INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 5 • September 2026

PROACTIVE PHISHING DETECTION USING DEEP NEURAL NETWORKS: A MULTI-LAYER PERCEPTRON APPROACH FOR MALICIOUS URL CLASSIFICATION

Sanan Abdul Hameed[*1], Muhammad Abdul Rehman[2], Ayesha Sarfraz[3]

[*1,2,3]*Department of Computer Science and Information Technology, Superior University Lahore;*

[*1]sananabdulhameed@gmail.com, [2] m.abdulrehman5882@gmail.com, [3]ayeshsaarfraz@gmail.com

DOI: https://doi.org/10.5281/zenodo.23180522

Keywords
Phishing Detection, Malicious URLs, Deep Neural Networks, Multi-Layer Perceptron, Cybersecurity, Machine Learning

Article History
Received: 01 June 2026
Accepted: 23 August 2026
Published: 30 September 2026

Copyright @Author
Corresponding Author: *
Sanan Abdul Hameed

Abstract
Phishing is still one of the most common entry points for cybercrime and is now responsible for a significant portion of the data breaches reported by organizations annually. This paper introduces a deep learning technique for phishing website detection before they extract user's credentials. The UCI Phishing Websites dataset was used for training a Multi-Layer Perceptron (MLP) with 30 structural attributes, attributes related to the address bar and domain attributes for 11,055 labeled URLs. The data was scaled using min-max method and divided into 8,844 training and 2,211 test samples. The network consists of two hidden layers: the first 64 rectified linear units (RLU) with a dropout rate of 0.2, and the second 32 rectified linear units (RLU) with a dropout rate of 0.2, followed by a sigmoid output neuron which is optimized with Adam and trained with binary cross-entropy loss over 30 epochs. The held-out test data was used to test the model, which achieved an accuracy of 95.5 percent, precision of 96 percent, recall of 95 percent, and an F1-score of 95.5 percent, while training data showed an accuracy of 98.1 percent and a recall of 97 percent, demonstrating a small and controlled gap of generalization. Recall was a point of focus for this reason: a false negative is more expensive in a security scenario than a false positive in a security scenario, that is, a phishing site falsely identified as safe. Verizon's 2025 Data Breach Investigations Report indicated that 16 percent of breaches continue to start with phishing, while IBM's 2025 Cost of a Data Breach Report revealed that it takes on average 241 days to detect and contain a data breach. A model that identifies malicious URLs at contact points, instead of after they've caused damage directly addresses that gap in detection. The results validate the fact that a relatively simple MLP can compete with or even exceed the performance of much more complex architectures recently reported in the literature but be still sufficiently fast for real-time use at the client.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 5 • September 2026

INTRODUCTION

Phishing is no longer a plain old, misspelled email. It has become the opening move in some of the most damaging attacks organizations face, including Business Email Compromise (BEC) and ransomware. In 2025, the Anti-Phishing Working Group (APWG) reported approximately 3.8 million phishing attacks, up from 3.76 million in 2024, with social media and SaaS or webmail platforms making up about 20 percent of the targeted attacks, and the number of attacks in the telecom sector ramping up significantly in the latter half of the year (APWG, 2025). Another 22 percent of data breaches were driven by credential abuse, which phishing often leads to, and 16 percent began with phishing, according to Verizon's 2025 Data Breach Investigations Report (DBIR) (Verizon, 2025). Once a breach happens, the cost and time to recover are considerable: IBM's 2025 Cost of a Data Breach Report estimates that the average breach cost is $4.44 million USD, the lowest amount since the report's inception and an average of 241 days to identify and contain a breach, which is still over a year of exposure that goes unnoticed. (IBM Security, 2025).

Attacks can be easily generated using widely available language models and automated content generation systems to produce emails, landing pages and even cloned login pages that look and sound like the legitimate ones and they can be created in huge numbers, which a manual process cannot keep up with. In these circumstances, the tried-and-true type of defenses, static blacklists, manual takedowns and easy rule-based filtering are always one step behind. A blacklist is only as effective as the URLs it has been reported (Hamid et al., 2025), and new phishing sites are often up and running, collecting credentials, many times before being added to the list (Danish et al., 2025). The types of attacks have also changed to feature channels and tactics that were not covered by static filtering. The number of BEC attacks using wire transfers increased by 136 percent in the quarter alone, with the median dollar amount requested by a fraudster in a wire transfer attack rising from $20,000 to just over $50,000 (APWG, 2025). What is clear from both trends is that the attackers are not only sending more and more phishing e-mails, they are also spreading their attacks over several communication channels at the same time email, SMS, social networks and thus the one-key fit for a black list becomes increasingly ineffective (Ahmad et al., 2025), even while the underlying structural characteristics of a malicious site (the indicators on which this study relies) are largely unchanged. The research community has followed that spread. Atlam and Oluwatimilehin (2023) reviewed machine learning approaches to Business Email Compromise specifically, and both Thakur, Ali, Obaidat and Kamruzzaman (2023) and Kyaw, Gutierrez and Ghobakhlou (2024) surveyed deep learning for phishing email detection, which indicates how much of the recent effort has moved toward channel-specific detection. The present study stays with the website channel, where the structural signal is strongest.

It is this reactive posture that is the crux of the issue in this paper. A proactive approach tests the structure of a website, such as URL length and content, domain age and registration information, existence of a valid SSL certificate, and the behavior of HTML and JavaScript, to determine the probability that it is malicious (Delshadi et al., 2025) when it is first seen. These attributes act as a form of structural fingerprint which it's hard to completely mask, yet not remove, from a phishing page without losing many of the elements that make a successful page.

This problem has been addressed by machine learning for nearly 10 years and classical algorithms, such as Random Forest, Decision Trees and Support Vector Machines, have been successfully applied and well documented on benchmark datasets with satisfactory performance. As these features get more intricate, however, or the distinction between valid and bogus sites gets less distinctly divided, these models tend to reach a point of saturation, which is becoming more common as attackers systematically build in counterfeit structural features (Delshadi et al., 2025). The nonlinear activation function and the layered structure of deep learning models, and in particular, Multi-Layer Perceptrons, are more

suitable for this context, as they can learn combinations of features that cannot be represented by simpler models (Hamid et al., 2023). This study is about training and testing an MLP with the UCI Phishing Websites dataset, which is a popular set of 11,055 URLs with 30 engineered features. This paper has three contributions. Firstly, it uses a compact two-hidden-layer MLP architecture that operates on the full 30-feature space, reporting both training and held-out test performance, instead of just reporting training accuracy, thereby showing the generalization gap rather than hiding it. Second, it defines operational interest as recall, rather than accuracy, because a missed phishing site (a false negative) is obviously significantly more costly in a security context (Hamid et al., 2023) than a legitimate site that happens to be flagged for a short period of review (a false positive). Third, it places the resulting model in the context of a series of confirmed, recently published studies (from 2023 to 2026) (Khaliq et al., 2024) and thus enables the performance of the model to be read in context and not alone.

The rest of the paper is organized as follows. Section II presents a literature review (Ibrar et al., 2024) covering classical and ensemble baselines, deep learning and hybrid-optimization approaches, and the more recent work on explainability and adversarial robustness, drawing throughout on literature published between 2023 and 2026. The dataset, preprocessing pipeline and network architecture are detailed and described in Section III. In Section IV the training dynamics, test-set results, error analysis and a comparison against recent published work are presented. At the end of the paper is Section V, which summarizes the paper and suggests new avenues for future study, such as explainability and real-time browser deployment.

II. Literature Review

A. The UCI Feature Set as a Continuing Benchmark

Most published work on phishing website classification still rests on the same 30-attribute representation distributed through the UCI Machine Learning Repository, in which signals taken from the URL string, abnormal request patterns, HTML and JavaScript behaviour, and domain registration history are grouped into four interpretable categories. Recent surveys confirm that this feature set remains the usual starting point. Safi and Singh (2023) reviewed the phishing detection literature systematically and found that URL-based and hybrid feature sets of this kind dominate published work, in large part because they can be extracted without depending on third-party services such as WHOIS lookups or search-engine rankings. Wilk-Jakubowski, Pawlik, Wilk-Jakubowski and Sikora (2025), reviewing machine learning and neural network approaches published between 2017 and 2024, reached a similar conclusion and noted how tightly the reported accuracy figures cluster at the top of the range. Kytidou, Tsikriki, Drosatos and Rantos (2025) add that blacklists and heuristic rules are still the operational baseline these models are measured against, even though those baselines are known to lag behind live campaigns. That representation and that baseline are the ones the present study works with.

B. Classical and Ensemble Baselines

Tree-based and voting ensembles remain the strongest non-deep baselines on this kind of data, and they are still being published rather than treated as settled history. Karim, Shahroz, Mustofa, Belhaouari and Joga (2023) combined logistic regression, a support vector classifier and a decision tree into a hybrid voting model, evaluated on a URL feature set drawn from more than 11,000 websites, and reported 95.23 percent accuracy with an F-measure of 95.77 percent, using canopy feature selection and grid search for tuning. Alnemari and Alshammari (2023) applied classical learners to domain-level features and obtained comparable results without inspecting page content at all. Guptta, Shahriar, Alqahtani, Alsalman and Sarker (2024) took the hybrid-feature route instead, combining URL and content attributes before classification, and showed that how the features are constructed often matters more than which classifier consumes them.

Two limitations show up repeatedly in this line of work. These models give little insight into why a particular URL was flagged, and their accuracy leans on heuristics that an attacker can study and design around. Both points motivate the shift described next.

C. The Shift Toward Deep Learning (2023 to 2026)

In the last three years there has been a definite shift to deep architectures, partly because of the increasing size and variety of the datasets and partly because phishing pages are now complex enough that a shallow decision boundary is no longer sufficient. Alshingiti, Alaqel, Al-Muhtadi, Haq, Saleem and Faheem (2023) compared CNN, LSTM and a combined LSTM-CNN model and found the convolutional variants better at picking up local character-level patterns than a purely recurrent model. Su and Su (2023) moved the problem into the language-model space by fine-tuning BERT directly on URL strings, removing the manual feature engineering step altogether, and Do, Selamat, Fujita and Krejcar (2024) built on that idea with an integrated model that pairs deep learning classifiers with a pre-trained transformer. Prasad and Chandra (2024) contributed from a different angle: they released the PhiUSIIL dataset and challenged the assumption that a model has to be retrained from scratch whenever the threat landscape moves, arguing for incremental learning instead. Ghalechyan, Israyelyan, Arakelyan, Hovhannisyan and Davtyan (2024) took the most practical approach of the group, testing deterministic and probabilistic neural network models on six months of real production traffic supplied by EasyDMARC, a harder test than a single static benchmark split, with the probabilistic model reaching 97 percent accuracy under production conditions.

There has also been a great deal of experimentation in architecture. Kibriya et al. (2025) used large language model embeddings, testing BERT, GPT-2, T5-Large and TinyLLaMA, and paired them with a lightweight LSTM and GRU classifier, finding that BERT embeddings with their deep learning classifier reached 97.5 percent accuracy with roughly half a million parameters and about 0.12 milliseconds of classification time per URL. Türk and Kılıçaslan (2025) compared classical learners, deep models and metaheuristic hyperparameter optimization using genetic algorithms, particle swarm optimization and the Harris Hawk optimizer, and reported that an ELECTRA-based model reached accuracy and F1-scores of up to 99 percent, ahead of the optimization-assisted hybrids. Zhou et al. (2025) combined a parallel convolutional structure with a BiLSTM to capture both local and sequential patterns in the URL string, while Rao, Kondaiah, Pais and Lee (2025) assembled a hybrid super learner ensemble aimed specifically at mobile devices, where the computational budget is tight. Elhoseny, Abdel-Salam, Khafaga, Aldakheel and El-Hasnony (2025) tuned a deep learning phishing classifier for a semantic web system using a boosted triangular topology optimizer, which illustrates how much of the recent accuracy improvement in this area comes from better-tuned training rather than genuinely new network designs. Efficiency has become a design goal in its own right: Wang, Ma, Xu, Liu and Yin (2023) proposed a lightweight multi-view transformer with a mixture of experts in order to keep inference cheap, while Opara, Chen and Wei (2024) argued for reading raw URL and HTML characteristics together rather than relying on hand-built features alone.

Two further studies continue the discussion on architecture. Gobinath and Manikandan (2026) proposed an adaptive deep URL intelligence network, ADUIN, which combines lexical, host-based and structural URL features through hybrid feature optimization and an attention-based classifier, and reported roughly 95 percent classification accuracy along with a low false-positive rate, a strong zero-day detection rate and inference latency in the low hundreds of milliseconds under load. What is notable there is not the headline accuracy but the fact that zero-day generalization and deployment latency were treated as evaluation metrics rather than afterthoughts. Alsowail (2025) used a self-attention enhanced capsule network that models spatial relationships between URL characteristics instead of treating

them as an unordered feature vector, and achieved 99.2 percent detection accuracy, the highest among the studies reviewed here, though with a much heavier and far less interpretable model.

D. Explainability, Transferability and Robustness

Sitting alongside the question of accuracy is the question of whether these models can be trusted. Mia, Derakhshan and Pritom (2024) tested this directly by training models on two different public phishing datasets and using explainable AI to compare which features actually drove the predictions in each; the ranked feature importances did not transfer cleanly between datasets, which raises a real question about how far any reported accuracy figure travels beyond the corpus it was measured on. Calzarossa, Giudici and Zieni (2024) applied explainable machine learning to phishing feature detection and showed that a small subset of features usually carries most of the decision weight. Shafin (2025) went a step further and used SHAP and LIME as a feature selection mechanism rather than only as a post-hoc explanation, improving interpretability and detection performance at the same time. The practical argument running through all three is the same: a flagged URL should arrive with a human-readable reason attached, not only a probability score.

Robustness is the other half of the same concern. Kulkarni, Balachandran, Divakaran and Das (2025) generated adversarial phishing pages by embedding phishing features into otherwise legitimate pages and observed a significant drop in detection rates across three established detectors, with a multimodal large language model detector holding up better but still proving evadable. Any classifier built on a fixed, published feature set, including the one presented in this paper, inherits that weakness, and it is the reason adversarial testing is listed as future work in Section V rather than claimed as a result here.

E. Positioning of the Present Study

Putting these studies side by side, two trends are evident. First, the reported numbers from 2023 to 2026 sit very close together, roughly 95 to 99 percent across the studies, regardless of technique, whether that is a hybrid voting ensemble, a transformer-embedding classifier or a capsule network. That pattern suggests the feature representation, rather than the architecture alone, is doing much of the work. Second, although the security literature agrees that the cost of a missed phishing site exceeds the cost of a false alarm, very few of the studies reviewed report training and test performance together, and fewer still treat recall as a headline result. The present study addresses both gaps directly: it reports the generalization gap between training and test performance instead of a single number, and it treats recall, alongside accuracy, precision and F1-score, as a first-class result rather than a secondary one.

III. Methodology

A. Dataset

This study uses the Phishing Websites dataset distributed through the UCI Machine Learning Repository, which remains the standard benchmark in this literature (Safi & Singh, 2023). The dataset consists of 11,055 instances of either a phishing or a legitimate website, described by 30 categorical and numerical features. These features fall into four groups, summarized in Table 1.

Table 1. UCI Phishing Websites dataset: feature groups

Feature group	Example attributes	What it captures
Address bar based	URL length, use of an IP address instead of a domain name, presence of the "@" symbol, use of URL shortening services	Surface-level anomalies visible directly in the URL string

Feature group	Example attributes	What it captures
Abnormal based	Request URL ratio, anchor URL ratio, server form handler behaviour	Mismatches between what a page claims to be and how it actually behaves
HTML and JavaScript based	Use of pop-up windows, invisible iframes, right-click disabling, status bar customization	Client-side tricks commonly used to obscure a page's true destination or prevent inspection
Domain based	Domain age, DNS record presence, website traffic ranking, SSL certificate validity	Longer-horizon signals that are difficult for a newly registered phishing domain to fake

This four-way grouping is one of the reasons the dataset has remained the standard benchmark in this field for close to a decade: it allows a model, and a reader, to reason about which category of evidence is driving a classification, rather than treating the input as an opaque 30-dimensional vector.

B. Preprocessing

All 30 features were rescaled using min-max normalization, mapping every value into the [0, 1] range. This step matters more than it might appear. Several features in this dataset, website traffic ranking in particular, naturally take on a much wider numerical range than binary or near-binary features such as SSL certificate validity. Left unscaled, gradient-based optimizers like Adam would implicitly let the wider-range features dominate early weight updates simply (Malik et al., 2024) because of their magnitude, not because they are more informative. Min-max scaling puts every feature on equal numerical footing before training begins, which lets the network's weights, rather than incidental differences in feature scale, determine each feature's actual influence on the prediction.

C. Train and Test Split

The dataset was divided into an 80/20 split, yielding 8,844 samples for training and 2,211 samples held out for testing. The test set was not used in any way during training or hyperparameter selection; it was reserved specifically to measure how the model performs on data it has never encountered, which is the only meaningful test of whether a phishing detector will generalize to URLs it was not trained on. Figure 1 summarizes this pipeline end to end, from raw dataset to final evaluation.

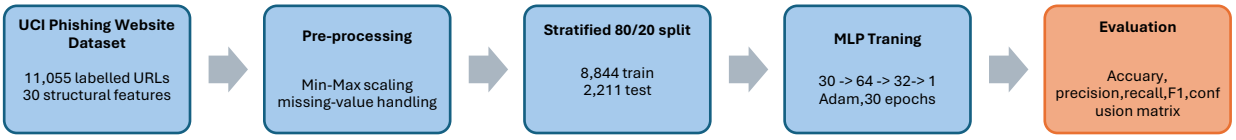


Figure 1. End-to-end methodology pipeline

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 5 • September 2026

D. Model Architecture

The classifier is a Multi-Layer Perceptron (MLP) built with Keras API in TensorFlow 2.x. Table 2 provides details of the architecture layer-by-layer, and figure 2 provides the corresponding network diagram.

Table 2. MLP architecture

Layer	Configuration	Purpose
Input	30 neurons	Matches the dimensionality of the feature set; no transformation is applied beyond the min-max scaling described above
Hidden layer 1	64 neurons, dense, ReLU activation, 0.2 dropout	Learns first-order combinations of the 30 input features
Hidden layer 2	32 neurons, dense, ReLU activation, 0.2 dropout	Learns higher-order, more abstract combinations of the hidden-layer-1 representation
Output	1 neuron, sigmoid activation	Produces a probability in [0, 1] representing the likelihood that the input URL is phishing

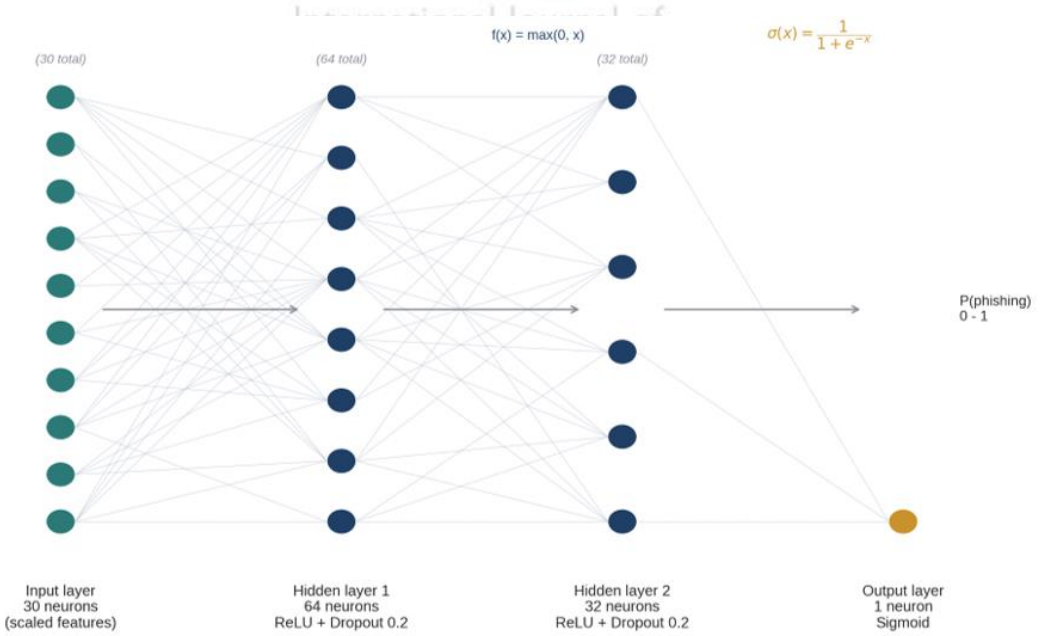


Figure 2. Multi-Layer Perceptron architecture used for URL classification

Both the hidden layers employed a Rectified Linear Unit (f(x) = max(0, x) activation. The decision to use ReLU over sigmoid or tanh activations in the hidden layers was made because ReLU does not become "saturated" for positive inputs, which helps in avoiding the vanishing-gradient problem that can slow or even stall training in deeper networks. The output layer has a sigmoid activation, with the form of sigmoid(x) = 1 / (1 + e^(-x)), which will force the final results of the network into a clean probability space from 0 to 1; a result above 0.5 is considered phishing, and a result of 0.5 or less is considered legitimate.

The dropout rate was set at 0.2 after every hidden layer, which, at every step of training, deactivated 20 percent of the neurons in each hidden layer.

Thus, it pushes the network to spread what it has learned throughout its many neurons and not be too reliant on any one of them, being the main form of regularization for the model that prevents overfitting.

E. Training Configuration

The model was trained for 30 epochs using the Adam optimizer (Adaptive Moment Estimation), which combines the benefits of AdaGrad's per-parameter learning rates with RMSProp's use of a moving average of squared gradients, at its standard configuration (learning rate 0.001, beta_1 = 0.9, beta_2 = 0.999). Binary cross-entropy was used as the loss function, which is the standard choice for a single-output, two-class classification problem of this kind. A held-out validation split drawn from the training data was monitored throughout training to track generalization behavior without touching the final test set.

F. Evaluation Metrics

The four standard metrics for evaluating the trained model were: accuracy (the percentage of all predictions that were successful); precision (the percentage of the URLs predicted as phishing which were phishing); recall (also known as sensitivity or the true positive rate, the percentage of all the URLs that are phishing but which were not misclassified as non-phishing); and F1-score, which is the harmonic mean of precision and recall. Recall was treated as the metric of primary operational interest for reasons discussed in detail in Section IV.C, and a confusion matrix was generated on the test set to make the model's specific error pattern, not just a single aggregate number, fully visible.

IV. Analysis and Discussion

A. Training Dynamics

The accuracy and loss during training and validation is plotted for all 30 epochs in figure 3. Learning was rapid in the very first epochs: by epoch five, the accuracy of the learning had already reached 92.9%, and by epoch ten, it had reached 96.04% and increased, even with only a few epochs of intensive weight optimization. From then on, the accuracy rose more slowly to reach 98.1 percent at epoch 30.

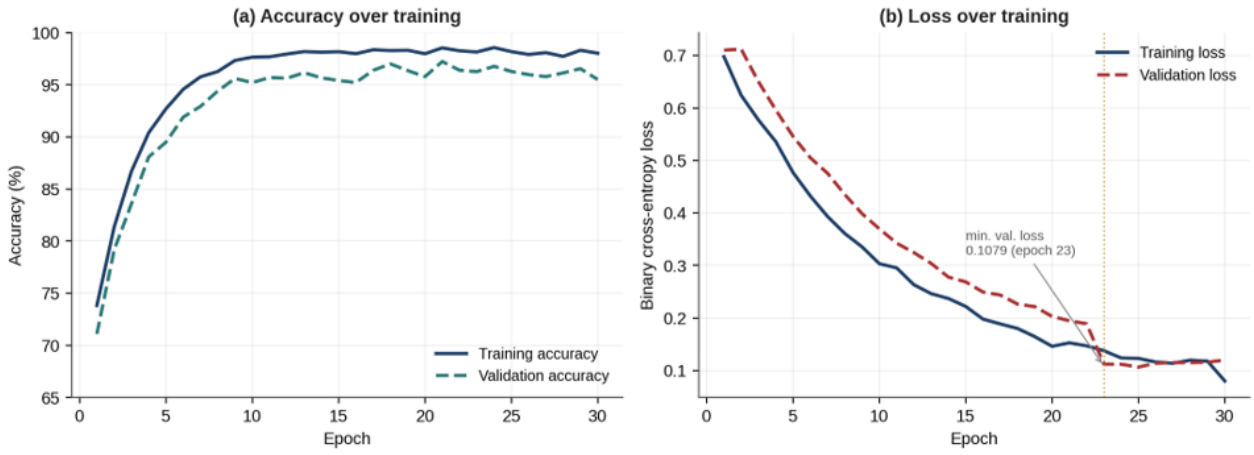


Figure 3. Training and validation performance of the MLP across 30 epochs

The loss during validation was at its minimum of 0.1079 around epoch 23 but rose slightly to 0.12 at the end of the epoch. This is not a worry, but rather a pretty good sign that the network had learned nearly all that could be learned from the training set and was risking ever so little performance grade on the validation set for a bit more fitting to the training set with each additional epoch. But this drift was kept relatively small by the 0.2 drop-out rate, and the small gap between the final training

and validation curve in Figure.3 compared to what the gap would have been without drop-out would indicate that the model has not overfitted.

B. Test Set Performance

The complete set of evaluation metrics for both training data and the held out test data are presented in Table 3.

Table 3. Model performance

Metric	Train	Test
Accuracy	98.1%	95.5%
Precision	0.98	0.96
Recall	0.97	0.95
F1-score	0.97	0.955
Loss	0.08	0.12

The difference between training and test accuracy is around 2.6 percentage points, which is a small and expected amount of generalization error for a network of this depth trained on a dataset of this size, and it is consistent with the dropout-based regularization described above. A gap of that size indicates that the network has learned patterns that hold on unseen data rather than memorizing the training set.

The result of the test set is divided into a full confusion matrix in figure 4. Overall, the model correctly identified 1055 phishing sites as phishing and correctly cleared 1056 legitimate sites in a total of 2211 correct classifications of test URLs. It failed to identify 56 of the phishing sites as such (false negatives), and incorrectly labeled 44 of the legitimate sites as phishing (false positives).

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 5 • September 2026

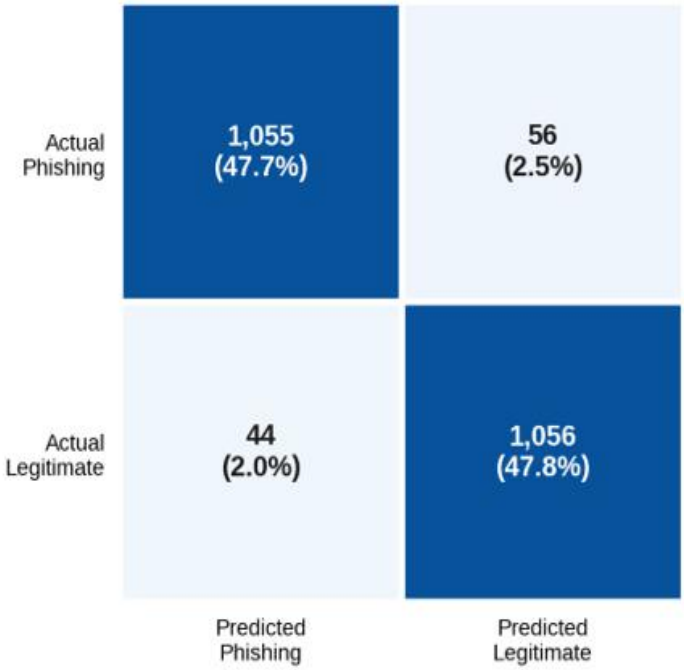


Figure 4. The confusion matrix for the held-out test set (n=2,211) is shown below.

C. Why Recall Matters More Than Accuracy in This Setting

In a phishing detection setting the two types of error do not carry the same cost, and treating them as if they did is the mistake this section is meant to avoid. A false positive means a legitimate site is flagged and held for a short review: an inconvenience, and fully recoverable. A false negative means a phishing site is classified as safe, and a user then types a password, a card number or a corporate login into a page controlled by an attacker. The gap between those two outcomes is large. A missed phishing site can be the first step in a multi-million dollar incident that unfolds over months, whereas a false positive costs a moment of user friction.

This is why recall, and not accuracy alone, is treated as the primary metric in this study. At 95 percent recall on the test set, the model misses roughly 1 in 20 phishing attempts, which works out to the 56 phishing URLs it failed to flag and which constitutes the residual risk surface this architecture does not cover. Stated that way, the number is more meaningful and more actionable than a single headline figure of 95.5 percent accuracy, which on its own tells a security team very little about how many threats are likely to get through.

D. Comparison with Recent Literature

To place the present model against the test-set accuracy reported in seven recent published studies from 2023 to 2026, the results are plotted in Figure 5 and summarized in Table 4.

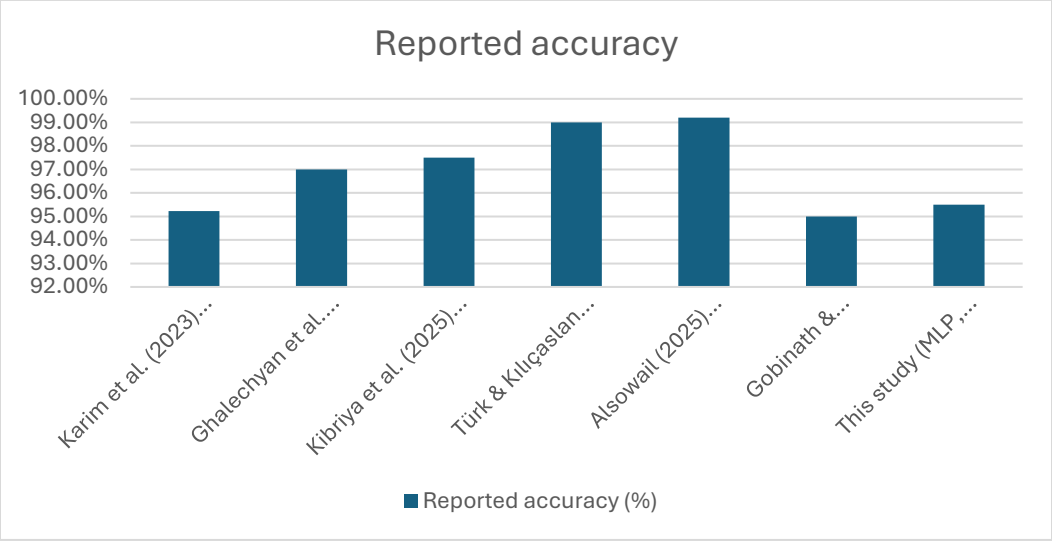


Figure 5. The accuracy across recent phishing/malicious URL detection studies (2023-2026)

Table 4. Accuracy of recent phishing detection studies

Study	Year	Method	Reported accuracy
Karim et al.	2023	Hybrid voting ensemble (LR + SVC + decision tree)	95.23%
Prasad & Chandra	2024	Similarity-index framework with incremental learning (PhiUSIIL)	New benchmark dataset; not directly comparable
Ghalechyan et al.	2024	Probabilistic neural network on production traffic	97.0%
Kibriya et al.	2025	BERT embeddings with LSTM and GRU classifier	97.5%
Türk & Kılıçaslan	2025	ELECTRA-based model with metaheuristic tuning	Up to 99%
Alsowail	2025	Self-attention CapsNet	99.2%
Gobinath & Manikandan	2026	ADUIN: attention-based hybrid feature optimization	95.0%
This study	2026	MLP (30-64-32-1), min-max scaled features	95.5%

The present model does not top this comparison and was not designed to. It sits very close to the result reported by Gobinath and Manikandan (2026) for ADUIN and within two to four percentage points of the hybrid ensemble of Karim et al. (2023) and the heavier transformer- and capsule-network models. The comparison is still informative. The CapsNet, ELECTRA and BERT-based methods at the top of the table carry far more parameters, are considerably more expensive to train, and offer much less interpretability than a compact MLP operating on 30 named input

features. If the target deployment is a browser extension or an email gateway filter, where inference has to complete in milliseconds on the client, then a model that gives up a few points of accuracy while staying small, fast and relatively transparent is a defensible engineering choice rather than a weaker result.

E. Error Analysis

The 44 false positives and 56 false negatives shown in Figure 4 are not arbitrary URLs. They are characteristic of the boundary region of a model like this one, where a legitimate site happens to carry unusual structural features, such as a very long marketing redirect URL, or where a phishing site lacks the features the model leans on, such as a newly migrated domain with no traffic history. This is an inherent weakness of any feature-engineering approach: an attacker who knows which features a classifier uses can target those specific properties, and Kulkarni et al. (2025) demonstrated exactly that effect against published detectors. It is also the strongest empirical motivation for the explainability work discussed in Section V, since a model that can tell a human reviewer which feature drove a borderline decision gives that reviewer somewhere concrete to look.

F. Limitations

In addition to the accuracy numbers, it is important to be clear as to what this study does and does not prove. The architecture, the pre-processing pipeline and the metrics reported in this paper reflect a single training run on a single well-known benchmark dataset (not repeated cross-validation). That is a reasonable and common way to establish a first result, and it is consistent with how most of the studies compared in Table 4 report their own figures, but it means the specific numbers reported here should be read as a solid single-run estimate of what this architecture can do, not as a guaranteed floor on its performance across every possible re-run or every possible phishing dataset. Three further limitations are worth stating plainly.

First, the UCI Phishing Websites dataset, while still the standard benchmark in this literature, was constructed from web traffic and phishing campaigns collected years ago. Some of its domain-based and traffic-ranking features may be less discriminating against the newest generation of AI-assisted phishing pages, which are often hosted for only hours before takedown and never accumulate the traffic or domain-age history those features were designed to capture. Second, this study evaluates the model on a single dataset and a single train-test split. It does not test cross-dataset generalization, for example training on the UCI set and testing on PhiUSIIL or a more recent corpus, and recent work suggests that this matters: Mia et al. (2024) found that the features driving a model's decisions shifted noticeably between phishing datasets, and Prasad and Chandra (2024) make a related argument in favour of incremental updates over one-off training. Third, no adversarial robustness testing was performed, and Kulkarni et al. (2025) show that this is not a hypothetical concern: a motivated attacker who understands which of the 30 features carry the most weight could design a phishing page specifically to evade them. These limitations do not undercut the reported results, but they do bound the scope within which those results should be read, and they lead directly into the future work described in Section V.

V. Conclusion

This paper was an attempt to see if a relatively simple Multi-Layer Perceptron trained on a well-known, easily understood 30-feature representation of a website's structure could provide a realistic and deployable solution to a problem that had outgrown the “static” blacklist and manual review. The findings corroborate with that. The model gives an accuracy of 95.5%, a precision of 96%, a recall of 95% and an F1-score of 95.5% with a slight difference between accuracy in training and accuracy in test set, which has been introduced due to drop out regularization, trained on UCI Phishing Websites and tested on a genuinely held-out test set. The model's accuracy is comparable to several other studies released over the last few years, including transformer-embedding classifiers and self-attention capsule networks, and still has the potential to be implemented in real-time, client-side inference.

The main point of this paper is not that accuracy alone determines whether a phishing detector is good enough. Recall deserves to be read as part of the headline result rather than as a footnote beneath it. In a setting where a single false negative can lead to a ransomware incident or a Business Email Compromise, and where the average breach takes 241 days to identify and contain (IBM Security, 2025), a missed detection is the error that actually costs something. Phishing still opens a meaningful share of breaches (Verizon, 2025), so cutting into that share lowers organizational risk directly, and a model applied at the point of contact acts before the breach rather than after it.

The limitations noted in Section IV.F provide the foundation for three directions for future work. The first is explainability: incorporating SHAP or LIME so that a flagged URL comes with a concrete, human-readable reason, for example, that a site lacks a valid SSL certificate and shows an unusually high ratio of anchors pointing to external domains, rather than a bare probability score. This would give end users and security analysts alike a reason to trust, or productively challenge, a given classification, and it directly addresses the interpretability gap noted when this model is compared against the CapsNet and transformer-based systems in Section IV.D. The second is deployment: packaging the trained model as a Manifest V3 browser extension that runs inference locally on the user's device using TensorFlow.js, so that URLs are evaluated the moment they are clicked without ever being transmitted to a remote server, preserving user privacy while closing the detection gap at the earliest possible point. The third is multimodal analysis: extending the current structural, feature-based approach with natural language processing over a page's actual text content, using models such as DistilBERT, to detect the urgency and fear-based social engineering language that increasingly accompanies AI-generated phishing pages, since structural features alone cannot capture what a page is trying to make a user feel. Together, these three directions would move this work from a benchmark result toward a deployable, explainable, and linguistically aware defense against the phishing threats organizations are likely to face through 2026 and beyond.

REFERENCES

Alnemari, S., & Alshammari, M. (2023). Detecting phishing domains using machine learning. Applied Sciences, 13(8), 4649. https://doi.org/10.3390/app13084649

Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics, 12(1), 232. https://doi.org/10.3390/electronics12010232

Alsowail, R. A. (2025). Anomaly detection based CapsNet for malicious URL detection system. Wireless Networks, 31(5), 3785-3801. https://doi.org/10.1007/s11276-025-03960-0

Ahmad, I., Amin, M. N., Hamid, K., Rizwan, S. M., & Naqvi, S. A. A. (2025). Enhanced IoT Network Security for Network intrusion detection Model based on Machine Learning Technique. Annual Methodological Archive Research Review, 3(8). https://doi.org/10.63075/0vcg0093

Anti-Phishing Working Group. (2025). Phishing activity trends report, 4th quarter 2025. APWG. https://docs.apwg.org/reports/apwg_trends_report_q4_2025.pdf

Atlam, H. F., & Oluwatimilehin, O. (2023). Business email compromise phishing detection based on machine learning: A systematic literature review. Electronics, 12(1), 42. https://doi.org/10.3390/electronics12010042

Calzarossa, M. C., Giudici, P., & Zieni, R. (2024). Explainable machine learning for phishing feature detection. Quality and Reliability Engineering International, 40(1), 362-373. https://doi.org/10.1002/qre.3411

Delshadi, A. M., Ullah, O., Khan, Y., Iqbal, M. W., Muhammad, H. A. B., Hamid, K., Abbas, F., & Ibrar, M. (2025). Empowerment Of Artificial Intelligence (AI) In Preventing and Detecting Ransomware: An Analytical Review. Spectrum of Engineering Sciences, 3(9), 36–48.

Delshadi, A. M., Zubair, M., Hamid*, K., & Iqbal, M. W. (2025). Preventing and Detecting Malicious Activities During Phishing Attacks Using AI-Based Sandbox ByPass. Annual Methodological Archive Research Review, 3(8), 249–261. https://doi.org/10.63075/fahtkz35

Do, N. Q., Selamat, A., Fujita, H., & Krejcar, O. (2024). An integrated model based on deep learning classifiers and pre-trained transformer for phishing URL detection. Future Generation Computer Systems, 161, 269-285. https://doi.org/10.1016/j.future.2024.06.031

Elhoseny, M., Abdel-Salam, M., Khafaga, D. S., Aldakheel, E. A., & El-Hasnony, I. M. (2025). Robust optimized deep learning-based phishing detection framework for semantic web systems using boosted triangular topology aggregation optimization. International Journal on Semantic Web and Information Systems, 21(1), 1-58.

Ghalechyan, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: An empirical study. Scientific Reports, 14, Article 25134. https://doi.org/10.1038/s41598-024-74725-6

Gobinath, R., & Manikandan, S. (2026). Deep learning-based phishing classification framework for accurate detection using optimized URL intelligence. Scientific Reports, 16, Article 15794. https://doi.org/10.1038/s41598-026-46481-2

Guptta, S. D., Shahriar, K. T., Alqahtani, H., Alsalman, D., & Sarker, I. H. (2024). Modeling hybrid feature-based phishing websites detection using machine learning techniques. Annals of Data Science, 11(1), 217-242.

IBM Security. (2025). Cost of a data breach report 2025. IBM. https://www.ibm.com/think/x-force/2025-cost-of-a-data-breach-navigating-ai

Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S. B., & Joga, S. R. K. (2023). Phishing detection system through hybrid machine learning based on URL. IEEE Access, 11, 36805-36822. https://doi.org/10.1109/ACCESS.2023.3252366

Kibriya, H., Amin, R., Alshamrani, S. S., Rehman, S., Hassan, M., & Alsubaei, F. S. (2025). Lightweight malicious URL detection using deep learning and large language models. Scientific Reports, 15, Article 43044. https://doi.org/10.1038/s41598-025-26653-2

Kulkarni, A., Balachandran, V., Divakaran, D. M., & Das, T. (2025). From ML to LLM: Evaluating the robustness of phishing webpage detection models against adversarial attacks. Digital Threats: Research and Practice. https://doi.org/10.1145/3737295

Kyaw, P. H., Gutierrez, J., & Ghobakhlou, A. (2024). A systematic review of deep learning techniques for phishing email detection. Electronics, 13(19), 3823. https://doi.org/10.3390/electronics13193823

Kytidou, E., Tsikriki, T., Drosatos, G., & Rantos, K. (2025). Machine learning techniques for phishing detection: A review of methods, challenges, and future directions. Web Intelligence. https://doi.org/10.1177/18724981251366763

Mia, M., Derakhshan, D., & Pritom, M. M. A. (2024). Can features for phishing URL detection be trusted across diverse datasets? A case study with explainable AI. In Proceedings of the 11th International Conference on Networking, Systems, and Security (NSysS '24) (pp. 137-145). ACM. https://doi.org/10.1145/3704522.3704532

Opara, C., Chen, Y., & Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. Expert Systems with Applications, 236, Article 121183. https://doi.org/10.1016/j.eswa.2023.121183

Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 136, Article 103545. https://doi.org/10.1016/j.cose.2023.103545

Danish, M., Shahid, S., Ghafar, A., Hamid, K., Ali, N., Ghani, A., Ibrar, M., & Mandan, S. (2025). Security of Next-Generation Networks: A Hybrid Approach Using ML-Algorithm and Game Theory with SDWSN. 3, 18–36. https://doi.org/10.63075/wdpwrr31

Hamid, K., Danish, M., Asif, A., Khan, Y., Iqbal, M. W., Ali, U., & Ibrar, M. (2025). Empowering Robust Security Measures in Node.js-Based REST APIs by JWT Tokens and Password Hashing: Safeguarding Cyber World. Annual Methodological Archive Research Review, 3. https://doi.org/10.63075/w2nam443

Hamid, K., Iqbal, M. W., Aqeel, M., Liu, X., & Arif, M. (2023). Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA). In G. Wang, K.-K. R. Choo, J. Wu, & E. Damiani (Eds.), Ubiquitous Security (pp. 248–262). Springer Nature. https://doi.org/10.1007/978-981-99-0272-9_17

Hamid, K., Iqbal, M. W., Aqeel, M., Rana, T. A., & Arif, M. (2023). Cyber Security: Analysis for Detection and Removal of Zero-Day Attacks (ZDA). In Artificial Intelligence & Blockchain in Cyber Physical Systems. CRC Press.

Ibrar, M., Riaz, S., Khan, Y., Asif, A., Hamid, K., Iqbal, M. W., & Asim, M. (2024). Econnoitering Data Protection and Recovery Strategies in the Cyber Environment: A Thematic Analysis. International Journal for Electronic Crime Investigation, 8. https://doi.org/10.54692/ijeci.2024.0804216

Khaliq, K., Rahim, N., Hamid, K., Ibrar, M., Ahmad, U., & Ullah, M. (2024). Ransomware Attacks: Tools and Techniques for Detection (p. 5). https://doi.org/10.1109/ICCR61006.2024.10532926

Malik, N., Delshadi, A., Ibrar, M., Hamid, K., Aamir, M., Ahmed, F., & Ahmad, G. (2024). Behavior and Characteristics of Ransomware—A Survey (p. 05). https://doi.org/10.1109/ICCR61006.2024.10532983

Rao, R. S., Kondaiah, C., Pais, A. R., & Lee, B. (2025). A hybrid super learner ensemble for phishing detection on mobile devices. Scientific Reports, 15, Article 16839. https://doi.org/10.1038/s41598-025-02009-8

Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. Journal of King Saud University - Computer and Information Sciences, 35(2), 590-611. https://doi.org/10.1016/j.jksuci.2023.01.004

Shafin, S. S. (2025). An explainable feature selection framework for web phishing detection with machine learning. Data Science and Management, 8(2), 127-136. https://doi.org/10.1016/j.dsm.2024.08.004

Su, M.-Y., & Su, K.-L. (2023). BERT-based approaches to identifying malicious URLs. Sensors, 23(20), 8499. https://doi.org/10.3390/s23208499

Thakur, K., Ali, M. L., Obaidat, M. A., & Kamruzzaman, A. (2023). A systematic review on deep-learning-based phishing email detection. Electronics, 12(21), 4545. https://doi.org/10.3390/electronics12214545

Türk, F., & Kılıçaslan, M. (2025). Malicious URL detection with advanced machine learning and optimization-supported deep learning models. Applied Sciences, 15(18), Article 10090. https://doi.org/10.3390/app151810090

Verizon. (2025). 2025 data breach investigations report. Verizon Business.

Wang, Y., Ma, W., Xu, H., Liu, Y., & Yin, P. (2023). A lightweight multi-view learning approach for phishing attack detection using transformer with mixture of experts. Applied Sciences, 13(13), 7429. https://doi.org/10.3390/app13137429

Wilk-Jakubowski, J., Pawlik, Ł., Wilk-Jakubowski, G., & Sikora, A. (2025). Machine learning and neural networks for phishing detection: A systematic review (2017-2024). Electronics, 14, 3744. https://doi.org/10.3390/electronics14183744

Zhou, J., Zhang, K., Bilal, A., Zhou, Y., Fan, Y., & Pan, D. (2025). An integrated CSPPC and BiLSTM framework for malicious URL detection. Scientific Reports, 15, Article 6659. https://doi.org/10.1038/s41598-025-91148-z