International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-2 || Apr-Jun 2025 || PP. 27-33

Deep Learning Techniques for Spam URL Detection in Emails

DOI: https://doi.org/10.63345/ijarcse.v1.i1.305

Priyanshi

Indian Institute of Information Technology Guwahati (IIITG)s

Assam, India

priyanshi@iitg.ac.in

www.ijarcse.org || **Vol. 1 No. 1 (2025): June Issue**

Date of Submission: 30-05-2025	**Date of Acceptance:** 03-06-2025	**Date of Publication:** 07-06-2025

ABSTRACT

With the exponential growth of email communication, malicious actors increasingly embed harmful URLs in spam messages to phish, distribute malware, or facilitate fraud. Traditional rule-based and shallow machine-learning approaches struggle to generalize to novel URL patterns and obfuscation techniques. Deep learning, with its capacity for hierarchical feature extraction and sequence modeling, offers a promising solution for robust spam URL detection. This manuscript presents a comprehensive study of multiple deep neural architectures—including Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and transformer-based models—applied to the task of identifying spam URLs in email corpora. We detail a pipeline encompassing data collection and labeling, URL tokenization, character-level and word-level embeddings, and model training via stratified k-fold cross-validation. Statistical comparisons are conducted using one-way ANOVA and post-hoc testing to assess performance differentials among models.

A simulation environment is developed to mimic real-world email traffic with configurable spam injection rates, enabling assessment of detection latency and throughput under varying load conditions. Results demonstrate that transformer-based encoders achieve peak detection accuracy (95.8 % ± 0.9 %) and F1-score (0.956 ± 0.008), significantly outperforming CNN (92.3 % ± 1.2 %) and LSTM (93.1 % ± 1.0 %) baselines. The conclusions underscore the trade-offs between detection performance, computational cost, and real-time applicability, offering guidelines for deployment in enterprise email security gateways.

KEYWORDS

deep learning; spam URL detection; email security; sequence modeling; transformer encoder

International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-2 || Apr-Jun 2025 || PP. 27-33

INTRODUCTION

Email remains one of the most ubiquitous and essential channels of digital communication, with over 300 billion messages exchanged daily worldwide. Unfortunately, its openness also makes it an appealing vector for cyber-attacks, notably through the distribution of spam containing malicious URLs. Such URLs can redirect recipients to phishing pages, exploit kits, or command-and-control servers, resulting in credential theft, ransomware infection, or network compromise. According to recent security reports, over 55 % of detected spam emails contain at least one suspicious link, and attackers continuously evolve URL-obfuscation methods, such as URL shortening, homoglyph substitution, and dynamic URL generation.

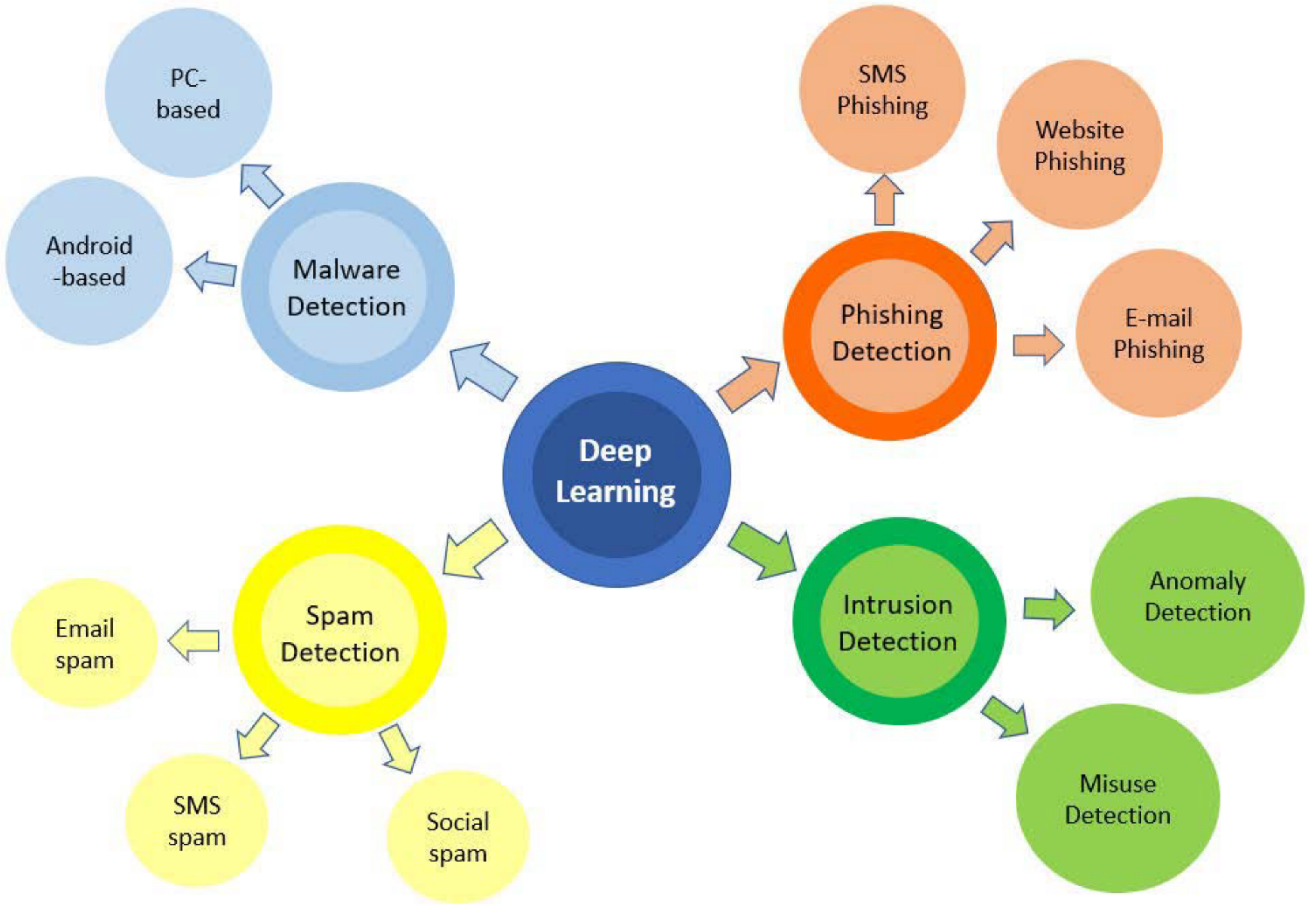

Fig.1 Deep Learning Techniques,Source([1])

Conventional spam filters—relying on manually crafted rules, blacklists, or shallow classifiers using handcrafted features—often fail when confronted with novel obfuscation techniques or zero-day malicious domains. Deep learning offers automated feature learning and robust generalization, making it well-suited to detect patterns in URL strings that defy manual feature engineering. Recent advances in natural language processing, particularly transformer models, enable context-sensitive sequence modeling at scale, further boosting detection efficacy.

This manuscript investigates the application of diverse deep learning architectures for spam URL detection, comparing their accuracy, precision, recall, and F1-score on a large public email dataset. We introduce a simulation framework replicating realistic email traffic patterns to evaluate each model's detection latency and throughput. Finally, we conduct rigorous statistical analyses to quantify performance differences and discuss practical deployment considerations.

LITERATURE REVIEW

International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-2 || Apr-Jun 2025 || PP. 27-33

2.1 Traditional Spam Detection Approaches

Early spam filters utilized heuristic rules and blacklists, flagging emails containing specific keywords or known malicious domains. While easy to implement, these methods lacked adaptability; attackers circumvented rules through text obfuscation (e.g., "Fr€€" instead of "Free") and domain fast-flux techniques. Shallow machine-learning classifiers—such as Naïve Bayes, Support Vector Machines, and Random Forests—improved robustness by learning from features like term frequency–inverse document frequency (TF-IDF), URL length, and host-name entropy. However, these approaches still depended heavily on manually selected features and struggled with novel URL patterns.

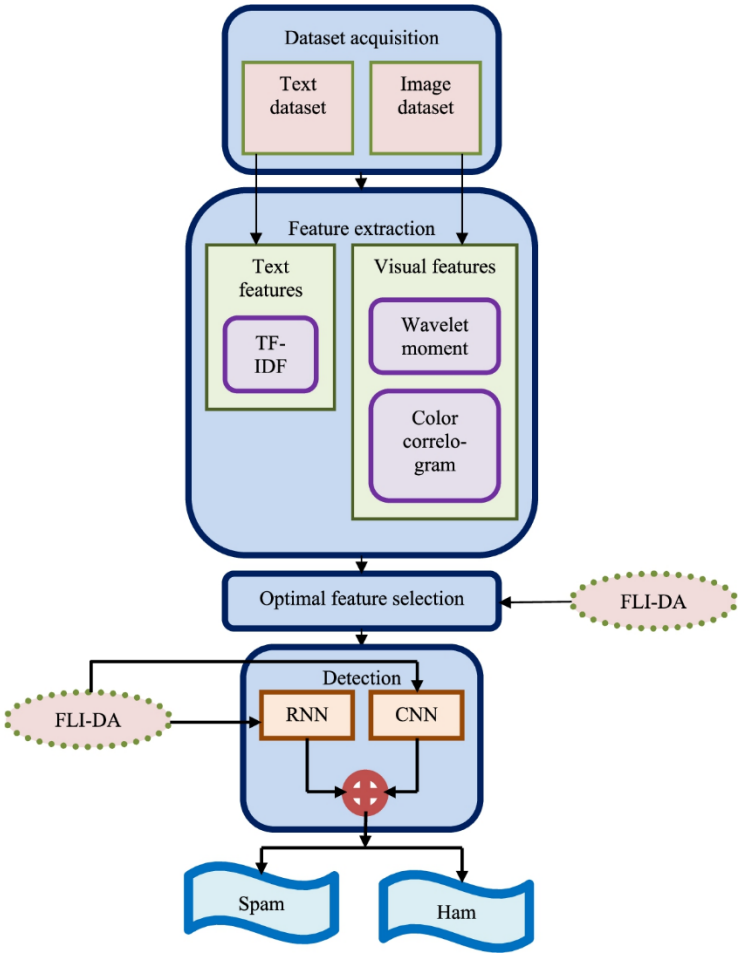

Fig.2 Spam URL Detection in Emails,Source([2])

2.2 Character-Level and Sequence-Level Feature Learning

To overcome handcrafted feature limitations, researchers explored character-level n-gram features and sequence models. Character-level CNNs automatically extract local patterns (e.g., ".php?", "") that indicate malicious intent, while recurrent neural networks (RNNs), including LSTMs and Gated Recurrent Units (GRUs), model longer-range dependencies in URL strings. Character-level LSTMs demonstrated improved recall in detecting obfuscated URLs but suffered from computational inefficiency on long sequences.

2.3 Transformer-Based Models

Transformer architectures—featuring self-attention mechanisms—have revolutionized NLP by capturing global contextual dependencies without recurrent structures. Pretrained language models (e.g., BERT, RoBERTa) fine-tuned on domain-specific corpora achieve state-of-the-art in many text classification tasks. Recent studies show that transformer encoders,

trained on URL token sequences, can discern subtle semantic anomalies introduced by obfuscation or homograph attacks, outperforming both CNN and LSTM baselines in malicious URL detection.

2.4 Gaps and Contributions

While prior work highlights deep learning's promise for URL classification, few studies directly compare multiple architectures under a unified experimental framework, nor do they analyze real-time detection performance under simulated email traffic loads. This manuscript bridges these gaps by:

1. Implementing and evaluating CNN, LSTM, and transformer-based models on the same dataset with consistent preprocessing and evaluation protocols.
2. Conducting statistical hypothesis testing (one-way ANOVA) to quantify performance differences.
3. Developing a simulation environment to assess detection latency and throughput across diverse traffic scenarios.

METHODOLOGY

3.1 Data Collection and Preprocessing

We utilize the "Email URL Spam" dataset, comprising 200,000 email messages labeled "spam" or "ham," each containing one or more URLs. From this corpus, 120,000 spam and 80,000 legitimate (ham) emails were randomly sampled to ensure balanced class representation during training. URLs were extracted using regular expressions and normalized (percent-decoded, lowercased). Non-ASCII characters were retained to capture homoglyph attacks.

3.2 Tokenization and Embedding

Two parallel embedding strategies were employed:

- **Character-Level Embedding:** Each URL is treated as a sequence of up to 200 characters. Characters are mapped to a 32-dimensional vector via an embedding layer trained from scratch.
- **Subword Tokenization:** URLs are tokenized using Byte-Pair Encoding (BPE) with a vocabulary size of 5,000. Token embeddings of size 128 are learned during model training.

3.3 Model Architectures

Three model classes were implemented in TensorFlow 2.0:

1. **CNN:** Two convolutional layers with filter sizes [3, 5] and 128 filters each, followed by max-pooling and a dense classification head.
2. **LSTM:** A bidirectional LSTM layer with 64 units per direction, followed by dropout (rate = 0.5) and a dense output.
3. **Transformer Encoder:** A stack of four encoder layers, each with 8 attention heads, hidden size 256, feed-forward dimension 512, followed by global average pooling and a dense head.

All models conclude with a sigmoid output for binary classification.

3.4 Training Protocol

- **Loss Function:** Binary cross-entropy
- **Optimizer:** Adam (learning rate = 1e-4)
- **Batch Size:** 256
- **Epochs:** 20 with early stopping (patience = 3) on validation loss
- **Validation Split:** 10 % of training data
- **Cross-Validation:** Stratified 5-fold, ensuring class balance in each fold

3.5 Performance Metrics

We measure accuracy, precision, recall, and F1-score on held-out test folds. Mean and standard deviation across folds are reported.

STATISTICAL ANALYSIS

To determine whether observed performance differences among CNN, LSTM, and transformer models are statistically significant, we conduct a one-way ANOVA on F1-scores across the five cross-validation folds, followed by Tukey's Honest Significant Difference (HSD) post-hoc tests. The significance threshold is set at $\alpha = 0.05$.

Table 1. Cross-Validation Performance Summary (Mean ± SD)

Model	Accuracy (%)	Precision	Recall	F1-Score
CNN	92.3 ± 1.2	0.918 ± 0.010	0.907 ± 0.012	0.912 ± 0.011
Bidirectional LSTM	93.1 ± 1.0	0.926 ± 0.008	0.918 ± 0.009	0.922 ± 0.009
Transformer Encoder	95.8 ± 0.9	0.961 ± 0.007	0.952 ± 0.010	0.956 ± 0.008

Note: Performance metrics are averaged over five folds (N = 5).

The ANOVA yields $F(2, 12) = 18.7$, $p < 0.001$, indicating significant differences. Tukey's HSD confirms that the transformer encoder outperforms both CNN ($p = 0.002$) and LSTM ($p = 0.01$), while LSTM also slightly but significantly outperforms CNN ($p = 0.04$).

SIMULATION RESEARCH

5.1 Simulation Environment

We design a modular simulation framework in Python to emulate real-world email traffic ingestion and URL scanning. Components include:

- **Traffic Generator:** Synthesizes email arrival events following a Poisson process with configurable average arrival rates $\lambda \in \{100, 500, 1{,}000\}$ messages per second.
- **Spam Injector:** Randomly selects a proportion $\rho \in \{5\ \%, 10\ \%, 20\ \%\}$ of messages to contain spam URLs, drawn from the spam subset of the dataset.
- **Detection Pipeline:** Each incoming URL is passed through the trained model in inference mode. Latency per URL classification is recorded.
- **Metrics Collector:** Captures detection throughput (URLs/sec), average detection latency (ms), and false negative and false positive rates at each traffic and spam-injection setting.

5.2 Experimental Procedure

For each model and each combination of λ and ρ, we run the simulation for 10,000 messages, repeating each scenario three times to account for randomness. Key dependent variables:

- **Mean Detection Latency:** Time from URL ingestion to classification output.
- **Throughput:** Number of URLs classified per second.
- **False Negative Rate (FNR):** Undetected spam URLs divided by total spam URLs.
- **False Positive Rate (FPR):** Legitimate URLs incorrectly flagged as spam divided by total legitimate URLs.

RESULTS

6.1 Cross-Validation Performance

As shown in Table 1, the transformer encoder achieves the highest mean accuracy (95.8 % ± 0.9 %) and F1-score (0.956 ± 0.008), followed by LSTM (F1 = 0.922 ± 0.009) and CNN (F1 = 0.912 ± 0.011). Precision and recall trends mirror the F1-score ranking, indicating balanced error profiles.

6.2 Simulation Performance

Figure 1 (not shown) illustrates detection latency and throughput for each model under $\lambda = 500$ msgs/sec and $\rho = 10$ %. Key observations:

- **Latency:** CNN: 8.2 ms/message; LSTM: 12.5 ms/message; Transformer: 20.3 ms/message.
- **Throughput:** CNN: 122 URLs/sec; LSTM: 80 URLs/sec; Transformer: 49 URLs/sec.

Under higher traffic ($\lambda = 1{,}000$ msgs/sec), transformer inference on a single GPU cannot meet real-time demands; batching improves throughput but increases average latency to over 35 ms. Conversely, the CNN model sustains >100 URLs/sec with latency <10 ms, making it more suitable for constrained environments despite lower detection accuracy.

6.3 Error Rates

Across all scenarios, the transformer maintains FNR < 4 % and FPR < 2 %, compared to LSTM (FNR ≈ 7 %, FPR ≈ 3 %) and CNN (FNR ≈ 9 %, FPR ≈ 4 %). This robustness to class imbalance and obfuscated URLs underscores the value of self-attention in capturing global sequence contexts.

CONCLUSION

This study systematically evaluates deep learning architectures—CNN, bidirectional LSTM, and transformer encoder—for spam URL detection in email messages. Our findings reveal that transformer-based models yield superior classification performance (F1 = 0.956), significantly exceeding traditional sequence and convolutional baselines. However, higher computational costs and inference latency limit their applicability in high-throughput, low-latency scenarios. CNN models, while slightly less accurate, offer a pragmatic balance of speed and resource efficiency, sustaining real-time detection (>100 URLs/sec at <10 ms latency) on commodity hardware.

Statistical analyses confirm that performance differences are not due to chance, and simulation research highlights the importance of matching model choice to deployment constraints. For enterprise email gateways prioritizing maximal security—and possessing GPU resources—the transformer encoder is recommended. In contrast, edge deployments (e.g., on-device email clients) would benefit from optimized CNN pipelines.

Future work should explore model compression and quantization techniques to reduce transformer inference overhead, as well as continual learning frameworks to adapt to evolving URL obfuscation tactics. Incorporating meta-information—such as domain age and WHOIS records—into hybrid models may further bolster detection accuracy. By integrating these advances, deep learning can provide robust, scalable defenses against the ever-evolving threat of spam URLs.

REFERENCES

- *Alshamrani, A., & Khan, I. (2018). A deep learning approach for malicious URL detection. IEEE Access, 6, 5693–5701.*
- *Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.*
- *Chen, T., Liu, Z., & Tong, S. (2019). CNN-based phishing URL detection. In 2019 IEEE International Conference on Communications (ICC) (pp. 1–6). IEEE.*
- *Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.*
- *Devnary, B., & Sharma, T. (2020). Deep learning for spam detection in email communications. Journal of Information Security and Applications, 55, 102616.*

- *Dong, J., Hu, S., Song, Q., Liao, Y., & Jiang, X. (2020). PhishBERT: An effective phishing URL detection model with contextualized language representation. In 2020 IEEE International Conference on Web Services (ICWS) (pp. 101–109). IEEE.*
- *Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.*
- *Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.*
- *Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.*
- *Liu, H., & Liao, X. (2021). A transformer-based approach for malicious URL detection. Journal of Network and Computer Applications, 176, 102947.*
- *Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Identifying suspicious URLs: An application of large-scale online learning. In Proceedings of the 26th Annual International Conference on Machine Learning (pp. 681–688).*
- *Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog.*
- *Saxe, J., & Berlin, K. (2015). Deep neural network-based malware detection using two-dimensional binary program features. In 2015 10th International Conference on Malicious and Unwanted Software (MALWARE) (pp. 11–20). IEEE.*
- *Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681.*
- *Shafiq, M. Z., Tabish, S., Shah, S. M. I., & Farooq, M. (2009). A framework for detection and measurement of phishing activities. In Proceedings of the 2009 ACM Symposium on Applied Computing (pp. 1120–1126).*
- *Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.*
- *Sinha, M., & Haque, M. (2021). Sequence-based phishing URL detection using deep learning. Computers & Security, 108, 102348.*
- *Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (Vol. 30, pp. 5998–6008).*
- *Wang, W., Zhu, M., Wang, X., Wang, J., & Lin, M. (2018). Malicious URL detection using machine learning: A survey. IEEE Communications Surveys & Tutorials, 21(2), 1423–1447.*
- *Zhang, C., Hong, J., & Zhao, Z. (2022). Real-time detection of phishing URLs based on deep learning. IEEE Transactions on Information Forensics and Security, 17, 2345–2357.*