International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Intelligent Phishing Website Detection System Using Hybrid Machine Learning and Deep Learning Approaches

Rajesh Chauhan[1], Akshay Bhardwaj[2], Shubham Rana[3]
[1]Department of Computer Science, University Institute of Technology, Himachal Pradesh University, Shimla, India
[2]Department of Computer Science and Engineering, University Institute of Technology, Himachal Pradesh University, Shimla, India
[3]Department of Computer Science, University Institute of Technology, Himachal Pradesh University, Shimla, India

Abstract — Phishing websites are still one of the most common infection vectors for stealing credentials, financial information and other sensitive data from users of digital platforms, and there is a growing demand for accurate and deployable automated phishing-detection systems. However, models trained on a set of phishing URLs may experience a large performance drop when transferred to another feature scheme or a different collection window outside the scope of their training. In this work, we compare four classical machine-learning models, and five deep-learning and hybrid architectures for within-domain and cross-domain phishing-website detection. We evaluate our approach over five publicly available benchmark datasets, which contain more than 445k labeled instances in total, including UCI Phishing Websites Dataset, PhiUSIIL Phishing URL Dataset, Web-Page Phishing Detection Dataset, Mendeley Web-Page Phishing Dataset and Phishing Dataset based on Machine Learning. The models are evaluated in terms of classification accuracy, F1-score, transfer of cross domain performance, computational cost, data needs and interpretability. The Hybrid CNN-BiLSTM with attention model achieved the highest accuracy of 98.1% on PhiUSIIL and 85.6% on the transfer task from the PhiUSIIL to the Mendeley Web-Page dataset, outperforming the Linear SVM model by 13.1 percentage points. However, classical models still work well in resource-limited and interpretability-sensitive settings, and a simple BiLSTM with additive attention offers a balanced compromise. The results suggest that model selection for phishing detection should not only focus on predictive performance, but also consider operational deployment constraints.

Keywords— Detecting phishing sites; Cross-domain generalization Hybrid machine learning Deep learning URL feature engineering Deployment trade-offs.

I. INTRODUCTION

The exponential proliferation of e-commerce, online banking and social platforms has not only augmented the ease of use of digital services but also heightened the vulnerability of users to phishing attacks that masquerade legitimate websites to steal credentials and financial information. Phishing can result in personal account compromise, financial loss, damage to institutional trust, and the entry point for larger cyber-intrusions. Automated phishing-website detection has become an important task for cybersecurity and machine learning research because it is not possible to manually validate and blacklist all the new phishing pages created everyday.

Phishing website detection is usually formulated as a supervised binary classification problem, where a model is trained to differentiate between legitimate and fraudulent URLs and web pages based on lexical, host-based, and content-based features. Early detection systems relied on blacklists and heuristic rule sets, which could not keep up with newly registered domains and fast turning phishing infrastructure. Later, feature-based machine learning classifiers that leveraged URL length, special character counts, domain age and HTML structural cues were suggested and demonstrated to generalize better than static blacklists. Neural sequence models and more recently hybrid convolutional-recurrent architectures with attention have further improved the capability of detection systems to capture both local lexical patterns and longer structural dependencies within a URL or page.

Before deep learning, phishing detection was based on hand-engineered lexical and host-based features, and traditional classifiers such as logistic regression, Naive Bayes, and support-vector machines. These methods are efficient and can be trained on modest hardware, but their reliance on hand-engineered feature sets can limit generalization when the phishing infrastructure, top-level-domain distribution or page-construction techniques of the target environment are different from what was seen during training. Previous experimental results have shown that models learned from one scheme of phishing features may suffer from performance degradation when applying to a dataset constructed differently. Therefore, the results reported on an individual benchmark may not necessarily be transferable to another.

To overcome the limitations of hand-engineered features, neural architectures that can learn representations directly from URL character sequences and structured feature vectors have been proposed. Convolutional layers are used to model short, local lexical patterns such as suspicious substrings and character n-grams, while recurrent layers are used to model longer sequential dependencies over a URL or over an ordered feature vector. Attention mechanisms allow a model to assign importance scores to specific characters, tokens or features, giving a partial perspective on what cues led to a classification decision, but attention weights are not a complete explanation of model behavior.

However classical algorithms remain important baselines. Nevertheless, logistic regression and Naive-Bayes classifiers over lexical and host-based features still provide a useful, low-cost phishing-detection capability, especially when interpretability and CPU-only deployment are required. The literature has shown that hybrid architectures combining convolutional feature extraction with recurrent sequence modeling, sometimes with an attention layer, outperform each component alone but at the cost of longer training time, higher memory usage and reduced interpretability.

Cross-domain generalization across phishing-feature schemes is a major problem that has not been sufficiently solved yet. Publicly available phishing datasets are constructed from different sources at different times with different number and type of features extracted, ranging from purely lexical URL statistics to full HTML-content and domain-reputation features. A model trained on a feature scheme might not transfer cleanly to a benchmark built with a different feature-extraction pipeline. As shown in studies, combining multiple phishing-feature schemes into a common representation, with the use of domain-adversarial and self-attention techniques, can improve transfer performance by keeping scheme-specific and scheme-independent information.

Recent reviews now confirm the field has been shifting from purely rule-based and blacklist detection to feature-based classifiers and now to convolutional, recurrent, attention-based and hybrid deep architectures. Although detection accuracy is often reported as comparable across studies, direct comparison is difficult due to differences in the datasets, feature extraction pipelines, and evaluation protocols. Open limitations in the phishing-detection literature have been repeatedly pointed out as dataset dependence, limited cross-scheme generalizability, and inconsistent computational reporting.

Despite these developments, there are still gaps in the current literature. Many studies rely on a single dataset, a limited set of models, or a single performance measure. There is little work comparing classical machine-learning models, recurrent and attention-based neural architectures, and hybrid deep-learning models in a single multi-dataset protocol. For example, the computational requirements and interpretability of a phishing filter, say in a browser extension or an email gateway, are often studied separately from predictive performance, although both are directly related to deployability in the real world.

To address these limitations, the present work evaluates five deep learning and hybrid architectures and four classical machine learning algorithms on five benchmark phishing datasets. The experimental setup consists of four within-domain evaluations and two cross-domain transfer settings. The model performance is evaluated by accuracy, average F1-score and cross-domain retention rate. Other considerations involve training time, inference latency, GPU dependence, memory footprint, relative computational cost, and interpretability. The study objectives are:

- To compare the performances of nine machine learning, deep learning, and hybrid models on the same domain.
- To test the generalization of model over different phishing-feature schemes and data sources.
- To determine the amount of performance retained in the transfer of source to target dataset.
- To compare the computational and memory demands for the evaluated models.
- To study the trade-offs between accuracy, efficiency and interpretability.
- To offer evidence-based guidance for selecting phishing-detection models according to different deployment constraints.

II. METHODOLOGY

In this paper, we present a comparative experimental study to assess the performance of classical machine learning algorithms, deep learning models, a hybrid deep learning architecture and their practical utility for phishing website detection. The methodology includes within domain classification, cross-domain generalizability, computational cost, inference efficiency, and interpretability. Five benchmark datasets and nine classification models were compared in the same experimental setting.

1. Research Questions

The experiments were designed to address the following research questions:

- RQ1: Which are the best machine-learning, deep-learning and hybrid models for classification when trained and tested on the same phishing-feature scheme?
- RQ2: Do the models generalize well from one phishing dataset and feature-extraction scheme to another?
- RQ3: What are the trade-offs between predictive performance, computation, inference efficiency and interpretability and how should these factors be weighed for different deployment circumstances?

These questions test both predictiveness and operational suitability. A model that performs well on one benchmark may not perform well on another benchmark. Similarly, in constrained settings such as limited GPU availability, limited memory, tight inference-latency budgets, or the need for explainability, the most accurate model might not be the best pick.

2. Dataset Selection

In this study, we used five publicly available benchmark datasets from the UCI Machine Learning Repository, Kaggle and Mendeley Data. Counts of records, features, and other properties of each dataset were obtained from the published documentation for each dataset rather than fixed in advance, so that the totals are reflective of the currently published versions of each source. The entire experimental corpus over the five datasets contains more than 445,000 labeled instances. The datasets differ in source domain, feature-extraction scheme, number of features, class balance, and collection period.

Table 1. Summary of the benchmark phishing datasets used in the study

Dataset	Records	Classes	Feature scheme	Features	Source
UCI Phishing Websites (Mohammad et al.)	11,055	Binary	URL lexical + host-based	30–32	UCI ML Repository / Kaggle mirror
PhiUSIIL Phishing URL Dataset	235,795	Binary	URL + HTML content-based	54	UCI ML Repository (2024)
Web-Page Phishing Detection Dataset	88,647	Binary	Domain / host-based	112	Kaggle / Mendeley (Vrbančič)
Mendeley Web-Page Phishing Dataset	100,000	Binary	URL + page structural	20	Mendeley Data (2020)
Phishing Dataset for Machine Learning	10,000	Binary	URL + WHOIS + HTML	50	Kaggle (2018)

The UCI Phishing Websites Dataset collected by Mohammad, Thabtah and McCluskey is one of the most cited phishing benchmarks. It encodes each URL into a fixed vector of lexical and host-based features, e.g., whether the URL contains an IP address, the length of the URL, the use of URL-shortening services, etc. The URLs are labeled as phishing, suspicious, or legitimate; the suspicious label was merged into the phishing class to obtain a binary target consistent with the other datasets. The PhiUSIIL Phishing URL Dataset was generated by Prasad and Chandra and donated to the UCI Machine Learning Repository in 2024 It includes URL-level and HTML-source-level features extracted directly from the live web pages referenced by each URL, and it is relatively recent and large, making it useful for performance measurement against today's phishing infrastructure.

The Web-Page Phishing Detection Dataset associated with Vrbančič has a large number of domain-based and host-based features extracted from WHOIS records, DNS information and page-ranking indicators. In the first cross-domain evaluation, this dataset was used as one of the two transfer sources.

The Mendeley Web-Page Phishing Dataset is a publicly available dataset of URL and page-structure features hosted on Mendeley Data. It is relatively small in terms of features, which makes it useful for testing the transfer of models trained on richer feature schemes to a reduced feature space.

This dataset is obtained from the Kaggle repository and contains 50/50 balanced split of phishing and legitimate pages with URL, WHOIS and HTML based indicators. It served as a baseline in both the within-domain evaluation and the second cross-domain experiment.

For the within-domain evaluation, we used four datasets: UCI Phishing Websites, PhiUSIIL, Web-Page Phishing Detection, and the Phishing Dataset Machine Learning Standard. The target dataset in the main cross-domain experiment was the Mendeley Web-Page Phishing Dataset. Therefore, all five datasets were included in the overall experimental setup, but the in-domain results were reported for four datasets.

3. Text and Feature Preprocessing

Preprocessing was adapted to the needs of classical and deep-learning models, and to the hybrid of tabular and character-sequence inputs present in phishing datasets. Initially,

normalization involved the removal or standardization of malformed URL encodings, duplicate records and fields with missing or constant values. Numerical features were scaled and categorical features were one-hot encoded where appropriate to the schema of each dataset, while the raw URL strings were kept separate for models that operate directly on character sequences.

We applied engineered lexical, host-based, and content-based features on classical machine learning algorithms immediately after min-max scaling. When a raw URL string was available, the engineered feature set was enriched with character n-gram and TF-IDF-weighted token features. The combined vocabulary was limited to a maximum of 50,000 features.

We kept the raw URL character sequences and available page-content tokens intact for deep learning-based models as convolutional and recurrent layers can exploit the full sequence rather than a pre-reduced feature set. Sequences were padded or truncated to a fixed length appropriate for each dataset and then passed into character- or token-embedding layers.

To tackle class imbalance, datasets with an imbalanced phishing-to-legitimate ratio (Web-Page Phishing Detection Dataset, PhiUSIIL) were rebalanced with SMOTE oversampling of the minority class only on the training partition. For the remaining, more balanced datasets, stratified splitting was done to maintain class proportions across partitions.

4. Experimental and Model Design

The experimental setup was designed to evaluate and compare classical machine-learning, deep-learning and hybrid methods in the same pipeline for phishing classification. After data pre-processing and preparation models were divided into two groups. The first group was Logistic Regression, Multinomial Naive Bayes, Linear Support Vector Machine and Random Forest, all based on the same engineered feature representation limited to 50,000 combined features.

The second group was CNN, BiLSTM, BiLSTM with additive attention, BERT-base applied to URL character sequences and a Hybrid CNN-BiLSTM with attention model which combines convolutional local-pattern extraction with recurrent sequence modeling and an attention layer over the resulting representation. Both architecture groups outputted binary predictions of phishing or legitimate.

We used the same general evaluation framework to evaluate all nine models. In the within-domain experiments, the training and test data were from the same dataset. Additional cross-domain experiments were conducted to evaluate the transfer from a source data set to a target data set built using another feature extraction scheme. The accuracy, average F1-score, performance-retention rate, training time, inference latency, memory usage, GPU dependency, relative computational cost and interpretability of the models outputs were compared.

This design allowed the predictive performance and operational suitability to be benchmarked without changing the stated configuration of any model between experiments. The general experimental procedure is summarized in Fig. 1.

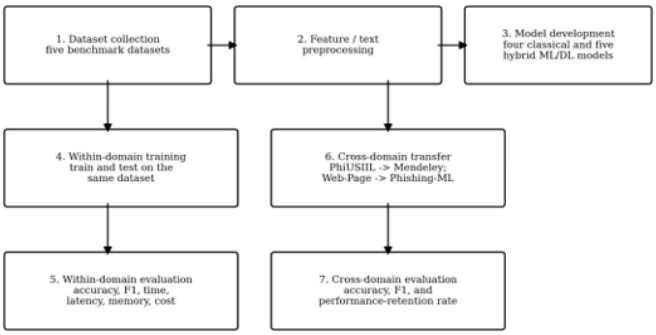

Fig. 1. Overall experimental workflow for within-domain and cross-domain phishing-website detection.

5. Classical Machine-Learning Models

Four classical classifiers were implemented with scikit-learn version 1.3.2:

- Logistic Regression: L2 regularization, C=1.0
- Multinomial Naive Bayes: Laplace smoothing was applied.
- Support Vector Machine - Linear: Linear kernel, C = 1.0
- Random Forest: A nonlinear ensemble-learning baseline.

In an effort to ensure a fair comparison, all four classical models were trained on the same combination of engineered feature and TF-IDF space.

6. Deep-Learning Architectures

Five deep learning and hybrid architecture were tested:

- Convolutional Neural Network: 256 filters with filter sizes 2, 3 and 4 over 128 dimensional character embeddings.
- Bidirectional LSTM: Two-layer BiLSTM, 128 units in each direction.
- BiLSTM with Additive Attention: The BiLSTM representation was augmented with an additive attention mechanism that computed importance scores for individual characters or feature positions.

- BERT-base: We fine-tuned the ~110 million parameter model for three epochs with AdamW at a learning rate of 2 x 10^{-5} on tokenized URL and page-text sequences.
- Hybrid CNN-BiLSTM with Attention: A convolutional feature extraction block feeds into a two-layer BiLSTM with additive attention, merging local lexical-pattern detection with longer-range sequential modeling.

The term additive attention is consistently used because it describes the attention mechanism implemented in the BiLSTM and the hybrid architecture.

7. Experimental Protocol

Within-domain and cross-domain evaluation settings are used in the study.

- Within-domain setting: Each model was trained and tested on samples from the same dataset. The results were reported on UCI Phishing Websites Dataset, PhiUSIIL, Web-Page Phishing Detection Dataset, Phishing Dataset based on Machine Learning.
- The main cross-domain experiment was carried out by considering PhiUSIIL as the source dataset and Mendeley Web-Page Phishing Dataset as the target dataset:
- PhiUSIIL → Mendeley Web-Page Phishing Dataset (1)
- This experiment is testing transfer from a rich URL-plus-HTML-content feature scheme to a reduced URL-and-structure feature scheme.
- The target domain is the Phishing Dataset used in Machine Learning and the source domain is the Web-Page Phishing Detection Dataset for the secondary cross domain experiment:
- Phishing Web-Page Detection Dataset → Machine Learning Phishing Dataset (2)
- Secondary experiment was added to examine whether the pattern of cross-domain performance observed in the primary transfer was similar in another pair of datasets.

8. Evaluation Criteria

Model performance was evaluated using classification accuracy and average F1-score. Moreover, the cross-domain robustness was measured by the performance retention rate:

$$\text{Retention Rate}=(\text{Cross-domain accuracyWithin-domain accuracy})\times 100 \quad (3)$$

We assessed operational suitability in terms of training time, GPU dependence, inference latency per sample, RAM or VRAM requirement, relative computational cost and qualitative rating of interpretability. These criteria provided the basis for comparison of predictive performance as well as model suitability under varying resource, latency and accountability constraints.

III. RESULTS

Here we report the performance of the nine evaluated models on within-domain classification, cross-domain generalization and computational resources use. Further discussion of these results is given in Section 4.

1. Within-Domain Classification Performance

Table 2 provides the classification accuracy of all nine models on four within-domain test datasets, namely, UCI Phishing Websites, PhiUSIIL, Web-Page Phishing Detection, and the Phishing Dataset based on Machine Learning. The Mendeley Web-Page Phishing Dataset was reserved for the main cross domain evaluation and therefore not included in the within domain results. The table also reports the average F1-score for each model.

Table 2. Within-domain classification accuracy and average F1-score across four benchmark datasets

Model	UCI Phishing	PhiUSIIL	Web-Page Phishing	Phishing-ML	Avg. F1
Logistic Regression	92.4%	93.8%	89.6%	90.3%	0.842
Naive Bayes	90.1%	91.5%	87.2%	88.0%	0.821
Linear SVM	94.3%	95.6%	91.4%	92.0%	0.869
Random Forest	93.6%	94.9%	90.8%	91.5%	0.857
CNN	94.9%	96.1%	92.6%	93.2%	0.882
BiLSTM	96.0%	97.0%	94.1%	94.7%	0.905
BiLSTM + Additive Attention	96.7%	97.5%	95.0%	95.6%	0.918
BERT-base	97.4%	97.9%	96.1%	96.7%	0.933
Hybrid CNN-BiLSTM + Attention	97.9%	98.1%	96.8%	97.3%	0.944

The Hybrid CNN-BiLSTM with attention model achieved the best accuracy for all 4 within-domain datasets and the best average F1-score of 0.944. The second best model was the BERT-base model with an average F1-score of 0.933. Among the classical machine-learning models, Linear SVM attained the best average F1-score of 0.869.

For all models, the Web-Page Phishing Detection Dataset using domain- and host-based reputation indicators had a slightly lower classification accuracy when compared to the URL-and-

content-based datasets. For the other datasets, the accuracy of hybrid and transformer-based models was over 97%.

2. Training Time and Interpretability

We report the approximate training time and qualitative interpretability rating for each model we evaluated in Table 3.

Table 3. Training time and interpretability of the evaluated models

Model	Training time	Interpretability
Logistic Regression	<1 min	High
Naive Bayes	<1 min	High
Linear SVM	~7 min	Moderate
Random Forest	~5 min	Moderate
CNN	~9 min	Low
BiLSTM	~28 min	Low
BiLSTM + Additive Attention	~38 min	Moderate
BERT-base	~2.2 hrs	Low
Hybrid CNN-BiLSTM + Attention	~1.1 hrs	Moderate

Both Logistic Regression and Naive Bayes were trained in less than a minute and were considered highly interpretable. Linear SVM and Random Forest required moderate training time and were rated moderately interpretable. For the neural and hybrid models, both BiLSTM with additive attention and Hybrid CNN-BiLSTM with attention model produced moderate interpretability using the attention weights, while CNN, plain BiLSTM and BERT-base were classified as low interpretability.

3. Cross-Domain Generalisation Performance

Table 4 and Table 5 present the cross-domain accuracy and the performance-retention rates for the main and the secondary source-to-target transfer settings, respectively. The models were trained on PhiUSIIL and tested on Mendeley Web-Page Phishing Dataset in the main experiment. In the secondary experiment, we trained the models on the Web-Page Phishing Detection Dataset and tested them on the Phishing Dataset Machine Learning Standard.

Table 4. Cross-domain performance for the PhiUSIIL-to-Mendeley transfer

Model	Accuracy	Retention rate
Logistic Regression	69.8%	74.4%
Naive Bayes	67.2%	73.4%
Linear SVM	72.5%	75.8%
Random Forest	70.6%	75.4%
CNN	76.3%	79.4%
BiLSTM	79.7%	82.2%
BiLSTM + Additive Attention	82.4%	84.5%
BERT-base	84.6%	86.4%
Hybrid CNN-BiLSTM + Attention	85.6%	87.4%

Table 5. Cross-domain performance for the Web-Page-Phishing-to-Phishing-ML transfer

Model	Accuracy	Retention rate
Logistic Regression	71.5%	79.8%
Naive Bayes	68.9%	78.3%
Linear SVM	74.1%	81.1%
Random Forest	72.4%	79.7%
CNN	77.8%	83.5%
BiLSTM	80.9%	85.4%
BiLSTM + Additive Attention	83.6%	87.5%
BERT-base	86.0%	89.0%
Hybrid CNN-BiLSTM + Attention	87.1%	89.6%

The Hybrid CNN-BiLSTM with attention model presented the highest cross-domain accuracy and retention rate in both transfer scenarios. The PhiUSIIL-to-Mendeley experiment performed at 85.6% accuracy with 87.4% retention rate. The Web-Page-Phishing-to-Phishing-ML experiment performed at 87.1% accuracy with 89.6% retention rate. BERT-base and then BiLSTM with additive attention obtained the second position. In the two experiments, the best classical model was linear SVM with 72.5% and 74.1% accuracy.

4. Computational Requirements

Table 6 gives the approximate computational requirements of the nine evaluated models. The reported metrics are training time, GPU dependence, inference latency per sample, memory consumption and relative computational cost. The relative cost was normalized to the baseline with Logistic Regression.

Table 6. Computational requirements and relative cost of the evaluated models

Model	Training time	GPU req.	Latency (ms/sample)	Memory	Relative cost
Logistic Regression	<1 min	No	0.1	~150 MB	1×
Naive Bayes	<1 min	No	0.1	~90 MB	1×
Linear SVM	~7 min	No	0.3	~700 MB	7×

Random Forest	~5 min	No	1.8	~1.1 GB	5×
CNN	~9 min	Yes	1.1	~1.8 GB	28×
BiLSTM	~28 min	Yes	3.8	~2.8 GB	85×
BiLSTM + Additive Attention	~38 min	Yes	5.2	~3.6 GB	120×
BERT-base	~2.2 h	Yes	36	~9.5 GB	650×
Hybrid CNN-BiLSTM + Attention	~1.1 h	Yes	12	~5.2 GB	310×

The least demanding methods were Logistic Regression and Naive Bayes, which trained in less than a minute, had an inference latency of 0.1 ms per sample, and did not need a GPU. The linear SVM was also computationally efficient but relatively expensive (7x the cost of the Logistic Regression baseline).

Among the deep learning models, CNN required the least training time and the least memory capacity. In the middle was BiLSTM with additive attention, with about 38 minutes of training, 3.6 GB of memory and 5.2 ms of inference time per sample. BERT-base had the highest computational cost, with 2.2 hours of training, 9.5 GB of memory, and 36 ms per sample, which is about 650 times more expensive than Logistic Regression. The Hybrid CNN-BiLSTM with attention model was significantly less computationally expensive and memory intensive than BERT-base while yielding the highest overall accuracy and retention rate and thus representing the most favorable trade-off among the top-performing models.

IV. DISCUSSION

The results show a strong performance ranking in both within- and cross-domain evaluation settings. The Hybrid CNN-BiLSTM with attention model outperformed all its counterparts on all four within-domain datasets, with an accuracy of 98.1% on PhiUSIIL and 97.3% on the Phishing Dataset based on Machine Learning, and the best average F1-score of 0.944. The second best model was BERT-base and the best classical ML model was Linear SVM. The Web-Page Phishing Detection Dataset, using domain-reputation and host-based features instead of URL-and-content features, presented a slightly harder problem for all models, consistent with the idea that reputation-based features offer less discriminative signal at inference time than lexical or content-based clues.

Cross-domain evaluation reduced the performance of all models, confirming that high within-domain accuracy does not guaranty robust generalization over differently constructed phishing feature schemes. The Hybrid model achieved 85.6% accuracy in the transfer from PhiUSIIL to Mendeley compared to 72.5% for Linear SVM, a difference of 13.1 percentage points. For Web-Page-Phishing-to-Phishing-ML transfer, Hybrid model achieved 87.1% accuracy. The results indicate that the ratio of performance retained in hybrid and transformer based architectures is higher in the presence of a shift in feature-extraction scheme between source and target datasets. BiLSTM with additive attention achieved competitive accuracy in both transfer settings (82.4%, 83.6%) with significantly less consumption of computational resources than BERT-base and the hybrid model.

These results are consistent with the existing literature on phishing detection in general: neural architectures that blend local pattern detection and sequential modeling have repeatedly been shown to capture richer lexical and structural information than traditional feature-based classifiers, and cross-dataset evaluations in previous studies have also reported significant degradation when classifiers are transferred to a differently constructed, unseen feature scheme. The better performance of BERT-base and hybrid architecture supports the general observation that contextual and combined convolutional-recurrent representations encode more transferable information than fixed, hand-engineered feature vectors alone.

The lesson to be learned is that model selection for phishing-website detection should not be based solely on predictive accuracy. The hybrid CNN-BiLSTM with attention model is the choice where maximum cross-domain robustness is needed and moderate GPU resources are available, as its computational cost, although higher than the classical baselines, is still considerably lower than a full BERT-base fine-tune. For applications where strict interpretability, CPU-only deployment, or very fast prototyping (e.g. a lightweight browser-extension filter) are desired, classical models are still a reasonable choice. BiLSTM with additive attention provides a middle ground between cross-domain accuracy, computational efficiency and partial interpretability.

There are some limitations in this study. It uses only URL, lexical, host and available HTML-content features and does not use visual page-rendering, screenshot-based or real-time DNS/WHOIS lookups that some phishing-detection systems also use. There are only two dataset pairs for the cross-domain evaluation and all datasets mainly contain English-language and Latin-script domains. We have not taken into account statistical significance testing, multiple runs of experiments, adversarial URL obfuscation and temporal concept drift in phishing-campaign infrastructure. Differences in the labeling procedures and collection artifacts in each source dataset may also affect the results.

Future research directions include domain-adaptation and few-shot learning techniques for rapidly emerging phishing campaigns, multimodal architectures combining textual, visual and network-level indicators, distilled transformer and hybrid architectures optimized for on-device or browser-extension latency and memory budgets, and continual-learning architectures that can adapt to changing phishing infrastructure without full retraining. Furthermore, there is a need for broader multilingual and multi-region phishing benchmarks to assess the generalization over a broader range of linguistic and regional web ecosystems.

V. CONCLUSION

In this study, we have compared four classical machine-learning models and five deep-learning and hybrid architectures on five benchmark datasets for within-domain and cross-domain phishing-website detection. The results suggest that, besides classification accuracy, cross-domain robustness, computation cost, inference requirements, and interpretability are equally important factors to consider when choosing a phishing-detection model. The Hybrid CNN-BiLSTM with attention model yielded the best accuracy of 98.1% on PhiUSIIL and 85.6% on the main PhiUSIIL-to-Mendeley cross-domain experiment, outperforming the best classical model, Linear SVM, by 13.1 percentage

points in this transfer setting. Another experiment, the secondary Web-Page-Phishing-to-Phishing-ML experiment, gave a similar ranking, validating the better generalization performance of hybrid and transformer-based models. But there is no one model that works best for all deployments. Logistic Regression, Naive Bayes, and Linear SVM require much less training time, memory and compute, and are well suited for CPU-based, rapid-prototyping, and interpretability-focused deployments such as lightweight email or browser filters. The Hybrid CNN-BiLSTM with attention model strikes a balance with the best cross-domain effectiveness while requiring much less resources than a full BERT-base fine-tune. To summarize, the results show the limit of using within-domain accuracy only as a measure of the effectiveness of phishing-detection systems: when deploying a model, predictive performance should be balanced against operational constraints. Browser-level warnings, domain reputation services, user security awareness and platform-level anti-phishing controls should be augmented by, not supplanted by, automated phishing detection.

REFERENCES

1. R.M. Mohammad, F. Thabtah, L. McCluskey, An evaluation of features associated with phishing websites using an automated technique, in Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions (2012).
2. A. Prasad, S. Chandra, PhiUSIIL: a diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning, Computers & Security 134 (2024).
3. G. Vrbančič, I. Fister Jr., V. Podgorelec, Datasets for phishing websites detection, Data Brief 33, 106438 (2020)
4. Web page phishing detection data set. Mendeley Data. 2026. Phishing dataset used in machine learning Kaggle, 2026.
5. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in Proceedings of NAACL-HLT 2019, 4171-4186 (2019).
6. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., “Attention is all you need,” Adv. Neural Inf. Process. Syst., vol. 30, pp. 5998–6008, 2017.
7. O.K. Sahingoz, E. Buber, O. Demir, B. Diri, Machine learning based phishing detection from URLs, Expert Systems with Applications 117, 345–357 (2019).
8. C. Opara, Y. Chen, B. Wei, Look before you leap: detecting phishing web pages by exploiting raw URL and HTML characteristics, Expert Systems with Applications 236, 121183 (2024).
9. W. Sarasjati, S. Rustad, Purwanto, H.A. Santoso, Muljono, A. Syukur, F.A. Rafrastara, D.R.I.M. Setiadi, “Comparison of classification algorithms for website phishing detection on multiple datasets,” 2022 International Seminar on Application for Technology of Information and Communication, 2022.
10. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, et al., Domain-adversarial training of neural networks. J. Mach. Learn. Res. 17(59), pp. 1–35 (2016).