DAFPD: The Dynamic and Adaptive Framework for Enhanced Phishing Detection Techniques

Sudhir Kumar Gupta
Lakshmibai College, University of Delhi, Ashok Vihar Phase III, Delhi, India
sudhir@lb.du.ac.in

Sangeeta Srivastava
Bhaskaracharya College of Applied Science, University of Delhi, Sector 2, Dwarka, Delhi, India
sangeeta.srivastava@bcas.du.ac.in (corresponding author)

Vandana Gandotra
Ram Lal Anand College, University of Delhi, South Campus, Anand Niketan, Delhi, India
vandanagandotra.cs@rla.du.ac.in

Received: 18 June 2025 | Revised: 13 July 2025, 29 July 2025, 31 August 2025, 16 September 2025, 23 September 2025, and 11 October 2025 | Accepted: 13 October 2025

 DOI: https://doi.org/10.48084/etasr.12487

ABSTRACT

This study introduces the Dynamic and Adaptive Framework for Enhanced Phishing Detection (DAFPD) that uses multi-stage machine learning and deep learning for real-time, explainable phishing detection. DAFPD captures dynamic features (lexical, host-based, content-based, graph-based) and uses transformer models (BERT, RoBERTa) and Graph Neural Networks (GNNs) to accurately contextualize information. Python and Anaconda with PhishTank live dataset were used to perform these experiments. The detection pipeline consists of a light-weight heuristic filter, a deep learning phase with the combination of CNN-LSTM and transformers, and an ensemble learning aided with autoencoder-based zero-day attack anomaly detectors. For on-chain transactions, reinforcement learning (Deep Q-Networks) automatically determines thresholds and features. Explainable AI techniques, such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), provide justifications of the model predictions. Enabling cloud services with an "as a service" implementation, DAFPD communicates with Security Information and Event Management (SIEM) to block threats as they occur. The experimental results demonstrate that the DAFPD achieves significantly better performance.

Keywords-phishing detection; machine learning; Graph Neural Networks (GNNs); anomaly detection; reinforcement learning

I. INTRODUCTION

The expansion of the Internet has elevated the danger of phishing, particularly against government, financial, and online services. Through a phishing attack, identity theft is generally carried out using methods, such as spoofed URLs and bogus websites, to steal the information. Machine learning can be used to enhance the detection and user-awareness against phishing attacks [1-3], including SMS phishing. Authors in [4, 5] employed social engineering to manipulate victims or users into disclosing personal information or downloading malware using phishing. In the fourth quarter of 2024, APWG [6] observed 989,123 phishing attacks, with financial services being the main target category. However, existing defenses are reactive and can protect the user only against those threat models that are already in use by the developer [7]. DAFPD fills this gap by proposing a dynamic and adaptive architecture that evolves in response to new threats and provides interpretable detection outputs. DAFPD has certain limitations, though, including its dependence on training data, undiscovered added methods, detection of false positives and false negatives, computational complexity, and integration to real-world applications.

Phishing can be done in different ways, such as deceptive, spear, whale, and URL phishing [8-10]. To resolve these issues, this study presents a machine learning methodology for real-time phishing detection and physically tests this approach on a realistic dataset. The present study investigates phishing vectors, models a detection system, compares models based on experimental design, and tests these models in a real-world environment. It includes attack analysis, model creation,

feature engineering, and finally, incorporation of the proposed framework into security systems.

II. LITERATURE REVIEW

Several approaches have been proposed for mitigating phishing in both enterprise and public domains, primarily categorized either as user training or software-based classification [11-13]. Training aims to increase user awareness, while classification aims to automatically recognize phishing sites to minimize human mistakes.

The proposed detection method is inspired by the study of statistics of phishing URLs. Phishing is a serious threat that abuses users' financial circumstances, technological illiteracy, misplaced trust, etc. Authors in [14, 15] introduced a hierarchical machine learning phishing detection scheme, proposed features, applied logistic regression, and presented a capsule-based neural network for better URL detection. These studies introduced a machine learning framework for examining URL, Email, and website features to improve upon existing models.

According to [18], machine learning techniques, such as Random Forest (RF), could achieve good accuracy in phishing detection. Authors in [16] proposed a semantic feature and multi-scale-based phishing detection model. They developed 11 groups of features from web pages and generated models with AdaBoost, 3 Bagging, and SMO based on features, such as domain name, URL, and main picture, to design a model by utilizing the datasets of Direct Industry and Anti-Phishing Alliance in China. Feature selection was performed with WEKA, and SMO performed better compared to other models.

III. PROPOSED APPROACH

A. Phishing Detection

Beyond keyword filters, models such as BERT and RoBERTa examine emails and embedded links to identify phishing attempts. Phishing URLs are identified through domain structures with hybrid CNN-LSTM models, and GNNs, such as GCN and GAT, explain attack links connecting domains, IPs, and servers. In response to zero-day exploits, Deep Q-Network (DQNs) adjust in real time. SHAP and LIME, which are XAI tools, boost model trust and insights by revealing critical elements.

The DAFPD implements real-time patching and multi-stage machine learning to counter evolving phishing threats with dynamic feature extraction. DAFPD employs host-based lexical and content-based phishing features and applies BERT, CNN-LSTM, GNNs, and ensemble methods for robust detection, including zero-day attack phishing. DAFPD also integrates in real time with email gateways, browsers, and SIEM tools for streaming email supervision, browser examination, and security event monitoring. Fine-grained threshold setting is performed by reinforcement learning, while PSO/GA feature derivation and SHAP/LIME interpretability enhance them.

1) Algorithm 1 DAFPD Part 1: Feature Extraction and Classification

- Input: URLs, Email, SMS, website data, feature extraction components.
- Output: Intermediate phishing classification.

a) Step 1: Dynamic Feature Extraction

- Extract lexical, host-based, content-based, and graph-based features.
- Apply NLP-based Transformer models (BERT, RoBERTa) for phishing analysis.
- Construct domain relationship graphs using GNNs.

b) Step 2: Multi-Stage Machine Learning Pipeline

1. Stage 1: Fast Filtering (Lightweight Heuristic Model).

- Apply heuristic rules: Blacklisted domains, known phishing signatures, and UR patterns.
- Use Logistic Regression for quick phishing detection.
- If phishing is detected, flag and log the result; otherwise, proceed to Stage 2.

2. Stage 2: Adaptive Deep Learning Analysis

- Analyze URL sequences using CNN-LSTM
- Process email/web content using Transformer-based models (BERT, RoBERTa)
- Evaluate domain-level relationships with GNNs.

3. Stage 3: Ensemble Learning for Robust Classification

- Use Weighted Voting (RF, Deep Neural Networks).
- Implement Anomaly Detection (Autoencoder) for zero-day phishing attacks.
- Generate intermediate classification result.
- Return: Classification result for Part 1 = 0.

Phishing attacks are evolving, requiring dynamic and intelligent detection mechanisms. The proposed DAFPD employs multi-stage machine learning techniques for real-time phishing detection.

2) Algorithm 2 Multi-Stage Machine Learning Pipeline for Phishing Detection

- Input: URLs, Email, SMS, website data.
- Output: Phishing detection decision.

a) Stage 1: Fast Filtering (Lightweight Heuristic Model)

- Apply heuristic rules based on blacklisted domains, phishing, Signatures, and URL patterns.
- Use Logistic Regression for quick phishing detection.
- If phishing is detected, then flag and log the result

b) Stage 2: Adaptive Deep Learning Analysis

- Analyze URL sequences using CNN-LSTM models.

- Process email and web content using Transformer-based models (BERT, RoBERTa).
- Evaluate domain-level relationships using GNNs.

c) Stage 3: Ensemble Learning for Robust Classification

- Combine outputs from heuristic, deep learning, and GNN models.
- Finalize phishing detection decision = 0.

B. Feature Engineering

Features extracted from URLs, emails, and website content are represented as:

$$F = \{f_1, f_2 \ldots f_n\} \quad (1)$$

C. Stage 1: Lightweight Initial Screening

The Logistic Regression model is used for quick phishing classification:

$$P(y = 1|x) = \frac{1}{1+e^{-\omega^T x+b}} \quad (2)$$

where x represents the feature vector, w is the learned weights, b is the bias, and T is the Transport of matrix.

D. Stage 2: Deep Learning-Based Analysis

A Hybrid CNN-LSTM Model is employed to analyze sequential patterns:

$$h_i = \sigma(W_c \cdot x_i + b_c) \quad (3)$$

where W_c and b_c are the convolution kernel weights and bias, respectively, and x is the feature vector.

The LSTM network processes sequential features given by:

$$h_t = \sigma(W_h \cdot h_t - 1 + W_x \cdot x_t + b_h) \quad (4)$$

where h_t represents the hidden state, and W_h and W_x are the weight matrices.

Additionally, the GNN model domain relationship is:

$$h_v^{(l+1)} = \sigma + \left(W^{(l)} \sum_{u \in N(v)} h_u^{(l)} +_{b^{(l)}}\right) \quad (5)$$

where $h_v^{(l)}$ is the node feature at layer l, and $N(v)$ denotes its neighbors.

E. Stage 3: Ensemble Learning for Robust Detection

A weighted ensemble model was used:

$$y = \sum_{i=1}^{k} \alpha_i h_i (x) \quad (6)$$

where α_i represents the classifier weights.

Auto encoder-based anomaly detection helps detect zero-day phishing attacks:

$$L = \|x - \ddot{x}\|^2 \quad (7)$$

F. Stage 4: Real-Time Adaptability with Reinforcement Learning

A DQN adjusts detection thresholds dynamically:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max Q(s',a') - Q(s,a) \quad (8)$$

where s is the state, a is the action, r is the reward, γ is the discount factor, and α is the learning rate.

Figure 1 shows the multi-stage machine learning pipeline employed for phishing detection.

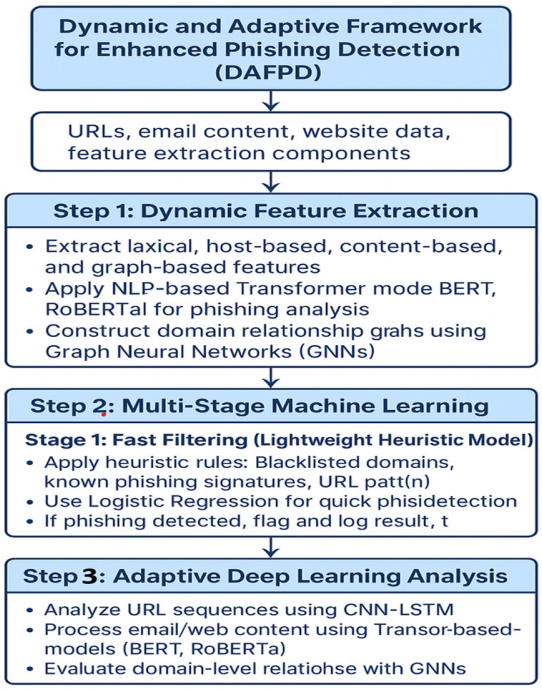

Fig. 1. Multi-stage machine learning pipeline.

G. SHapley Additive exPlanations

SHAP, a game theory approach, obtains contribution values to the final prediction for each feature. Through SHAP values, researchers can extract significant features, such as the RL length, the age of the domain, and the occurrence of some special characters in a URL or email, which help determine whether it is phishing.

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (9)$$

where ϕ_i is the SHAP value for feature i, N is the set of all features, S is a subset of features excluding i, and $f(S)$ represents the model's output with subset S.

H. Local Interpretable Model-Agnostic Explanations

LIME creates understandable approximations of a model's decision-making process by perturbing input samples and observing the impact on predictions. It builds a locally weighted linear model to explain individual predictions. Given an original input x, LIME perturbs it to create a set of samples x', assigns weights $w(x, x')$, and fits an interpretable model g such that:

$$g =$$

$$arg\ min \sum_{x' \in X} w(x, x')(f(x') - g(x'))2 + \Omega(g) \quad (10)$$

where $\Omega(g)$ is a complexity penalty to maintain interpretability.

By integrating these techniques, the phishing detection framework ensures that security analysts can understand, verify, and improve the model's decision-making, enhancing trust and reducing false positives.

1) Deployment in Dynamic Environments

To ensure effective real-time phishing detection, the model is deployed as a cloud-based API, enabling seamless integration with various security infrastructures. The API allows external applications, such as email gateways, web browsers, and cybersecurity tools, to send requests and receive phishing classification responses with minimal latency.

SIEM platforms aggregate and analyze security-related data from multiple sources, helping organizations find out and respond to intrusions. By integrating the phishing detection model with SIEM, organizations can:

- Automate phishing alerts: When a phishing attempt is detected, SIEM can trigger automated responses, such as blocking malicious URLs or quarantining emails.
- Enhance threat intelligence: The model continuously learns from new phishing patterns, updating the SIEM database with emerging threats.
- Facilitate forensic analysis: Security teams can trace attack origins and analyze past phishing attempts to strengthen defenses.

Mathematically, let $P(x)$ be the probability of a URL or email being phishing, and T be a dynamic threshold:

$$Action = \begin{cases} \text{Allow}, & P(x) < T \\ \text{Quarantine}, & T \leq P(x) < 1 - T \\ \text{Block}, & P(x) \geq 1 - T \end{cases} \quad (11)$$

where Allow means that the request is legitimate, Quarantine means that the email or URL is flagged for further inspection, and Block denotes immediate rejection of the phishing content.

IV. EXPERIMENTAL ANALYSIS

A. Dataset

In the first step, the Phishing Tank dataset [17], which originally contained 18 columns, was used. These data were pre-processed to produce 16 feature columns and one target column. The domain column was removed, as it should not be used in training. The features were examined using the data frame method, while a comparison of the data distribution and the relationship of the features were visualized. To prevent overfitting, the order was randomized, and the dataset was divided into test and training datasets, with equal distribution of data.

B. Data Collection

Phishing URLs were collected from the open-source Phish Tank platform, which also offers a real-time data feed of phishing sites that could be accessed in different file formats, including CSV and JSON. The dataset is refreshed every hour. The present work used 5,000 phishing URLs to train the proposed machine learning models, utilizing random sampling for a diverse and representative sample.

C. Data Sanitization and Pre-Processing

Missing values were added, noisy data were smoothed, outliers were removed, and anomalies were corrected to improve data quality for training and evaluation.

D. Feature Extraction

Phishing URLs exhibit unique traits compared to legitimate sites. Feature extraction was focused on encoding categorical variables, normalizing numerical features, and tokenizing text-based data, followed by the removal of redundant and irrelevant elements to enhance model efficiency and accuracy. The categories of features used in phishing detection are:

- Lexical Features: URL length, special character count, presence of obfuscation techniques.
- Hosting Attributes: Age of domain, WHOIS details, and SSL certificate information.
- Content-Based Features: Page content analysis, embedded scripts, and redirection.
- Graph-Based Features: Relationship mapping between domains to identify suspicious patterns.
- Natural Language Processing (NLP) Features: Use of Transformer-based models (e.g., BERT) to analyze phishing email and webpage content.
- Social Engineering Indicators: Analysis of email headers and domain relationships using GNNs.

E. Models and Training

A total of 10,000 samples were allocated into 8,000 training samples and 2,000 testing samples, using a supervised learning methodology. As phishing detection is a classification problem, the machine learning models used for training are:

- Decision Tree (DT): A simple, easy-to-interpret model that employs hierarchical decision rules.
- Multilayer Perceptron (MLP): A feedforward neural network for pattern recognition.
- RF: A DT ensemble to improve generalization
- Autoencoder Neural Network (ANN): Unseen phishing detection using perplexity-based features.
- XGBoost (XGB): A Gradient-boosted DT classifier that has high accuracy and efficiency
- Support Vector Machines (SVM): A powerful classifier that efficiently distinguishes between genuine and fraudulent URLs.
- Proposed DAFPD Approach: The proposed approach integrates advanced deep learning and ensemble learning techniques for enhanced phishing detection.

F. Performance Metrics

Table I compares the baseline models used in the experiments with the proposed DAFPD framework. Simpler and more interpretable, traditional models, such as TDs, RFs, and SVMs, still fall short in terms of adaptability and defenses against zero-day attacks. Likewise, ANNs and Autoencoders perform nonlinear learning and anomaly detection, but are standalone and lack robustness. Although XGB reaches high accuracy, it struggles too with evolving phishing strategies. In contrast, in Table II, DAFPD combines CNN-LSTM and transformers with GNNs, Autoencoders, ensemble learning, reinforcement learning, as well as SHAP/LIME explainability, and thus, achieves greater performance, flexibility, and interpretability than any other model.

TABLE I. CONFIGURATION DETAILS OF BASELINE MODELS USED IN EXPERIMENTS

Model	Model configuration details
RF	Criterion: Gini index; Maximum depth: 10; Splitting strategy: Best split; Minimum samples per leaf: 2
MLP/ANN	Architecture: 3 hidden layers (128, 64, 32 neurons); Activation: ReLU (hidden layers), Softmax (output); Optimizer: Adam; Learning Rate: 0.001; Epochs: 50; Batch size: 32
ANN	Encoder: 64 ⟶ 32 ⟶ 16 Neurons; Decoder: 16 ⟶ 32 ⟶ 64 Neurons; Latent dimension: 16; Activation: ReLU (hidden layers), Sigmoid (output); Training epochs: 50; Optimizer: Adam
XGB	Number of estimators: 300; Max depth: 8; Learning rate: 0.05; Subsample: 0.8; Regularization: $\lambda = 1$, $\alpha = 0$; Booster: gbtree
SVM	Kernel: Radial basis function; Regularization parameter (C): 1.0; Gamma: 0.01; Shrinking: Enabled

TABLE II. COMPONENTS OF THE PROPOSED DAFPD FRAMEWORK: PURPOSE, CONTRIBUTION, AND HYPERPARAMETERS

Component name and purpose	Contribution to framework	Hyperparameters used
CNN-LSTM sequential pattern recognition in URLs	Captures structural and temporal dependencies in URL sequences to detect obfuscation patterns.	Conv1D filters: 128; Kernel size: 3; LSTM units: 64; Dropout: 0.3; Optimizer: Adam; LR: 0.001; Epochs: 50; Batch size: 32
Transformers (BERT, RoBERTa) semantic and contextual analysis of emails/webpages	Understands language patterns and context, improving the detection of sophisticated phishing text.	Pre-trained BERT-base/RoBERTa-base; Max seq. length: 128 tokens; Batch size: 16; LR: 2e-5; Epochs: 3 (fine-tuning)
GNN (GCN, GAT) Domain–IP relationship modelling	Identifies hidden relationships between domains, servers, and IPs to uncover coordinated phishing campaigns.	Layers: 2; Hidden units: 64; Dropout: 0.2; LR: 0.001; Epochs: 100
Autoencoder anomaly detection for zero-day phishing	Learns normal feature patterns and flags deviations, enabling detection of unseen attacks.	Encoder: [64 ⟶ 32 ⟶ 16]; Decoder: [16 ⟶ 32 ⟶ 64]; Latent dim: 16; Activation: ReLU / Sigmoid; Optimizer: Adam; Epochs: 50
Ensemble learning (Weighted voting) Fusion of multiple classifiers	Combines the strengths of traditional machine learning and deep learning models, reducing misclassification risk.	Classifiers: CNN-LSTM, Transformer, GNN, Autoencoder, XGB; Voting: Weighted by validation accuracy
Reinforcement learning (DQN) Dynamic threshold adjustment	Continuously adapts detection thresholds in real time, enhancing resilience to evolving phishing tactics.	LR: 0.01; Discount factor γ: 0.9; Exploration rate ϵ: 1.0 ⟶ 0.1 decay; Episodes: 500
Explainable AI (SHAP, LIME) Model interpretability	Provides human-understandable justifications for predictions, increasing analyst trust and usability.	SHAP: Kernel explainer; LIME: 500 samples per explanation

TABLE III. PERFORMANCE METRICS AND THEIR FORMULAS

Metric	Formula
Accuracy (%)	$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100$
Precision (%)	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100$
Recall (%)	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100$
F1-score	$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
False Positive Rate (FPR)	$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$
False Negative Rate (FNR)	$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}}$
Detection time (ms)	$\text{Detection time} = \frac{\text{Total execution time}}{\text{Number of samples}}$
Zero-day attack detection rate (%)	$\text{Zero} - \text{day attack detection rate} = \frac{\text{Detected zero} - \text{day attacks}}{\text{Total zero} - \text{day attacks}} \times 100$

V. RESULTS AND DISCUSSION

A performance comparison of multiple machine learning models used for phishing detection is shown in Table IV. It also recorded the lowest detection time at 12.7 ms and demonstrated superior zero-day detection performance at 89.5% and explainability at 88.2%. The proposed DAFPD outperformed all other models, even under worst-case conditions

Figures 2-4 illustrate the implementation of CNN-LSTM, Transformers, GNNs, and DQNs with threshold tuning. They also demonstrate how explainability tools such as SHAP, LIME, and the DAFPD framework achieve high adaptability and explainability , resulting in improved phishing detection accuracy that surpasses traditional machine learning models

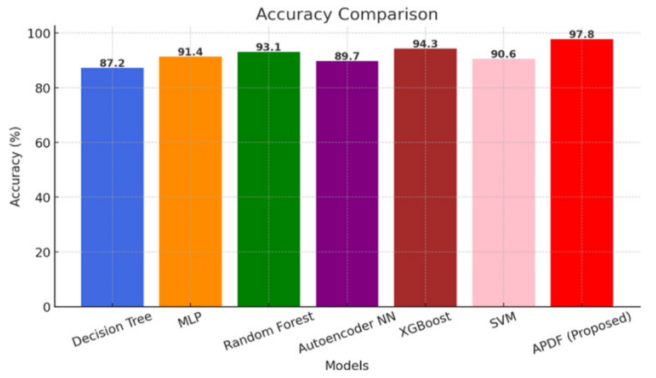

Fig. 2. Accuracy comparison.

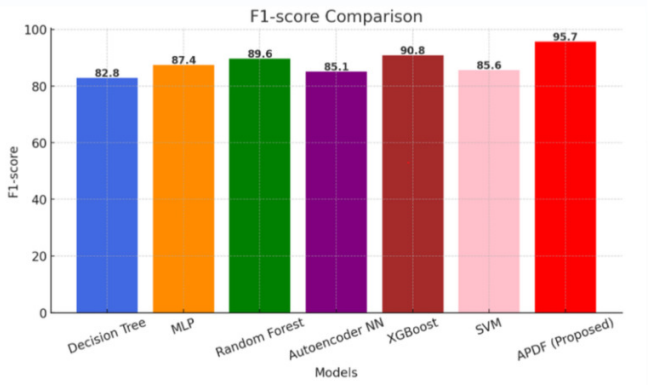

Fig. 3. F1-score comparison.

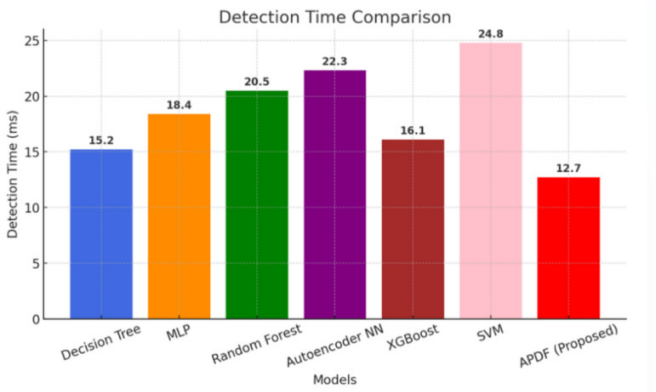

Fig. 4. Detection time comparison.

TABLE IV. MODEL PERFORMANCE COMPARISON

Model	Accuracy (%)	Precision (%)	Recall (%)	F1- score	Detection time (ms)	Zero-day attack detection rate %
DT	87.2	84.5	81.3	82.8	15.2	-
MLP	91.4	89.2	85.7	87.4	18.4	-
RF	93.1	90.8	88.5	89.6	20.5	-
ANN	89.7	86.4	83.9	85.1	22.3	-
XGB	94.3	91.9	89.7	90.8	16.1	-
SVM	90.6	87.3	84.1	85.6	24.8	-
DAFPD	97.8	96.5	94.9	95.7	12.7	89.5

VI. CONCLUSION AND FUTURE WORK

Phishing attacks persist despite existing defenses using awareness, machine learning, deep learning, and explainable AI, which often fail against new patterns. This study proposes the Dynamic and Adaptive Framework for Enhanced Phishing Detection (DAFPD), a multi-stage system integrating diverse features, heuristic filtering, contextual analysis, graph modeling, ensemble fusion, and anomaly detection.

DAFPD was evaluated on real-world phishing datasets and outperformed the conventional models. While achieving a 89.5% zero-day detection rate, DAFPD outperformed conventional models, such as Decision Trees (DT), Random Forests (RF), XGBoost (XGB), and Support Vector Machines (SVM), achieving 97.8% accuracy and a detection speed of 12.7 ms. The proposed framework of Shapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) guarantees decision transparency, a feature lacking in most high-performing deep learning systems. Unlike other works that focus on detection accuracy or interpretability, DAFPD provides both equitably alongside operational efficiency.

The DAFPD's novelty stems from its deployment-ready architecture, which connects directly to enterprise Security Information and Event Management (SIEM) systems, real-time adaptability, integration of multi-stage pipelines with heterogeneous features, and adaptability through reinforcement learning. DAFPD enables real-world applicability and making it a robust model in dynamic, uncontrolled environments, not just outperforming high-precision benchmarks. It signifies a practical, scalable, and interpretable phishing defense solution for ever-evolving digital infrastructures.

DATA AVAILABILITY STATEMENT

The experimental data used in this study are openly available in the GitHub repository at: https://github.com/ProfSudhir/experimental-dataset [17].

REFERENCES

[1] R. K. Ayeni, A. A. Adebiyi, J. O. Okesola, and E. Igbekele, "Phishing Attacks and Detection Techniques: A Systematic Review," in *2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals*, Omu-Aran, Nigeria, Apr. 2024, pp. 1–17, https://doi.org/10.1109/SEB4SDG60871.2024.10630203.

[2] M. Schmitt and I. Flechais, "Digital Deception: Generative Artificial Intelligence in Social Engineering and Phishing," *Artificial Intelligence Review*, vol. 57, no. 12, Oct. 2024, Art. no. 324, https://doi.org/10.1007/s10462-024-10973-2.

[3] M. Nanda, M. Saraswat, and P. K. Sharma, "Enhancing Cybersecurity: A Review and Comparative Analysis of Convolutional Neural Network Approaches for Detecting URL-based Phishing Attacks," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 8, June 2024, Art. no. 100533, https://doi.org/10.1016/j.prime.2024.100533.

[4] A. Darem, "Anti-Phishing Awareness Delivery Methods," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7944–7949, Dec. 2021, https://doi.org/10.48084/etasr.4600.

[5] S. K. Gupta, S. Srivastava, and V. Gandotra, "SMISHBAN: Framework for Detecting SMSHING SMS and Phishing Email Using Machine Learning Algorithms," in *Innovative Computing and Communications*, vol. 1435, A. E. Hassanien, S. Anand, A. Jaiswal, and P. Kumar, Eds. Singapore: Springer Nature Singapore, 2025, pp. 289–301.

[6] "Phishing Activity Trends Reports," *Anti-Phishing Working Group*, Mar. 2025. https://apwg.org/trendsreports/.

[7] A. K. Jain and B. B. Gupta, "A Novel Approach to Protect Against Phishing Attacks at Client Side Using Auto-updated White-list," *EURASIP Journal on Information Security*, vol. 2016, no. 1, Dec. 2016, Art. no. 9, https://doi.org/10.1186/s13635-016-0034-3.

[8] A. A. Akinyelu and A. O. Adewumi, "Classification of Phishing Email Using Random Forest Machine Learning Technique," *Journal of Applied Mathematics*, vol. 2014, pp. 1–6, 2014, https://doi.org/10.1155/2014/425731.

[9] M. Karthick Kumar and N. Sivakumar, "URL Phishing Attack Detection using Machine Learning Algorithms," in *2024 OPJU International Technology Conference on Smart Computing for Innovation and Advancement in Industry 4.0*, Raigarh, India, June 2024, pp. 1–8, https://doi.org/10.1109/OTCON60325.2024.10687804.

[10] M. A. Uddin and I. H. Sarker, "An Explainable Transformer-Based Model for Phishing Email Detection: A Large Language Model Approach." SSRN PrePrint, 2024, https://doi.org/10.2139/ssrn.4785953.

[11] G. Karat, J. M. Kannimoola, N. Nair, A. Vazhayil, V. G. Sujadevi, and P. Poornachandran, "CNN-LSTM Hybrid Model for Enhanced Malware Analysis and Detection," *Procedia Computer Science*, vol. 233, pp. 492–503, 2024, https://doi.org/10.1016/j.procs.2024.03.239.

[12] S. Ratra, M. Ghosh, N. Baliyan, J. Rashmitha Mohan, and S. Singh, "Graph Neural Network Based Phishing Account Detection in Ethereum," *The Computer Journal*, vol. 67, no. 12, pp. 3160–3168, Dec. 2024, https://doi.org/10.1093/comjnl/bxae079.

[13] S. Fan, H. Xu, S. Fu, Y. Luo, and M. Xu, "Edge-feature Modeling-based Topological Graph Neural Networks for Phishing Scams Detection on Ethereum," in *2024 IEEE/ACM 32nd International Symposium on Quality of Service*, Guangzhou, China, June 2024, pp. 1–10, https://doi.org/10.1109/IWQoS61813.2024.10682857.

[14] H. Kamal, S. Gautam, D. Mehrotra, and M. S. Sharif, "Reinforcement Learning Model for Detecting Phishing Websites," in *Cybersecurity and Artificial Intelligence*, H. Jahankhani, G. Bowen, M. S. Sharif, and O. Hussien, Eds. Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 309–326.

[15] S. K. Birthriya, P. Ahlawat, and A. K. Jain, "Phishing URL Detection using Deep Q-Networks with Convolutional Neural Networks," in *2024 International Conference on Intelligent Systems for Cybersecurity*, Gurugram, India, May 2024, pp. 1–6, https://doi.org/10.1109/ISCS61804.2024.10581203.

[16] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024, https://doi.org/10.1109/ACCESS.2024.3368377.

[17] S. K. Gupta, "Experimental Dataset." GitHub, Nov. 2025, [Online]. Available: https://github.com/ProfSudhir/experimental-dataset.

[18] A. A. Albishri and M. M. Dessouky, "A Comparative Analysis of Machine Learning Techniques for URL Phishing Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18495–18501, Dec. 2024, https://doi.org/10.48084/etasr.8920.