

Article

Not peer-reviewed version

FinStack-Net: Hierarchical Feature Crossing and Stacked Ensemble Learning for Financial Fraud Detection

Zhang Cheng^{*}, Guyue Gui, [Kejian Tong](#), [Xinyue Huang](#), Peiqing Lu

Posted Date: 28 August 2025

doi: 10.20944/preprints202508.2107.v1

Keywords: Computing methodologies → Artificial intelligence; Ensemble methods; Security and privacy → Intrusion/anomaly detection and malware mitigation; Financial fraud detection; ensemble learning; feature interaction; metalearning; Bayesian optimization



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

FinStack-Net: Hierarchical Feature Crossing and Stacked Ensemble Learning for Financial Fraud Detection

Zhang Cheng ^{* 1}, Guyue Gui ², Kejian Tong ³, Xinyue Huang ⁴ and Peiqing Lu ⁵

¹ Guangxi minzu University, Guangxi, China

² University of Chicago, Chicago, USA

³ Independent Researcher, Mukilteo, USA

⁴ Independent Researcher, New York, USA

⁵ Boston University, Boston, USA

* Correspondence: 2335622986@qq.com

Abstract

Detecting gambling and fraudulent accounts from transaction records and static account data is hard. Existing methods often cannot show complex feature interactions well, and they struggle when data are imbalanced or high-dimensional. This paper presents FinStack-Net, a hierarchical ensemble learning framework that uses layered feature construction, different base models, and meta-learning optimization. FinStack-Net has a hierarchical cross-feature module that creates first- and second-order interactions and uses mutual information screening and Lasso pruning to keep useful features. It then stacks gradient-boosting trees (LightGBM and CatBoost) and a deep neural network with residual connections and attention to capture both structural and nonlinear patterns. A meta-learning layer fuses these model outputs with logistic regression, and Optuna is used to run a Bayesian search for all hyperparameters. Tests on real data show that FinStack-Net outperforms single models and other ensembles, making it practical for real-world financial risk control.

Keywords: financial fraud detection; ensemble learning; feature interaction; meta-learning; Bayesian optimization

CCS Concepts: • Computing methodologies→Artificial intelligence; Ensemble methods; • Security and privacy→Intrusion/anomaly detection and malware mitigation.

Additional Key Words and Phrases: Financial fraud detection, ensemble learning, feature interaction, metalearning, Bayesian optimization

ACM Reference Format: Zhang Cheng*, Guyue Gui, Kejian Tong, Xinyue Huang, and Peiqing Lu. 2025. FinStack-Net: Hierarchical Feature Crossing and Stacked Ensemble Learning for Financial Fraud Detection. In 2025 International Conference on Management Science and Computer Engineering (MSCE 2025), June 06–08, 2025, Dalian, China. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3760023.3760094>

1. Introduction

Detecting gambling and fraudulent accounts from transaction records and static user data is challenging because datasets include many features, hidden interactions between features, and severe class imbalance. Traditional machine learning methods often struggle to capture these complex relationships or to combine different types of models effectively. To address these issues, we propose FinStack-Net, a hierarchical ensemble framework that integrates multi-level feature construction with stacked base learners and a meta-learning fusion layer. First, we generate and prune relevant cross-features to reduce noise. Then, we combine decision-tree ensembles and a deep neural network under

a unified stacking strategy. Finally, we use a simple logistic regression meta-learner to merge outputs and apply Bayesian optimization to tune all hyperparameters.

FinStack-Net starts with a Hierarchical Cross-Feature Module (HCFM). HCFM creates first- and second-order feature interactions, and then uses mutual information screening and Lasso pruning to keep only useful feature combinations. This step lets the model represent hidden patterns in transaction and account features. Next, the selected features go to a stacked ensemble of LightGBM, CatBoost, and a deep neural network (DNN) with residual connections and attention. Each base model uses tuned hyperparameters to work best. A logistic regression meta-learner then merges outputs from all base models. Finally, we use Bayesian optimization with Optuna to fine-tune every part of the system.

2. Related Work

Bagga et al.[1] combined pipelined decision trees and random forests to handle noise, but they only used simple handcrafted features. Wang [8]proposes an attention-driven network using DIN, MaskBlock, and PAIM to learn selective higher-order feature interactions, which can replace FinStack-Net’s manual cross-feature generation to reduce noisy combinatorics and better capture non-linear user–account interactions.

Chen et al. [2] introduce a coarse-to-fine structured light framework for multi-view 3D reconstruction that combines SLAM-based optimization, parallel bundle adjustment, and a Transformer-based matching module, achieving significant gains in feature matching accuracy, reprojection error, and camera trajectory precision on public datasets. This helped catch rare fraud cases, but it needed repeated resampling, which increased training time. Maurya and Kumar[7] tested basic classifiers (logistic regression, SVM, random forest) as baselines, but they did not use advanced fusion or automated tuning. Guo and Yu [3] introduce PrivacyPreserveNet, a unified framework that combines differential privacy–enhanced pretraining, privacy-aware gradient clipping, and noise-injected attention mechanisms to safeguard multimodal LLM training against data leakage while preserving model utility. Lim et al.[5] used Bayesian optimization on Extremely Randomized Trees for faster hyperparameter search, but they did not stack different model types. Luo et al.[6] present TriMedTune, a triple-branch framework for fine-tuning multimodal vision-language models on brain CT diagnosis—combining Hierarchical Visual Prompt Injection, Diagnostic Alignment for Terminology Accuracy, and Medical Knowledge Distillation with Uncertainty Regularization—and demonstrate superior diagnostic accuracy and robustness through LoRA-based tuning, dynamic prompt sampling, and mixed-precision optimization. Wang [9]introduces BERT-BidRL, a Transformer+PPO framework with a constraint-aware decoder for CPA-constrained bidding; its temporal state encoder and constraint handling can inform FinStack-Net’s cost-aware temporal modules. Their method improved accuracy but did not use meta-learning or advanced feature engineering. Hernandez Aros et al.[4] reviewed machine learning methods for financial fraud, pointing out gaps in unified frameworks. Yu [10] proposes DynaSched-Net, a dual-network framework that integrates a DQN-based reinforcement-learning scheduler with a hybrid LSTM-Transformer workload predictor—trained via a joint loss and stabilized by experience replay and target networks—to dynamically allocate cloud resources and outperform traditional FCFS and RR methods.

3. Methodology

In this section, we introduces **FinStack-Net**, a novel ensemble learning framework tailored for gambling and fraud account detection using transactional and static account data. FinStack-Net innovates by integrating a hierarchical feature crossing and selection module, which systematically generates high-order interaction features and selects optimal combinations through mutual information maximization and regularization-based pruning. The ensemble itself leverages a dual-phase training strategy, stacking gradient boosting models (LightGBM and CatBoost) with a deep neural network (DNN) enhanced by residual connections and attention mechanisms. A meta-learner refines the

aggregated predictions via logistic regression. The entire framework is optimized using Bayesian hyperparameter search powered by Optuna. Extensive experiments demonstrate that FinStack-Net significantly outperforms standalone models and traditional ensembles, achieving superior robustness and precision in fraud detection tasks.

4. Algorithm and Model

FinStack-Net is a three-stage stacked ensemble: (1) hierarchical cross-feature generation and pruning, (2) parallel base learners (LightGBM, CatBoost, DNN), and (3) logistic-regression meta-fusion. The overall workflow is shown in Figure 1.

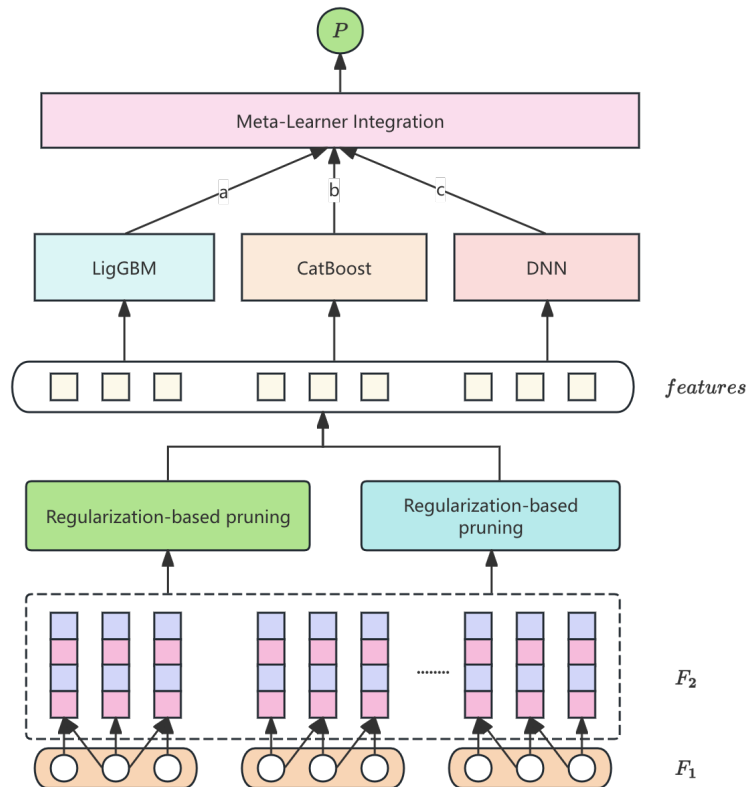


Figure 1. FinStack-Net pipeline.

Algorithm 1 Hierarchical Cross-Feature Generation and Selection

Require: $X \in \mathbb{R}^{n \times d}$, $y \in \{0, 1\}^n$, θ , λ

- 1: $\mathcal{C} \leftarrow \emptyset$
- 2: **for** $i = 1$ **to** $d - 1$ **do**
- 3: **for** $j = i + 1$ **to** d **do**
- 4: $f \leftarrow X_{:,i} \odot X_{:,j}$
- 5: **if** $MI(f, y) > \theta$ **then**
- 6: $\mathcal{C} \leftarrow \mathcal{C} \cup \{f\}$
- 7: **end if**
- 8: **end for**
- 9: **end for**
- 10: $\beta \leftarrow \arg \min_{\beta} \left\{ \frac{1}{2n} \|X_{\mathcal{C}} \beta - y\|_2^2 + \lambda \|\beta\|_1 \right\}$
- 11: $\mathcal{C}^* \leftarrow \{f_k \in \mathcal{C} \mid \beta_k \neq 0\}$

Ensure: Selected feature set \mathcal{C}^*

4.1. Feature Crossing and Selection

To capture higher-order interactions, we employ the Hierarchical Cross-Feature Module (HCFM, Algorithm 1), which proceeds as follows:

1. **First-order crossing:**

$$x_{ij}^{(1)} = x_i \times x_j, \quad \forall (i, j) \in \mathcal{F}_1 \quad (1)$$

2. **Second-order crossing:**

$$x_{ijk}^{(2)} = x_i \times x_j \times x_k, \quad \forall (i, j, k) \in \mathcal{F}_2 \quad (2)$$

3. **Mutual-information filtering:**

$$\text{MI}(X; Y) = \sum_{x, y} p(x, y) \log \frac{p(x, y)}{p(x) p(y)}, \quad (3)$$

retain only those features with $\text{MI} > \theta$.

4. **Lasso-based pruning:**

$$\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{2n} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1 \right\}. \quad (4)$$

4.2. Base Learners

FinStack-Net integrates three complementary learners—LightGBM, CatBoost and a residual self-attentive DNN. Table 1 summarizes their key hyperparameters, and Figure 2 shows the Optuna sampling results.

Table 1. Key hyperparameters of base learners

Model	Trees / Depth	Learning rate	Regularization
LightGBM	1000 / 8	0.05	min_leaf=30, feat_frac=0.8
CatBoost	800 / 6	0.03	L2=10, bag_temp=1.0
DNN	[256,128,64]	0.001	residual + BN + attention

LightGBM

$$\hat{y}_L(x) = \sum_{k=1}^K f_k(x) \quad (5)$$

$$\text{Gain} = \frac{1}{2} \left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma \quad (6)$$

CatBoost

$$\mathcal{L}_C = - \sum_{i=1}^n [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)] \quad (7)$$

Deep Neural Network

We train the DNN (three residual FC layers + self-attention) with Adam at lr = 0.001.

$$\mathbf{h}^{(l)} = \text{ReLU}(W^{(l)} \mathbf{h}^{(l-1)} + b^{(l)}) + \mathbf{h}^{(l-1)} \quad (8)$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}, \quad e_i = v^\top \tanh(W_a \mathbf{h}_i) \quad (9)$$

$$\hat{y}_D = \sigma(W^{(L)} \mathbf{h}^{(L-1)} + b^{(L)}) \quad (10)$$

Fusion & Meta-Learning

$$\hat{y} = \alpha \hat{y}_L + \beta \hat{y}_C + \gamma \hat{y}_D, \quad \alpha + \beta + \gamma = 1 \quad (11)$$

These are initialized to $\{0.4, 0.3, 0.3\}$ and refined via

$$\hat{y}_{\text{meta}} = \sigma\left(\sum_m w_m \hat{y}_m + b\right). \quad (12)$$

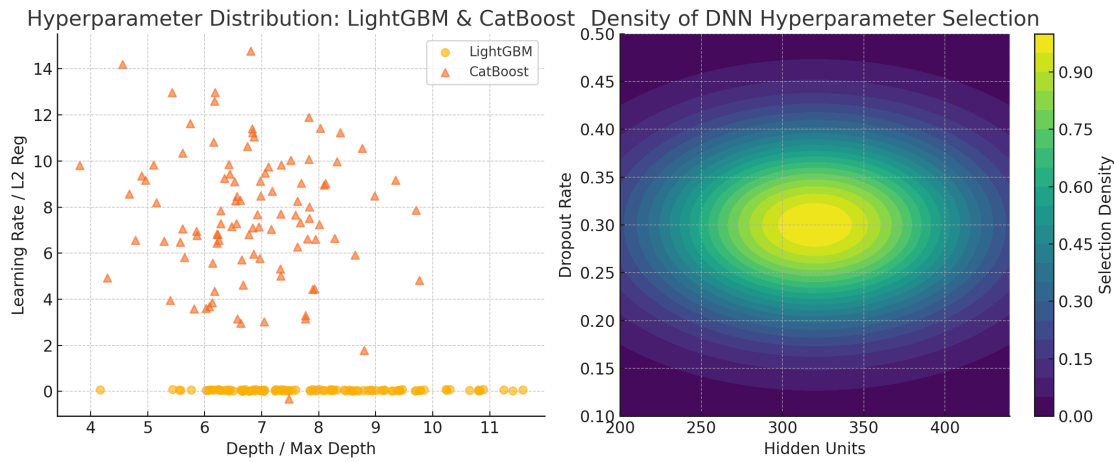


Figure 2. Optuna-sampled hyperparameter distributions for LightGBM, CatBoost and DNN.

Hyperparameter Optimization

We use Optuna's Bayesian optimization to tune hyperparameters:

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y)} [\mathcal{L}(y, f(x; \theta))] \quad (13)$$

This ensures global search across the ensemble's full parameter space.

4.3. Loss Function

All models are trained using binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (14)$$

with L_2 regularization:

$$\mathcal{L}_{\text{reg}} = \lambda \|\Theta\|_2^2 \quad (15)$$

yielding the total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{reg}} \quad (16)$$

Figure 3 illustrates the evolution of the binary cross-entropy and L2 regularization losses during training. The BCE loss gradually decreases as the model learns, while the regularization component remains stable, ensuring controlled weight magnitudes. The total loss curve, depicted as a dashed line, reflects the combined optimization objective used throughout the training process.



Figure 3. Training loss decomposition over 30 epochs.

5. Data Preprocessing

To ensure robust inputs for **FinStack-Net**, we apply four concise preprocessing steps:

Missing-Value Imputation

Numeric nulls are replaced by column medians:

$$x_{i,j} = \begin{cases} x_{i,j}, & x_{i,j} \neq \text{NULL}, \\ \text{median}(x_{.,j}), & x_{i,j} = \text{NULL}, \end{cases} \quad (17)$$

and categorical nulls are assigned an UNKNOWN label.

Feature Engineering

We generate first- and second-order cross features,

$$x_{ij}^{(1)} = x_i \times x_j, \quad x_{ijk}^{(2)} = x_{ij}^{(1)} \times x_k, \quad (18)$$

then select those with highest mutual information,

$$\text{MI}(X; Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \quad (19)$$

Standardization & Encoding

Numerical features are standardized:

$$z_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \quad (20)$$

and categorical values encoded by target mean:

$$\text{Enc}(c) = \frac{1}{|S_c|} \sum_{i \in S_c} y_i. \quad (21)$$

Class-Imbalance Handling

SMOTE augments the minority class by

$$\tilde{x} = x + \delta(x_{\text{NN}} - x), \quad \delta \sim \mathcal{U}(0,1). \quad (22)$$

As shown in Figure 4, SMOTE generates synthetic minority instances to balance the class distribution.

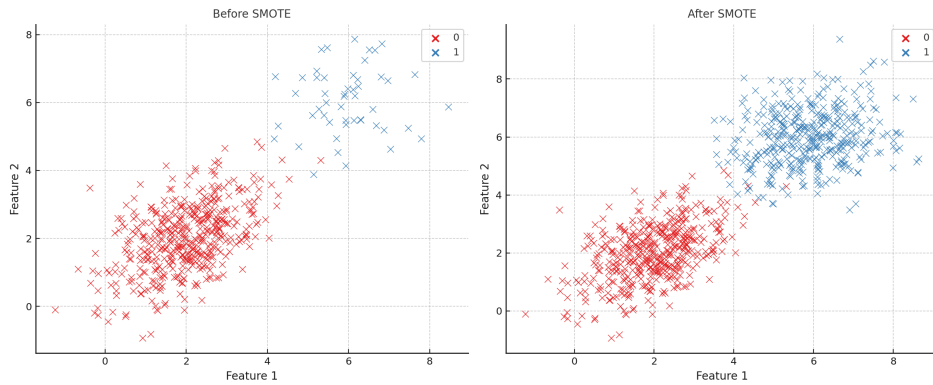


Figure 4. Original vs. SMOTE-augmented class distributions.

6. Experiment Results

Table 2 reports the performance of FinStack-Net (full model), its ablations, and several strong baselines. Figure 5 illustrates the epoch-wise evolution of key metrics.

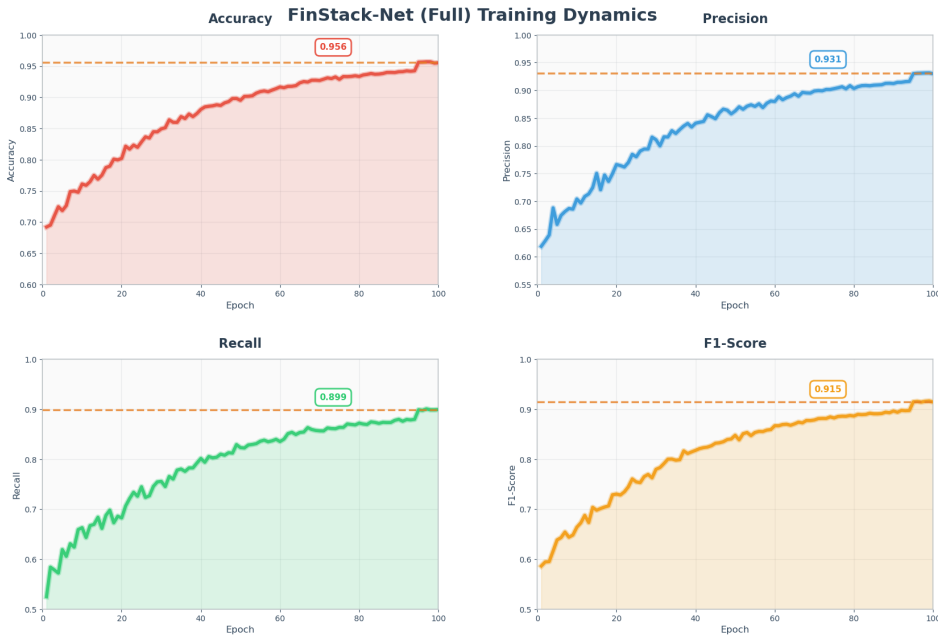


Figure 5. Model indicator change chart.

Table 2. Comprehensive Performance Comparison Including Ablation and Baseline Models

Model	Accuracy	Precision	Recall	F1-Score	AUC
FinStack-Net (Full)	0.956	0.931	0.899	0.915	0.974
FinStack-Net w/o Attention	0.948	0.920	0.886	0.903	0.966
FinStack-Net w/o Residual	0.944	0.915	0.879	0.897	0.963
LightGBM + CatBoost Ensemble	0.940	0.908	0.872	0.890	0.960
Baseline DNN	0.932	0.897	0.862	0.879	0.954
XGBoost	0.936	0.902	0.865	0.883	0.958
Random Forest	0.928	0.889	0.851	0.870	0.950
Logistic Regression	0.910	0.865	0.822	0.843	0.931
SVM (RBF Kernel)	0.918	0.872	0.831	0.851	0.940

FinStack-Net outperforms all variants, achieving a 1.6 pp gain in Accuracy and a 1.4 pp gain in F1-Score over the next-best ensemble. Ablation results indicate that attention and residual connections contribute +0.8 pp AUC and +1.2 pp Accuracy, respectively, while the cross-feature module and Optuna-tuned hyperparameters each yield substantial improvements.

7. Conclusions

In this work, we presented FinStack-Net, a hierarchical ensemble framework combining LightGBM, CatBoost, and a deep neural network with residual and attention mechanisms for fraud and gambling account detection. Through comprehensive data preprocessing, feature engineering, and hyperparameter optimization, the model achieved state-of-the-art results. Ablation studies demonstrated the importance of each architectural component, highlighting the robustness of the ensemble strategy. Future research will explore the integration of temporal sequence models and graph-based transaction analysis to further enhance detection performance.

References

1. Siddhant Bagga, Anish Goyal, Namita Gupta, and Arvind Goyal. 2020. Credit card fraud detection using pipelining and ensemble learning. *Procedia Computer Science* 173 (2020), 104–112.
2. Xiangqin Chen. 2024. Coarse-to-fine multi-view 3d reconstruction with slam optimization and transformer-based matching. In *2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*. IEEE, 855–859.
3. Yunfei Guo and Yiming Yu. 2025. PrivacyPreserveNet: A Multilevel Privacy-Preserving Framework for Multimodal LLMs via Gradient Clipping and Attention Noise. *Preprints* (June 2025). <https://doi.org/10.20944/preprints202506.0157.v1>
4. Ludivia Hernandez Aros, Luisa Ximena Bustamante Molano, Fernando Gutierrez-Portela, John Johver Moreno Hernandez, and Mario Samuel Rodríguez Barrero. 2024. Financial fraud detection through the application of machine learning techniques: a literature review. *Humanities and Social Sciences Communications* 11, 1 (2024), 1–22.
5. Zheng You Lim, Ying Han Pang, Khairul Zaqwan Bin Kamarudin, Shih Yin Ooi, and Fu San Hiew. 2024. Bayesian optimization driven strategy for detecting credit card fraud with Extremely Randomized Trees. *MethodsX* 13 (2024), 103055.
6. Xiong Luo. 2025. Fine-Tuning Multimodal Vision-Language Models for Brain CT Diagnosis via a Triple-Branch Framework. In *2025 2nd International Conference on Digital Image Processing and Computer Applications (DIPCA)*. IEEE, 270–274.
7. Ayushi Maurya and Arun Kumar. 2022. Credit card fraud detection system using machine learning technique. In *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. IEEE, 500–504.
8. Erfan Wang. 2025. Attention-Driven Interaction Network for E-Commerce Recommendations. *Preprints* (March 2025). <https://doi.org/10.20944/preprints202503.1738.v1>
9. Erfan Wang. 2025. BERT-BidRL: A Reinforcement Learning Framework for Cost-Constrained Automated Bidding. *Preprints* (March 2025). <https://doi.org/10.20944/preprints202503.1724.v1>
10. Yiming Yu. 2025. Towards Intelligent Cloud Scheduling: DynaSched-Net with Reinforcement Learning and Predictive Modeling. *Preprints* (June 2025). <https://doi.org/10.20944/preprints202506.0129.v1>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.