuracy, interpretability, and computational efficiency, illuminating the trade-offs between model complexity and performance in the fraud detection domain.

The dynamic nature of fraudulent activities further complicates credit card fraud detection. As noted by [14], fraudsters continuously adapt their strategies, necessitating fraud detection systems that can evolve and remain effective over time. This adaptability is particularly crucial given the imbalanced nature of fraud datasets challenge [12], [13], where legitimate transactions far outnumber fraudulent ones [15]. Our study addresses these challenges by exploring the potential of KAN models to provide high accuracy and interpretability in this imbalanced fraud dataset.

This paper contributes significantly to the field of credit card fraud detection in several ways:

- Introduction of Effective KAN Models: We present the first comprehensive study of KAN and its efficient variant for credit card fraud detection using two publicly available datasets. This builds upon the works of [4] and [5] by extending KAN applications to the critical domain of financial fraud detection.
- Comparative Analysis: We conduct a thorough comparison of MLP, KAN, and an optimized KAN variant across these datasets, highlighting their respective strengths and weaknesses in terms of accuracy, precision, recall, and F1-score.
- Efficiency Evaluation: We provide insights into the computational efficiency and resource utilization of KAN models compared to traditional MLPs by examining training and inference times.

The remainder of the paper is structured as follows:

Section II details the datasets used, describes the MLP and KAN models, and introduces the efficient variant of KAN. It also comprehensively reviews recent advancements in interpretable machine learning for fraud detection. Section III outlines the preprocessing steps, implementation details for each model, and the parameter settings and tuning strategies employed. We also discuss the challenges of handling imbalanced datasets in fraud detection and our approach to addressing this issue. Section IV presents experimental results and performance metrics, comparing MLP, KAN, and efficient KAN models and includes an analysis of the findings. Section VI interprets experimental results, compares model complexities, and discusses the interpretability and practical implications of KAN models in fraud detection systems. Section VII summarizes key findings, highlights contributions, and suggests future research directions, including the potential integration of KAN models with other advanced techniques, such as ensemble methods and deep learning approaches for fraud detection.

## II. LITERATURE REVIEW

Credit card fraud detection remains a paramount concern in the financial sector, driving continuous innovation in detection methodologies. This section presents a comprehensive overview of existing fraud detection approaches, critically examines the limitations inherent in traditional MLP models, and introduces KAN as a promising alternative. We explore how KAN models potentially address the challenges faced by conventional techniques, offering a balance between predictive power and interpretability crucial in the evolving landscape of financial security.

### A. TRADITIONAL APPROACHES TO CREDIT CARD FRAUD DETECTION

Early fraud detection systems predominantly relied on rule-based methods, employing predetermined criteria to flag potentially fraudulent transactions [16], [17]. While effective for known fraud patterns, these systems struggled to adapt to novel, evolving fraud techniques. This limitation catalyzed the adoption of machine learning approaches, which offered enhanced flexibility and adaptability.

Supervised learning techniques have gained prominence in fraud detection due to their ability to learn from historical data. Bhattacharyya et al. [18] conducted a comparative analysis of support vector machines (SVMs), random forests, and logistic regression in fraud detection, concluding that random forests generally outperformed other methods. Yee et al. [19] applied supervised classification using various Bayesian network classifiers, including K2, Tree Augmented Naïve Bayes (TAN), Naïve Bayes, logistic regression, and J48 classifiers. However, these techniques often face challenges related to imbalanced datasets, where legitimate transactions significantly outnumber fraudulent ones, potentially skewing model performance.

Recent studies have continued to advance the field by exploring new methodologies and improving existing models. Vengatesan et al. [20] discussed the application of various data analytic techniques in credit card fraud detection, emphasizing the effectiveness of advanced data processing methods. Asha et al. [21] explored using artificial neural networks, which showed promise in capturing complex

patterns in transaction data. Alfaiz and Fati [22] presented an enhanced model using machine learning techniques to improve fraud detection accuracy. These works highlight ongoing efforts to address the challenges of imbalanced datasets and improve detection capabilities, suggesting that hybrid approaches and novel algorithms could further enhance fraud detection performance.

### B. NEURAL NETWORK APPROACHES AND MLPS

Neural networks, particularly Multilayer Perceptrons (MLPs), have gained prominence in credit card fraud detection due to their capacity to model complex, non-linear relationships in data [23]. West and Bhattacharya [24] provided a comprehensive review of computational intelligence techniques in fraud detection, emphasizing the effectiveness of neural networks in capturing subtle patterns indicative of fraud.

Riffi et al. [25] introduced fraudulent transaction detection using MLP and Extreme Learning Machine (ELM) applied to credit card fraud datasets. Esenogho et al. [26] proposed an MLP ANN-based model solution to enhance the accuracy of credit card fault detection.

Despite their success, MLPs face several limitations in the context of fraud detection: (1) Lack of Interpretability: As highlighted by [27] and [28], the black-box nature of MLPs impedes explanation of their decision-making process, which is crucial in financial applications where regulatory compliance and customer trust are paramount. (2) Overfitting: MLPs with complex architectures are prone to overfitting, especially when dealing with imbalanced datasets typical in fraud detection scenarios [14], [29], [30]. (3) Computational Intensity: Training and updating large MLPs can be computationally expensive, potentially limiting their real-time application in high-volume transaction environments [31], [32], [33].

### C. ENSEMBLE METHODS AND ADVANCED TECHNIQUES

Early fraud detection systems predominantly relied on rule-based methods, employing predetermined criteria to flag potentially fraudulent transactions [16], [17]. While effective for known fraud patterns, these systems struggled to adapt to novel, evolving fraud techniques. This limitation catalyzed the adoption of machine learning approaches, which offered enhanced flexibility and adaptability.

Supervised learning techniques have gained prominence in fraud detection due to their ability to learn from historical data. Bhattacharyya et al. [18] conducted a comparative analysis of support vector machines (SVMs), random forests, and logistic regression in fraud detection, concluding that random forests generally outperformed other methods. Yee et al. [19] applied supervised classification using various Bayesian network classifiers, including K2, Tree Augmented Naïve Bayes (TAN), Naïve Bayes, logistic regression, and J48 classifiers. However, these techniques often face challenges related to imbalanced datasets, where legitimate transactions significantly outnumber fraudulent ones, potentially skewing model performance. For example, Bhattacharyya et al. [18] report a sensitivity of 0.65 and a specificity of 0.98, indicating that while the model performs well in identifying non-fraudulent transactions, it struggles significantly with detecting fraudulent ones.

Recent studies have continued to advance the field by exploring new methodologies and improving existing models. Vengatesan et al. [20] discussed the application of various data analytic techniques in credit card fraud detection, emphasizing the effectiveness of advanced data processing methods. Asha and KR [21] explored using artificial neural networks, which showed promise in capturing complex patterns in transaction data. Alfaiz and Fati [22] presented an enhanced model using machine learning techniques to improve fraud detection accuracy. These works highlight ongoing efforts to address the challenges of imbalanced datasets and improve detection capabilities, suggesting that hybrid approaches and novel algorithms could further enhance fraud detection performance. The imbalance between legitimate ($Y = 0$) and fraudulent ($Y = 1$) transactions challenges traditional machine learning models like MLPs in fraud detection. These models tend to overfit the majority class, leading to biased predictions and poor fraud detection. Metrics such as accuracy can be misleading in this context, making precision, recall, and F1-score more relevant [18].

Ensemble methods have also been explored to overcome the limitations of individual models in fraud detection. Pozzolo et al. [34] proposed a hybrid approach combining random forests with Poisson bagging to handle class imbalance effectively. Zareapoor and Shamsolmoali [35] demonstrated the effectiveness of bagging ensembles of MLPs in enhancing detection accuracy. More recent advancements include deep learning techniques. Jurgovsky et al. [36] utilized Long Short-Term Memory (LSTM) networks to capture temporal dependencies in transaction sequences, showing improvements over traditional MLPs. Alghofaili et al. [37] extended LSTM-based detection to financial fraud in the context of big data, although such methods often exacerbate interpretability issues. Purohit and Vishwakarma [38] employed XGBoost and Convolutional Neural Networks (CNNs) to build their fraud detection model. Mienye and Sun [39] proposed a robust deep learning approach combining LSTM and gated recurrent unit (GRU) neural networks as base learners in a stacking ensemble framework, with a multilayer perceptron (MLP) as the meta-learner. Soni et al. [40] introduced a deep learning-based framework incorporating various unsupervised learning algorithms for credit card fraud detection. Tang et al. [41] federated training techniques and their application in extending transaction graphs, which reinforces connections between financial institution data and enriches their coverage of advanced fraud detection algorithms.

For future advancements, Mim et al. [42] proposed a soft voting ensemble learning approach, which integrates multiple models to improve detection accuracy. Additionally,

Tayebi and El Kafhali [43] explored hyperparameter optimization using differential evolution to enhance credit card fraud detection models. El Kafhali and Tayebi [44] focused on XGBoost-based solutions for detecting fraudulent transactions, further highlighting the potential of ensemble methods and optimized algorithms in improving detection performance.

These studies underscore the growing sophistication and variety in fraud detection methods, reflecting an ongoing trend toward integrating advanced machine learning and deep learning techniques to address complex fraud detection challenges.

### D. INTERPRETABLE MODELS AND KAN

KAN offers a solution through its hierarchical structure, which incorporates customized loss functions, adaptive learning, and regularization techniques to prevent overfitting and ensure robust performance [45]. Unlike MLPs, KANs provide greater interpretability, making them better suited for real-world fraud detection where transparency is crucial.

The growing demand for model interpretability has renewed interest in inherently transparent models. Decision trees and rule-based systems have regained attention due to their clarity, albeit potentially at the cost of reduced accuracy compared to more complex models [46], [47], [48].

The potential of KAN in fraud detection lies in their ability to: (1) Provide clear decision paths, enhancing model interpretability. (2) Maintain high accuracy comparable to traditional neural networks. (3) Offer insights into the importance of different features in fraud detection.

An efficient variant of KAN [5], customized based on the original KAN, promises to overcome some limitations by reducing memory cost and simplifying computation to straightforward matrix multiplication. This efficient KAN operates seamlessly with both forward and backward passes, enhancing its practicality for real-world applications.

While KANs have shown promise in various domains such as satellite traffic forecasting task [6], discovering hidden physics and predicting dynamic evolution [7], [8], smart grid [9], their application to credit card fraud detection remains unexplored. This gap in the literature motivates our current study, which aims to evaluate the effectiveness of KANs compared to traditional MLPs in credit card fraud detection.

### E. EFFICIENCY CONSIDERATIONS IN FRAUD DETECTION MODELS

As financial transaction volumes continue to surge, the efficiency of fraud detection models becomes increasingly critical [10]. Carcillo et al. [31] highlighted the importance of scalable machine learning techniques for real-time fraud detection. This consideration has spurred research into model compression and efficient architectures, which could be applied to Kolmogorov-Arnold Networks (KANs) to enhance their practical applicability in high-volume transaction environments.

In conclusion, while significant strides have been made in credit card fraud detection, challenges persist in balancing model performance, interpretability, and efficiency. Our work addresses these challenges by exploring the potential of KAN models in this critical domain, aiming to contribute to the development of more effective and transparent fraud detection systems.

## III. METHODOLOGY

This section outlines the methodology employed for conducting experiments and evaluating the performance of Multilayer Perceptron (MLP) and Kolmogorov-Arnold Networks (KAN) models in credit card fraud detection. We describe the datasets used in our study, followed by detailed explanations of the MLP model, the KAN model, and any efficient variants of KAN utilized. The concept of our approach is depicted in Figure 1.

### A. DATASETS

In our experiment, we utilize two widely recognized and published datasets as detailed in Table 1. These datasets serve as established benchmarks in fraud detection, enabling us to evaluate the performance of different models under consistent conditions:

- **Credit Card Fraud Dataset** [49]: This dataset comprises transactions made by European cardholders over two days in September 2013. It contains 284,807 transactions, of which 492 (0.172%) are fraudulent. The dataset features numerically transformed variables (V1-V28) resulting from PCA, along with 'Time' and 'Amount' variables. 'Time' represents seconds elapsed since the first transaction, while 'Amount' denotes the transaction value. The target variable 'Class' is binary, with 1 indicating fraud and 0 otherwise.
- **Credit Card Fraud Mega Dataset** [50]: This extensive dataset encompasses 7,358,269 transactions, including 38,135 fraudulent cases. It comprises 25 input features and a binary output variable, where 1 signifies fraud and 0 indicates a normal transaction.

These datasets provide diverse scenarios for evaluating our models, offering varying levels of class imbalance and feature complexity.

Table 1 summarizes both datasets' input and output features. The Credit Card Fraud Dataset consists of numerical features transformed through PCA, highlighting the significance of dimensionality reduction and feature extraction in fraud detection. In contrast, the Credit Card Fraud Mega Dataset encompasses a wider range of features, from personal information to transaction specifics, illustrating the variety and depth of data utilized in comprehensive fraud detection systems. This diversity in feature types necessitates robust preprocessing and model training strategies to effectively harness each dataset's predictive power.

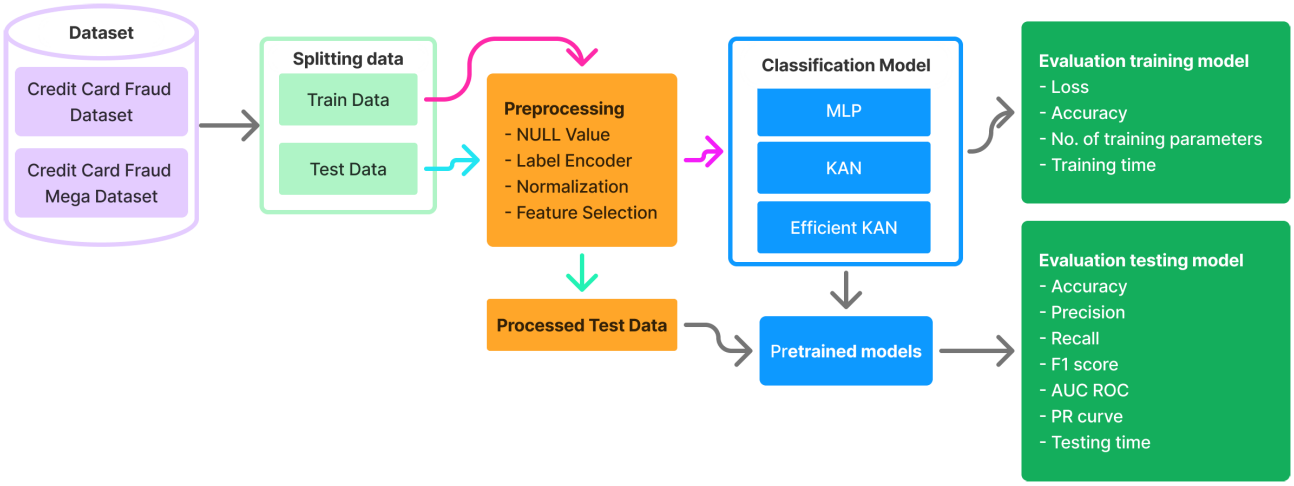


**FIGURE 1.** The proposed method for credit card fraud detection based on KAN models.

**TABLE 1.** Details of the datasets used in the experiments.

| Dataset | Input | Output |
|---|---|---|
| Credit Card Fraud | V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15,V16, V17, V18, V19, V20, V21, V22,V23, V24, V25, V26, V27, V28, Time, Amount | Class |
| Credit Card Fraud Mega | ssn, cc_num, first, last, gender, street, city, state, zip, lat, long, city_pop, job, dob, acct_num, profile, trans_num, trans_date, trans_time, unix_time, category, amt, merchant, merch_lat, merch_long | is_fraud |

### B. DATA PREPROCESSING

To prepare the datasets for modeling, we performed several preprocessing steps:

- **Data Cleaning**: This step involves identifying and removing duplicate or inconsistent entries to ensure data integrity. Let $\mathcal{D}$ be the dataset and $\mathcal{D}^*$ be the cleaned dataset. The process can be denoted as:

$$\mathcal{D}^* = \{x \in \mathcal{D} \mid \text{not duplicate or inconsistent}(x)\}$$

- **Handling Missing Values**: Missing data can be handled using various imputation techniques. Let $x_i$ represent a data point with missing values, and $\hat{x}_i$ be the imputed data point. One common approach is mean imputation for numerical features:

$$\hat{x}_i = \begin{cases} x_i & \text{if } x_i \text{ is not missing} \\ \frac{1}{N}\sum_{j=1}^{N} x_j & \text{if } x_i \text{ is missing} \end{cases}$$

  where $N$ is the total number of non-missing values in the feature.
- **Normalization**: Normalization ensures that the numerical features are on a similar scale, which is crucial for many machine learning algorithms. We applied the RobustScaler, which scales features using statistics robust to outliers. Let $x_i$ be a feature, then the normalized feature $x_i'$ is given by:

$$x_i' = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

  where $Q_1(x)$ and $Q_3(x)$ are the first and third quartiles of the feature $x$, and $Q_2(x)$ is the median.
- **Encoding Categorical Variables**: Categorical features must be converted into numerical values for model compatibility. We used LabelEncoder for this purpose. Let $c_i$ be a categorical feature with $k$ unique values, $\{c_1, c_2, \ldots, c_k\}$. LabelEncoder assigns an integer value $e_i \in \{0, 1, \ldots, k-1\}$ to each category:

$$e_i = f(c_i)$$

  , where $f$ is a mapping function that assigns unique integers to unique categories.
- **Feature Selection**: Feature selection is a crucial preprocessing task that can enhance the performance of classification models in credit card fraud detection [51]. In this work, we measured the feature importance scores of each input feature corresponding to the output feature on two datasets using the Mean Decrease in Impurity (MDI) method, as shown in Figure 2. For the first dataset, Credit Card Fraud, the top five most important features are `V17`, `V14`, `V12`, `V16`, and `V10`. For the second dataset, Credit Card Fraud Mega, the top five features of importance are `amt`, `unix_time`, `trans_date`, `trans_time`, and `category`.

These preprocessing steps ensure the datasets are suitable for effectively training and evaluating the models.

### C. CREDIT CARD FRAUD DETECTION

#### 1) MLP MODEL

MLP is a feedforward neural network characterized by multiple layers of neurons. In our implementation:

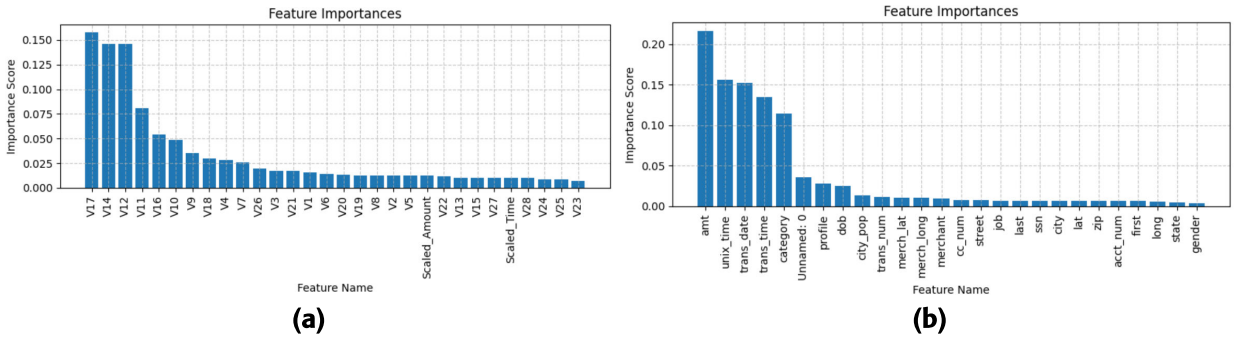


**FIGURE 2.** Feature importance measurement across two datasets. (a) Credit card fraud dataset; (b) Credit card fraud mega dataset.

- The MLP model is configured with $L$ layers, each layer $l$ consisting of $n_l$ neurons. ReLU activation functions are employed for hidden layers, while the Sigmoid function is used for the output layer.
- Training parameters are set as follows: batch size $b = 32$, learning rate $\eta = 0.001$, and the Adam optimization algorithm [52] is utilized.
- To assess model performance and mitigate overfitting, we employ 5-fold cross-validation.

The credit card fraud detection process based on MLP is detailed in Algorithm 1.

### 2) KOLMOGOROV-ARNOLD NETWORKS

(KAN) for Fraud Detection KAN leverages a structured approach that combines elements of neural networks with decision trees, enabling interpretable predictions. Key characteristics of our KAN implementation include:

- **Hierarchical Structure:** KAN employs a hierarchical architecture organized into multiple layers, each representing different levels of abstraction. This structure enhances interpretability by visualizing the decision-making process as a tree, where each node corresponds to a decision based on kernel activations.
- **KAN Parameters:** Several key parameters define the architecture of KAN: `width`: Number of neurons in each layer; `grid`: Grid configuration defining kernel activations; `k`: Order of the spline used in kernel activation functions; `device`: Computation device, typically set to 'gpu' for accelerated training.
- **Efficiency Enhancements:** Various adjustments were implemented to improve the efficiency of KAN, specifically tailored for credit card fraud detection: `Regularization`: Techniques such as L1 and L2 regularization were introduced to mitigate overfitting; `Grid Optimization`: Adaptive grid mechanisms dynamically adjust kernel activations based on input data distribution, optimizing model performance; `Batch Training`: Mini-batch training enhances generalization and convergence.

The detailed implementation of credit card fraud detection using KAN is presented in Algorithm 2.

**Algorithm 1** Applying MLP Model for Credit Card Fraud Detection

**Input**: Credit card dataset $\mathbf{X}$ with features and binary labels $\mathbf{y}$ (fraud or not)

**Output**: Evaluation metrics and performance indicators

Split $\mathbf{X}$ and $\mathbf{y}$ into training set $\mathbf{X}_{\text{train}}$, $\mathbf{y}_{\text{train}}$ and test set $\mathbf{X}_{\text{test}}$, $\mathbf{y}_{\text{test}}$
**Training Phase:**
**for** each epoch $e$ from 1 to max_epochs **do**
    Shuffle the training data
    **for** each mini-batch ($\mathbf{X}_{\text{batch}}$, $\mathbf{y}_{\text{batch}}$) in the training set **do**
        Perform forward pass to compute the output
        Compute loss using the loss function
        Perform backward pass to compute gradients
        Update model parameters using the Adam optimizer
    **end for**
**end for**
**Validation and Model Selection Phase:**
$\mathbf{y}_{\text{pred}} \leftarrow \text{predict}(\mathbf{X}_{\text{test}})$
Evaluate $\mathbf{y}_{\text{pred}}$ against $\mathbf{y}_{\text{test}}$ to compute metrics
**Compute Metrics:**
accuracy $\leftarrow \frac{TP+TN}{TP+TN+FP+FN}$
precision $\leftarrow \frac{TP}{TP+FP}$
recall $\leftarrow \frac{TP}{TP+FN}$
f1 $\leftarrow 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision}+\text{recall}}$
loss $\leftarrow \text{compute\_loss}(\mathbf{y}_{\text{test}}, \mathbf{y}_{\text{pred}})$
**Output Results:**
Print training time, testing time, number of parameters, accuracy, precision, recall, F1-score, and loss

### 3) EFFICIENT KAN

In addition to the standard Kolmogorov-Arnold Networks(KAN), we explored an efficient variant tailored to optimize computational resources and enhance model performance. Key aspects of this variant include:

- **Architecture Modifications:** We introduced several enhancements to streamline the KAN architecture,

**Algorithm 2** KAN for Credit Card Fraud Detection

**Procedure** `init_KAN`(width, grid, k, device='gpu'):
width ← width, grid ← grid, k ← k
base_fun ← base_fun, device ← device
act_fun ← [], symbolic_fun ← [], biases ← [], grid_eps ← 0.1
**End Procedure**
**Procedure** `train`(KAN, data, opt='LBFGS', steps=10):
optimizer ← initialize_optimizer(opt, KAN.parameters())
**for** step ← 0*to*steps **do**
    pred ← forward(KAN, data['train_input'].to(device))
    loss ← compute_loss(pred, data['train_label'].to(device))
    regularization ← compute_regularization()
    objective ← loss + lamb × regularization
    optimizer.zero_grad()
    objective.backward()
    optimizer.step()
**end for**
**End Procedure**
**Procedure** `predict`(KAN, input):
output ← forward(KAN, input)
**return** output {Binary prediction (fraud or not)}
**End Procedure**
**Procedure** `evaluate`(KAN, dataset):
total_correct ← 0
total_samples ← 0
**for** batch ∈ dataset **do**
    predictions ← predict(KAN, batch_inputs)
    Update evaluation metrics using predictions and batch labels
**end for**
Compute and return evaluation metrics (e.g., accuracy, precision, recall, F1 score)
**End Procedure**

including reduced layer sizes, optimized activation functions, and efficient grid implementations.

- **Comparative Analysis:** A thorough comparative analysis was conducted with the standard KAN model to evaluate improvements in computational efficiency, memory usage, and accuracy. The efficient KAN significantly enhanced speed and scalability while maintaining comparable accuracy levels.

This structured approach ensures consistent model evaluation and provides valuable insights into the strengths and limitations of each method within the context of credit card fraud detection. The detailed implementation of the efficient KAN for the credit card fraud detection model is presented in Algorithm 3.

## IV. EXPERIMENTAL SETUP

This section details the experimental setup used to evaluate the performance and interpretability of MLP, KAN, and Efficient KAN models in detecting credit card fraud. The experiments were conducted using specific hyperparameters tailored to optimize the performance of each model. We outline the parameter settings and the tuning strategies employed.

**Algorithm 3** Efficient KAN for Credit Card Fraud Detection

**Initialize KAN**
Initialize parameters: *input_size*, *hidden_sizes*, *output_size*, *grid_size*, *spline_order*, etc.
Initialize weights and biases for each layer
Create and register grid
Set activation function and scales: *base_activation*, *grid_eps*
**Forward Pass**
layer_input ← input
**for** hidden_size ∈ hidden_sizes **do**
    Compute *base_output* using *base_weight*
    Compute *spline_output* using B-spline bases and *scaled_spline_weight*
    layer_input ← base_output + spline_output
**end for**
output ← layer_input × output_weights + output_biases
**return** output
**Training Procedure**
optimizer ← initialize_optimizer(opt, self.parameters())
**for** epoch ∈ [1, epochs] **do**
    **for** batch ∈ dataset **do**
        predictions ← forward(batch_inputs)
        loss ← compute_loss(predictions, batch_labels)
        optimizer.step(loss)
    **end for**
**end for**
**Evaluation Procedure**
total_correct ← 0
total_samples ← 0
**for** batch ∈ dataset **do**
    predictions ← predict(batch_inputs)
    Update evaluation metrics using predictions and batch labels
**end for**
Compute and return evaluation metrics (e.g., accuracy, precision, recall, F1 score)

### A. MODEL HYPERPARAMETERS SETTING

The experimental setup and hyperparameter settings detailed here form the foundation for evaluating the performance of MLP, KAN, and Efficient KAN models in detecting credit card fraud. Each model was implemented with specific hyperparameters to optimize performance, as shown in Table 2.

The experimental setup and hyperparameter settings detailed here form the foundation for evaluating the performance of MLP, KAN, and Efficient KAN models in detecting credit card fraud. Each model was implemented with specific hyperparameters to optimize performance.

The hyperparameters for each model are chosen to optimize their performance on the given datasets. The MLP model, with its relatively simpler architecture, is set with standard parameters like learning rate and optimizer to ensure

**TABLE 2.** Hyperparameters setting for three models.

| Model | Hyperparameters | Values |
|---|---|---|
| MLP | Input Size | 30 (Dataset 1)<br>26 (Dataset 2) |
| | Hidden Size | 64 |
| | Output Size | 2 |
| | Learning Rate | 1e-3 |
| | Weight Decay | 1e-4 |
| | Optimizer | `AdamW` |
| | Loss Function | `CrossEntropyLoss` |
| | Epochs | 10 |
| KAN | Width | [30, 2] (Dataset 1)<br>[26, 2] (Dataset 2) |
| | Grid Size | 5 |
| | Order of Piecewise Polynomial | 3 |
| | Noise Scale | 0.1 |
| | Noise Scale Base | 0.1 |
| | Activation Function | `SiLU` |
| | Grid Epsilon | 1.0 |
| | Grid Range | [-1, 1] |
| | Optimizer | `LBFGS` |
| | Loss Function | `CrossEntropyLoss` |
| | Epochs | 10 |
| Efficient KAN | Input Features | 30 (Dataset 1)<br>26 (Dataset 2) |
| | Output Features | 2 |
| | Grid Size | 5 |
| | Spline Order | 3 |
| | Scale Noise | 0.1 |
| | Scale Base | 1.0 |
| | Scale Spline | 1.0 |
| | Enable Standalone Scale Spline | True |
| | Grid Epsilon | 0.02 |
| | Grid Range | [-1, 1] |
| | Activation Function | `SiLU` |
| | Loss Function | `CrossEntropyLoss` |
| | Epochs | 10 |

robust training. KAN, known for its interpretability, uses parameters like grid size and order of piecewise polynomials to manage its complexity and noise scales to enhance learning stability. Efficient KAN, designed for computational efficiency, adjusts spline orders and noise scales accordingly. In summary,

- **MLP**: Uses standard settings to ensure robust training. Input size varies by dataset, with a hidden size of 64, an output size of 2, and a learning rate of 1e-3. `AdamW` optimizer and `CrossEntropyLoss` are employed over 10 epochs.
- **KAN**: Employs parameters to balance complexity and interpretability. Uses a grid size of 5 and a polynomial order of 3, with noise scales set at 0.1. Optimized with `LBFGS` (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) and trained with `CrossEntropyLoss` over 10 epochs. The LBFGS algorithm uses limited memory, making it suitable for large-scale optimization problems where storing the full Hessian matrix is impractical.
- **Efficient KAN**: Focuses on computational efficiency with spline orders and noise scales adjusted accordingly. It uses a grid size of 5 and includes settings for scale parameters. Optimized for precision with `SiLU` activation and `CrossEntropyLoss` over 10 epochs.

### B. TUNING STRATEGIES

Parameter tuning plays a crucial role in optimizing model performance. The following strategies were employed:

- **Early Stopping:** training is halted when the validation performance ceases to improve to prevent overfitting. Let $\text{Val}_t$ be the validation metric at epoch $t$. Training stops if $\text{Val}_t > \text{Val}_{t+1}$ for a predetermined number of epochs (patience parameter). This ensures that the model does not overfit the training data while still being trained sufficiently.
- **Performance Metrics:** Models are evaluated based on the following metrics:
  **Accuracy:** The proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

  where: TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives
  **Precision:** The proportion of true positive results among the total number of positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

  **Recall:** The proportion of true positive results among the total number of actual positives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  **F1-score:** The harmonic mean of precision and recall, providing a balance between the two.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

  **AUC ROC**: A Area Under the Receiver Operating Characteristic Curve (AUC ROC) performance metric for binary classification models. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. True Positive Rate (TPR), also known as sensitivity or recall:

$$\text{TPR} = \frac{TP}{TP + FN}$$

  False Positive Rate (FPR):

$$\text{FPR} = \frac{FP}{FPP + TN}$$

  The AUC (Area Under the Curve) measures the entire two-dimensional area under the ROC curve, providing an aggregate performance measure across all classification thresholds. The AUC ranges from 0 to 1, with 1 representing a perfect model and 0.5 representing a model with no discrimination ability.

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR})$$

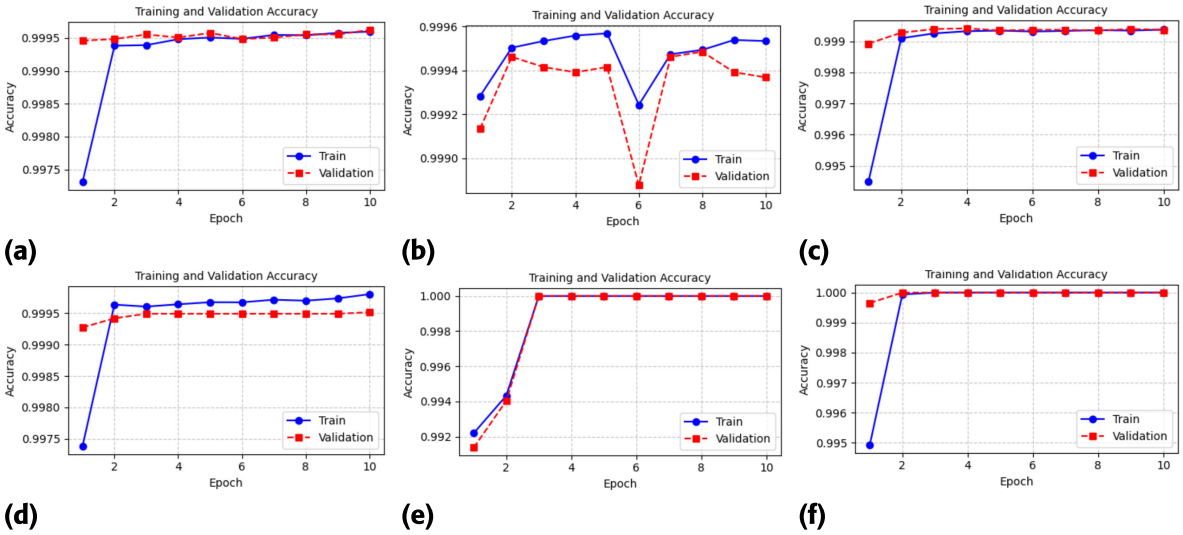


**FIGURE 3.** Training and validation accuracy of three models on two datasets: Credit card fraud dataset (ds1) and credit card fraud mega dataset (ds2). (a) MLP on ds1; (b) KAN on ds1; (c) Efficient KAN on ds1; (d) MLP on ds2; (e) KAN on ds2; (f) Efficient KAN on ds2.

, where $d(FPR)$ is a mathematical notation indicating that the AUC is the area under the curve created by plotting TPR against FPR as FPR changes incrementally from 0 to 1.

**Precision-Recall (PR) Curve**: The PR curve is constructed by varying the decision threshold and plotting Precision against Recall. The area under the Precision-Recall curve (AUC-PR) can be mathematically expressed as:

$$\text{AUC-PR} = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall})$$

These strategies ensure the models are tuned effectively to achieve the best possible performance in detecting credit card fraud.

## V. RESULTS

This section presents the experimental results and performance metrics obtained from evaluating MLP, KAN, and an efficient variant of KAN for credit card fraud detection. First, we present the accuracy and loss results of the training and validation phases. Then, we compare the performance of these models using evaluation metrics such as accuracy, precision, recall, and F1 score on testing data. Finally, we analyze the findings and insights gained by discussing the relationship between the training parameters, performance metrics (e.g., F1 score), and training and testing times.

### A. ACCURACY AND LOSS RESULTS ON TRAINING AND VALIDATION DATA

In this section, we present the results of the training and validation processes used to evaluate three models, including MLP, KAN, and Efficient KAN, across two datasets, using plot charts to illustrate the findings.

Figure 3 indicates that with the second dataset, Credit Card Fraud Mega, the KAN and Efficient KAN models exhibit better stability and performance than the MLP model during the training and validation phases. For the first dataset, Credit Card Fraud, while the Efficient KAN model maintains stability and performs competitively with the MLP model, the KAN model encounters a bottleneck at epoch 6 (as depicted in Figure 3(b)), resulting in a sudden decrease in performance.

Next, Figure 4 depicts the training and validation loss results for the three models. On the first dataset, the KAN model experiences a bottleneck at epoch 6 (as depicted in Figure 4(b)), where the loss suddenly increases. Meanwhile, the Efficient KAN model achieves the best performance, with stable loss results in both the training and testing phases. The MLP model shows promising loss performance during training, although its testing loss is higher than that of the Efficient KAN model. On the second dataset, both the KAN and Efficient KAN models demonstrate slightly better loss performance than the MLP model.

### B. COMPARISON OF PERFORMANCE MODELS ON TESTING DATA

First, we represent the performance of each model that was evaluated using established metrics for binary classification tasks, including accuracy, precision, recall, and F1-score. Figure 5 summarizes the metrics obtained for MLP, KAN, and Efficient KAN models on the two datasets. On the first dataset, all three models performed well, with KAN and Efficient KAN achieving slightly better precision than MLP. On the second dataset, KAN and Efficient KAN achieved 100% performance on all measurement metrics, slightly outperforming MLP.

Secondly, we provide an evaluation performance comparison of the three models on two datasets via AUC

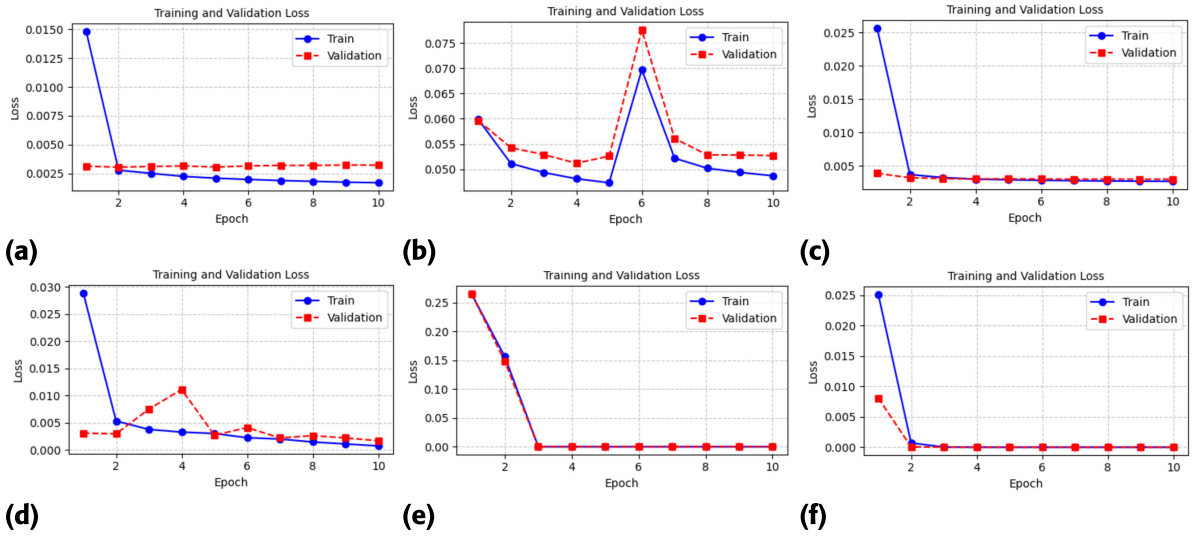


**FIGURE 4.** Training and validation loss of three models on two datasets: Credit card fraud dataset (ds1) and credit card fraud mega dataset (ds2). (a) MLP on ds1; (b) KAN on ds1; (c) Efficient KAN on ds1; (d) MLP on ds2; (e) KAN on ds2; (f) Efficient KAN on ds2.

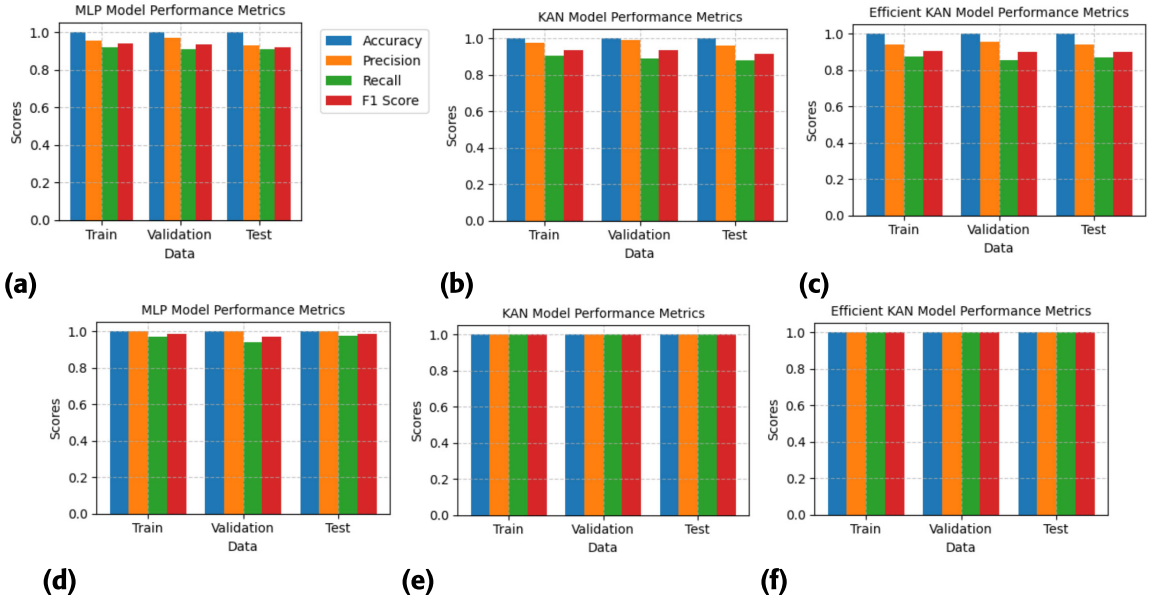


**FIGURE 5.** Performance accuracy of three models on two datasets: Credit card fraud dataset (ds1) and Credit Card Fraud Mega Dataset (ds2). (a) MLP on ds1; (b) KAN on ds1; (c) Efficient KAN on ds1; (d) MLP on ds2; (e) KAN on ds2; (f) Efficient KAN on ds2.

ROC in Figure 6 and the PR curve in Figure 7. Notably, all three models achieve a perfect ROC AUC of 100% on the credit card fraud mega dataset. Meanwhile, on the credit card fraud dataset, Efficient KAN obtains a ROC AUC score of 99%, KAN achieves 98%, and MLP scores 97%.

Regarding PR curve results, Efficient KAN achieves the best AP (Average Precision) score of 88%, while KAN and MLP obtain AP scores of 84% and 82%, respectively, on the Credit Card Fraud dataset. On the other hand, on the Credit Card Fraud Mega dataset, all three models achieve an identical AP score of 100%.

### C. ANALYSIS OF FINDINGS

This section delves into two primary relationships observed during the training and validation processes of the three model evaluations. The first relationship investigates the interaction between model training performance, the number of training parameters, and training time across two datasets. The second relationship focuses on model testing performance and testing time across the same datasets.

Figure 8 illustrates the first relationship. Despite the MLP model having the largest number of training parameters (over 2,000 for dataset 1 and 2,500 for dataset 2), its training time is comparable to that of the Efficient KAN model,

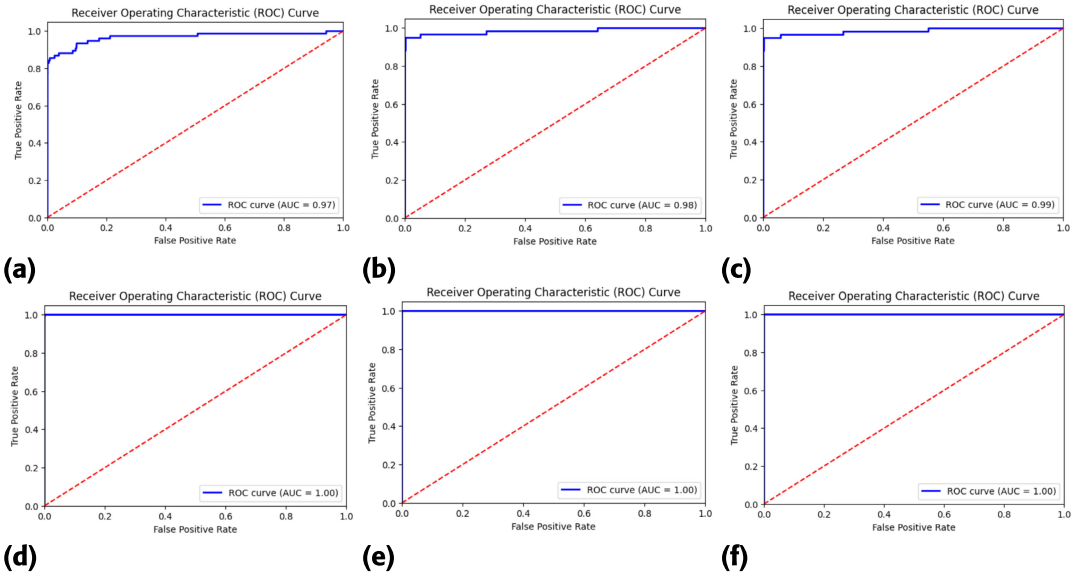


**FIGURE 6.** Performance AUC ROC of three models on two datasets: Credit card fraud dataset (ds1) and credit card fraud mega dataset (ds2). (a) MLP on ds1; (b) KAN on ds1; (c) Efficient KAN on ds1; (d) MLP on ds2; (e) KAN on ds2; (f) Efficient KAN on ds2.

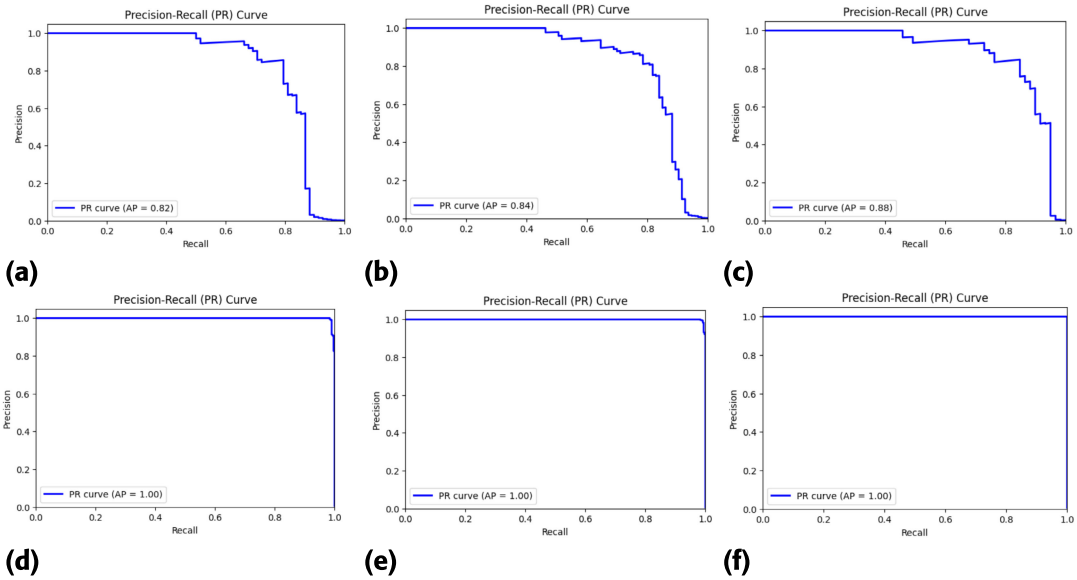


**FIGURE 7.** Performance PR curve of three models on two datasets: Credit card fraud dataset (ds1) and credit card fraud mega dataset (ds2). (a) MLP on ds1; (b) KAN on ds1; (c) Efficient KAN on ds1; (d) MLP on ds2; (e) KAN on ds2; (f) Efficient KAN on ds2.

taking 200 seconds for dataset 1 and even less for dataset 2. In contrast, despite having fewer parameters on both datasets, the KAN model exhibits the slowest training times, requiring 900 seconds for dataset 1 and over 280 seconds for dataset 2. However, the KAN model outperforms the MLP model regarding F1 score on both datasets.

Figure 9 illustrates the second relationship, focusing on testing performance and time across the two datasets. The Efficient KAN achieves the highest F1 score of 100% and the quickest detection time of 0.1 seconds. Meanwhile, the KAN model achieves the best F1 score on both datasets but has the slowest inference time compared to the other two models, taking 1 second longer on the first dataset. In contrast, the MLP model exhibits the slowest inference time on the second dataset, taking over 1 second, while the F1 scores of the KAN and Efficient KAN models slightly outperform the MLP model on the first dataset.

In summary, the key findings can be highlighted as follows:

**Training Phase Findings**:

- Model Performance vs. Training Parameters: Despite MLP having the highest number of training parameters, Efficient KAN shows comparable training times,

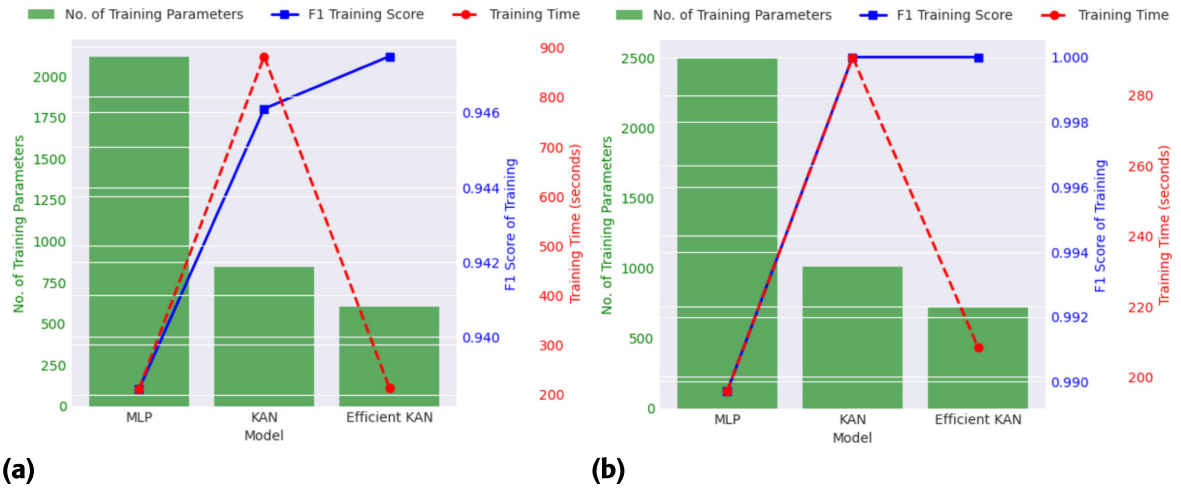


**FIGURE 8.** Relationship between F1 training score, number of training parameters, and training time across two datasets. (a) Credit card fraud dataset; (b) Credit card fraud mega dataset.

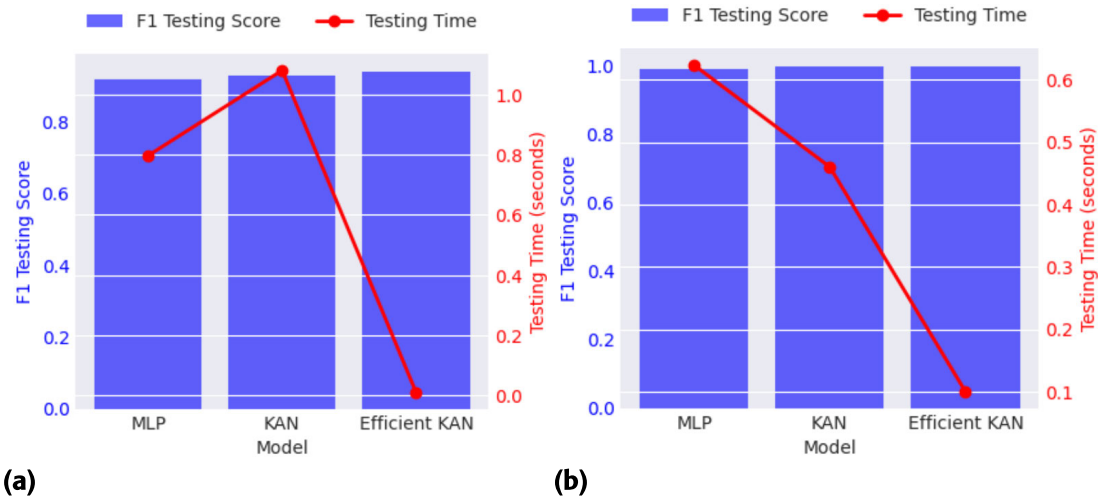


**FIGURE 9.** F1 testing score and testing time on two datasets. (a) Credit card fraud dataset; (b) Credit card fraud mega dataset.

indicating Efficient KAN's efficiency in parameter utilization.

- Training Time Discrepancy: KAN, despite having fewer parameters, exhibits significantly longer training times compared to MLP and Efficient KAN, suggesting potential inefficiencies in its training process.

**Testing Phase Findings**:

- Performance and Inference Time: Efficient KAN achieves the highest F1 score of 100% and the quickest detection time of 0.1 seconds, showcasing its effectiveness in both accuracy and speed.
- KAN's Performance: KAN achieves the best F1 score across both datasets but lags in inference time, particularly taking 1 second longer on the first dataset than Efficient KAN.

**General Model Comparison**:

- Performance Consistency: Across both training and testing phases, Efficient KAN consistently outperforms MLP and KAN in terms of both accuracy metrics and efficiency in inference times.
- Trade-offs: While KAN shows superior performance in F1 scores, it comes at the cost of longer training and inference times, highlighting a trade-off between model accuracy and computational efficiency.

**Dataset Comparison**:

- Dataset Impact: Differences observed between datasets (Credit Card Fraud vs. Credit Card Fraud Mega) suggest that model performance and efficiency metrics can vary significantly based on the dataset size and complexity.

## VI. DISCUSSION

### A. INTERPRETATION OF PERFORMANCE DIFFERENCES

The superior performance of KAN and its efficient variant over MLP can be attributed to several factors:

- **Model Complexity:** KAN utilizes a hierarchical structure that combines neural network capabilities

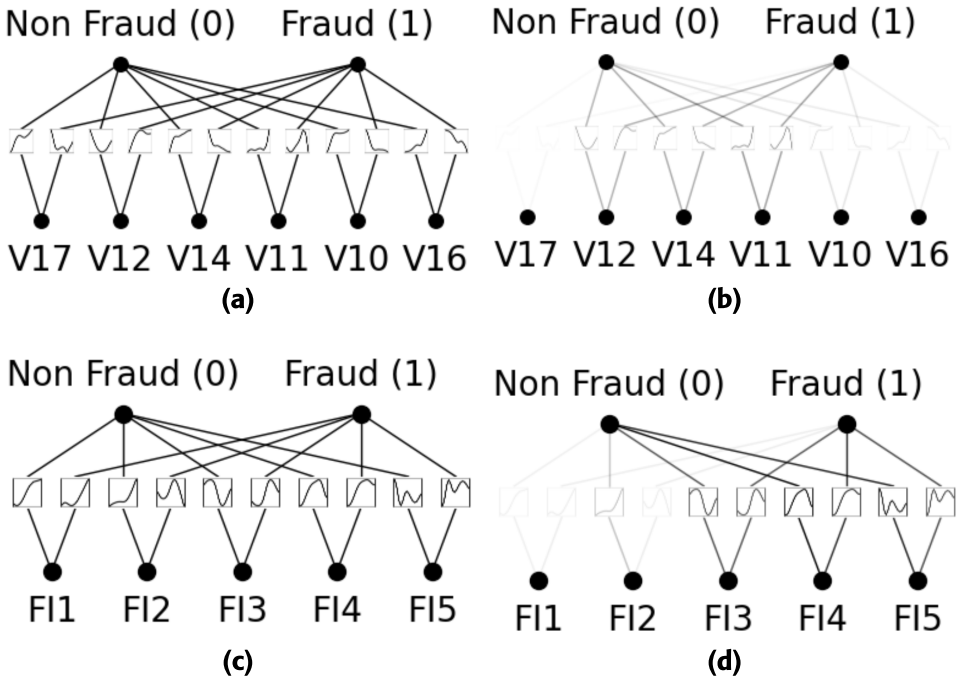


**FIGURE 10.** Interpretable KAN models for two datasets: Credit card fraud (DB1) in the first row and credit card fraud mega (DB2) in the second row. (a) With $\beta$ applied on DB1, (b) Without $\beta$ applied on DB1; (c) With $\beta$ applied on DB2, (d) Without $\beta$ applied on DB2. Note: `FI1` denotes `amt`, `FI2` denotes `trans_date`, `FI3` denotes `unix_time`, `FI4` denotes `trans_time`, `FI5` denotes `category`.

with decision-tree-like interpretability. This architecture enables KAN to capture complex patterns in credit card transactions more effectively than MLP, which may struggle with learning intricate relationships in high-dimensional data.

- **Interpretability:** Unlike MLP models, which operate as black boxes, KAN models provide transparent insights into the decision-making process. By breaking down predictions into interpretable components, KAN enhances stakeholders' understanding of fraud detection mechanisms, facilitating validation and trust in automated systems.

### B. INTERPRETABLE EFFICIENT KAN MODEL VISUALIZATION

Regarding model interpretability, we visualize the interpretability of the Efficient KAN model applied to the two datasets, as shown in Figure 10. For the first dataset, Credit Card Fraud, the top 5 most feature importance selected as inputs to our KAN models are `V17`, `V12`, `V14`, `V11`, and `V10` (see Figure 2a). For the second dataset, Credit Card Fraud Mega, the top 5 feature importance are `amt`, `trans_date`, `unix_time`, `trans_time`, and `category` (see Figure 2b).

In particular, we interpreted our KAN models with two options: with and without $\beta$. The parameter $\beta$ controls the transparency of activations. A larger $\beta$ value allows more activation functions to be visible (see Figure 10a for DS1 and Figure 10c for DS2). Typically, we prefer to set it without $\beta$ so that only important connections are visually significant (see Figure 10b for DS1 and Figure 10d for DS2), showcasing different activation functions. This clear representation of the KAN model helps us understand the model's structure, including the connections between input, hidden, and output layers via different activation functions. Consequently, the KAN model offers an interpretable transparency process rather than a learning model with a black-box approach.

### C. LIMITATIONS AND POTENTIAL FUTURE WORKS

This section addresses potential challenges, such as integration with existing systems and scalability, and proposes strategies for overcoming these challenges to ensure effective deployment.

- Enhanced Accuracy and Efficiency: Although KAN and Efficient KAN outperform MLP by reducing model complexity and computation time, achieving the best results on the Credit Card Fraud Mega dataset, there is still room for improvement on the Credit Card Fraud dataset. To address this, we suggest exploring hybrid sampling methods (such as SMOTE or Tomek Link) or fuzzy approaches to tackle class imbalance issues and integrating them with Efficient KAN for more effective handling.
- Model Decision Explanation: While the KAN model provides interpretability through its structure and decision-making process, further investigation into the detailed explainability of KAN's decisions on training and testing datasets is necessary to build greater

stakeholder trust. There is currently a lack of studies focusing on this aspect, and we aim to open a discussion and encourage broader attention to this issue among readers.

## VII. CONCLUSION

This study conducted a comparative analysis of Multilayer Perceptron (MLP), Kolmogorov-Arnold Networks (KAN), and an efficient variant of KAN for credit card fraud detection using two datasets. The findings consistently demonstrated that KAN and its efficient variant outperformed MLP across various performance metrics, including accuracy, precision, recall, F1-score, AUC-ROC, and PR curve. This advantage is primarily due to KAN's hierarchical structure, which effectively combines the predictive power of neural networks with the interpretability of decision-tree-like models.

The efficient variant of KAN was particularly notable for optimizing computational resources without compromising performance, making it an ideal solution for deploying fraud detection systems in resource-constrained environments. The study also emphasized the practical benefits of adopting KAN models, such as enhanced operational efficiency, regulatory compliance, and scalability for managing large volumes of transaction data.

For future work, research could focus on developing advanced variants of KAN to improve performance metrics further and enhance the interpretability of the KAN model's decisions. This might involve integrating sophisticated sampling techniques to address data imbalance better and conducting more in-depth investigations into the explainability of KAN's decision-making process, thereby increasing stakeholder trust in these models.

## REFERENCES


[1] *Nilson Report*. Accessed: Jun. 6, 2024. [Online]. Available: https://nilsonreport.com/articles/card-fraud-losses-worldwide-2/

[2] B. Sasikala and S. Sachan, "Decoding decision-making: Embracing explainable AI for trust and transparency," in *Exploring the Frontiers of Artificial Intelligence and Machine Learning Technologies*, vol. 42, A. U. Mirza and B. Kumar, Eds., 2024, ch. 3, pp. 42–63, doi: 10.59646/efaimltC3/133.

[3] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016.

[4] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov–Arnold networks," 2024, *arXiv:2404.19756*.

[5] *Efficient KAN Source Code*. Accessed: Jun. 20, 2024. [Online]. Available: https://github.com/Blealtan/efficient-kan

[6] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, "Kolmogorov–Arnold networks (KANs) for time series analysis," 2024, *arXiv:2405.08790*.

[7] B. C. Koenig, S. Kim, and S. Deng, "KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics," 2024, *arXiv:2407.04192*.

[8] Y. Wang, J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, and Y. Liu, "Kolmogorov Arnold informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov Arnold networks," 2024, *arXiv:2406.11045*.

[9] H. Morwani, "Innovative energy prediction using Kolmogorov–Arnold networks (KAN) and liquid neural networks (LNN) for smart grids," *Commun. Appl. Nonlinear Anal.*, vol. 31, no. 6s, pp. 74–80, Aug. 2024.

[10] K. Patel, "Credit card analytics: A review of fraud detection and risk assessment techniques," *Int. J. Comput. Trends Technol.*, vol. 71, no. 10, pp. 69–79, Oct. 2023.

[11] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[12] S. E. Kafhali and M. Tayebi, "Generative adversarial neural networks based oversampling technique for imbalanced credit card dataset," in *Proc. 6th SLAAI Int. Conf. Artif. Intell. (SLAAI-ICAI)*, Dec. 2022, pp. 1–5, doi: 10.1109/SLAAI-ICAI56923.2022.10002630.

[13] I. D. Mienye and N. Jere, "Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions," *IEEE Access*, vol. 12, pp. 96893–96910, 2024.

[14] A. D. Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, Aug. 2014.

[15] SamanehSorournejad, Z. Zojaji, R. E. Atani, and A. H. Monadjemi, "A survey of credit card fraud detection techniques: Data and technique oriented perspective," 2016, *arXiv:1611.06439*.

[16] Y. Kou, C. T. Lu, S. Sirwongwattana, and Y. P. Huang, "Survey of fraud detection techniques," in *Proc. IEEE Int. Conf. Netw.*, vol. 2, Mar. 2004, pp. 749–754.

[17] M. E. Edge and P. R. F. Sampaio, "The design of FFML: A rule-based policy modelling language for proactive fraud management in financial data streams," *Expert Syst. Appl.*, vol. 39, no. 11, pp. 9966–9985, Sep. 2012.

[18] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, Feb. 2011.

[19] O. S. Yee, S. Sagadevan, and N. H. A. H. Malim, "Credit card fraud detection using machine learning as data mining technique," *J. Telecommun., Electron. Comput. Eng.*, vol. 10, nos. 1–4, pp. 23–27, 2018.

[20] K. Vengatesan, A. Kumar, S. Yuvraj, V. D. A. Kumar, and S. S. Sabnis, "Credit card fraud detection using data analytics techniques," *Adv. Math., Sci. J.*, vol. 9, no. 3, pp. 1177–1188, Jun. 2020, doi: 10.37418/amsj.9.3.43.

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

[22] N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, p. 662, Feb. 2022, doi: 10.3390/electronics11040662.

[23] M. K. Mishra and R. Dash, "A comparative study of Chebyshev functional link artificial neural network, multi-layer perceptron and decision tree for credit card fraud detection," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2014, pp. 228–233.

[24] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," *Comput. Secur.*, vol. 57, pp. 47–66, Mar. 2016.

[25] F. Z. E. Hlouli, J. Riffi, M. A. Mahraz, A. E. Yahyaouy, and H. Tairi, "Credit card fraud detection based on multilayer perceptron and extreme learning machine architectures," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Jun. 2020, pp. 1–5.

[26] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A neural network ensemble with feature engineering for improved credit card fraud detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022.

[27] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Sep. 2019.

[28] P. J. G. Lisboa, S. Saralajew, A. Vellido, R. Fernández-Domenech, and T. Villmann, "The coming of age of interpretable and explainable machine learning models," *Neurocomputing*, vol. 535, pp. 25–39, May 2023.

[29] F. Anowar and S. Sadaoui, "Incremental neural-network learning for big fraud data," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 3551–3557.

[30] I. de Zarzà, J. de Curtò, and C. T. Calafate, "Optimizing neural networks for imbalanced data," *Electronics*, vol. 12, no. 12, p. 2674, Jun. 2023.

[31] F. Carcillo, Y.-A. Le Borgne, O. Caelen, and G. Bontempi, "Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization," *Int. J. Data Sci. Anal.*, vol. 5, no. 4, pp. 285–300, Jun. 2018.

[32] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, "Deep learning and explainable artificial intelligence techniques applied for detecting money laundering—A critical review," *IEEE Access*, vol. 9, pp. 82300–82317, 2021.
[33] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, "Credit card fraud detection in the era of disruptive technologies: A systematic review," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 1, pp. 145–174, Jan. 2023.
[34] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 159–166.
[35] M. Zareapoor and P. Shamsolmoali, "Application of credit card fraud detection: Based on bagging ensemble classifier," *Proc. Comput. Sci.*, vol. 48, pp. 679–685, Dec. 2015.
[36] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Expert Syst. Appl.*, vol. 100, pp. 234–245, Jun. 2018.
[37] Y. Alghofaili, A. Albattah, and M. A. Rassam, "A financial fraud detection model based on LSTM deep learning technique," *J. Appl. Secur. Res.*, vol. 15, no. 4, pp. 498–516, Oct. 2020.
[38] N. Purohit and R. G. Vishwakarma, "Filtering the credit card fraud detection dataset for enhancing the classification performance," *Math. Statistician Eng. Appl.*, vol. 71, no. 4, pp. 10122–10130, 2022.
[39] 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.
[40] J. Soni, P. Gangwani, S. Sirigineedi, S. Joshi, N. Prabakar, H. Upadhyay, and S. A. Kulkarni, "Deep learning approach for detection of fraudulent credit card transactions," in *Artificial Intelligence in Cyber Security: Theories and Applications*. Cham, Switzerland: Springer, 2023, pp. 125–138.
[41] Y. Tang and Y. Liang, "Credit card fraud detection based on federated graph learning," *Expert Syst. Appl.*, vol. 256, Dec. 2024, Art. no. 124979.
[42] M. Azim Mim, N. Majadi, and P. Mazumder, "A soft voting ensemble learning approach for credit card fraud detection," *Heliyon*, vol. 10, no. 3, Feb. 2024, Art. no. e25466.
[43] M. Tayebi and S. E. Kafhali, "Credit card fraud detection based on hyperparameters optimization using the differential evolution," *Int. J. Inf. Secur. Privacy*, vol. 16, no. 1, pp. 1–21, Nov. 2022, doi: 10.4018/ijisp.314156.
[44] S. E. Kafhali and M. Tayebi, "XGBoost based solutions for detecting fraudulent credit card transactions," in *Proc. Int. Conf. Adv. Creative Netw. Intell. Syst. (ICACNIS)*, Nov. 2022, pp. 1–6, doi: 10.1109/ICACNIS57039.2022.10054965.
[45] Z. Bozorgasl and H. Chen, "Wav-KAN: Wavelet Kolmogorov–Arnold networks," 2024, *arXiv:2405.12832*.
[46] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.
[47] D. Streeb, Y. Metz, U. Schlegel, B. Schneider, M. El-Assady, H. Neth, M. Chen, and D. A. Keim, "Task-based visual interactive modeling: Decision trees and rule-based classifiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 9, pp. 3307–3323, Sep. 2022.
[48] D. Sepiolo and A. Liga, "Towards explainability of tree-based ensemble models. A critical overview," in *Proc. Int. Conf. Dependability Complex Syst.* Cham, Switzerland: Springer, 2022, pp. 287–296.
[49] Kaggle. *Credit Card Fraud Dataset*. Accessed: Apr. 23, 2024. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
[50] Kaggle. *Credit Card Fraud Dataset*. Accessed: Apr. 23, 2024. [Online]. Available: https://www.kaggle.com/datasets/karthikgangula/credit-card-fraud-mega-dataset
[51] H. Zhu, M. Zhou, Y. Xie, and A. Albeshri, "A self-adapting and efficient dandelion algorithm and its application to feature selection for credit card fraud detection," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 2, pp. 377–390, Feb. 2024.
[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

**THI-THU-HUONG LE** received the bachelor's degree from the Hung Yen University of Technology and Education (HYUTE), Vietnam, in 2007, the master's degree from Hanoi University of Science and Technology (HUST), in 2013, and the Ph.D. degree from Pusan National University (PNU), South Korea, in 2020. She has three years of experience as a Postdoctoral Researcher with PNU, in 2020. She has seven years of experience as a Lecturer with HYUTE. She is currently a Research Professor with the Blockchain Platform Research Center, PNU. She has participated in machine learning projects, such as NILM, IDS, AI industry 4.0, AI security, and deep learning-based CFD. Her research interests include machine learning, deep learning, generative AI, data analysis, explainable AI, and signal processing. She has served several academic services, such as the guest editor, a technical committee, and a reviewer.

**YEONJEONG HWANG** received the B.S. degree in psychology from Kyungpook National University (KNU), South Korea. She is currently pursuing the master's degree with PNU. Her research interests include cybersecurity, cryptography, and AI applications.

**HYOEUN KANG** received the B.S. degree in applied IT and engineering and the M.S. and Ph.D. degrees in computer engineering from Pusan National University (PNU), South Korea. She is currently a Senior Researcher with SmartM2M. Her research interests include data analysis, natural language processing, machine learning, deep learning, and AI applications.

**HOWON KIM** (Member, IEEE) received the bachelor's degree from Kyungpook National University (KNU), and the Ph.D. degree from Pohang University of Science and Technology (POSTECH). He was a Visiting Postdoctoral Researcher with the Communication Security Group (COSY), Ruhr-University Bochum, Germany, from July 2003 to June 2004. He is currently a Professor with the Department of Computer Science and Engineering and the Chief of the Energy Internet of Things (IoT), IT Research Center (ITRC), and the Information Security Education Center (ISEC), Pusan National University (PNU). Before joining PNU, he was with the Electronics and Telecommunications Research Institute (ETRI), as the Team Leader for ten years beginning, in December 1998.