

Layered Deep Learning Approach for Malicious URL Detection via Neural Sieve Cascade

Aditya S^a, Ashwin Naresh M^a, Padmacharan R^a, Shree Santh B^a and S.Manimaran^{a,*}

^aAmrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India

ARTICLE INFO

Keywords:

Malicious URL Detection, Cybersecurity, Deep Learning, Cascade Architecture, Transformer Models

ABSTRACT

Users face greater risks due to the presence of harmful sites. Therefore, the solution must be both accurate and fast enough to operate in real time against potential threats. In this paper, a triple-filter classification system is presented to reward speed and accuracy in classifying malicious URLs: the Neural Sieve Cascade. The first stage consists of a super-simple-and-fast classifier (TF-IDF with Logistic Regression) that filters out obvious cases. The remaining links are passed to phase two, where deep learning models of CNN, LSTM, and BiLSTM analyze the structure and patterns of the URL text. Then, for a deeper view of the dilemma, TinyBERT considers the most difficult links at the third stage. A big group of 651,191 URLs consisting of benign, defacement, phishing, and malware samples was trained and tested. The NSC's overall accuracy was 97.92%, with Precision, Recall, and F1 rates all exceeding 91% for all classes. It also reduced false negatives in phishing and malware by 15% and 12%, respectively, compared to the standalone models. This way, the system maximizes the use of computing resources by quickly checking URLs while maintaining very high accuracy. Hence, it stands out as a very effective solution capable of defending in real time against cyberattacks.

1. Introduction

With the rapid growth of the Internet and digital services, there is no limit to the ways cybercriminals can exploit existing loopholes. Malicious URLs rank as one of the most common attacks-because they can be used for phishing, malware distribution, credential theft, website defacement, etc. These malicious URLs are designed to either get the user to reveal sensitive information or to download malicious payloads. With about a billion URLs being accessed every day, even a few hundred malicious URL hits can bring in a ton of financial losses, privacy issues, and erosion in trust from the user side [1], [2].

Conventional defense methods against a malicious URL included blacklists, whitelists, and manual rules, but with certain limitations. Blacklists depend on domains that have been reported as malicious; therefore, they are always at least a step behind new or obfuscated URLs, which browser-vendors attempt to get around with fast-flux, disposable domains, and lexical padding [3], [4], [5]. Similarly, being made by human beings, these rule-based systems lack the agility to keep up with cyber attackers [1]. This, in turn, leads to an urgent call for dynamic, automated, and intelligent detection frameworks.

Malicious websites in the wild have presented the highest risk of digital crime ever in phishing, malware, and defacement attacks in recent times. Among these is phishing; APWG reported a total of over 932,000 phishing attacks in the third quarter of 2024, where countless fake websites and brand-spoofing attacks were included [6]. With the ever-increasing defacement rate, the defect becomes quite serious. A 2025 study has seen 453 websites defaced in

Indonesia in just one month, with the attackers using gambling content across more than 5,900 domains and 8,800 URLs [7]. Shifts of malware incidents, however, caused even greater damage. In the year 2017, the WannaCry ransomware managed to infect over 200,000 computers across 150 countries, thus shutting down hospitals, businesses, and transport systems, causing losses that ran into billions of dollars [8]. From the cases, it becomes clear that phishing, defacement, and malware URLs are all threats, not just issues to set before research, which are why we propose a system capable of effectively detecting all three.

The machine-learning activities were considered acceptable alternatives. Extraction of lexical and host-based attributes allowed ML classifiers to increase the accuracy levels over blacklisting, such as Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines (SVMs) [9]. Sahingoz et al. [10] stated that detection rates shall be improved in ensemble classifiers as opposed to shallow ones, or Manjeri et al.[9] declared that URL-based features are more generalized toward unseen domains, still classical ML approaches are greatly relying on manual feature engineering that restricts their generalizability, making them vulnerable to highly sophisticated attacks [1].

Another revolution happened in suspicious-URL detection with the arrival of DL architectures. Deep CNNs learn character-level patterns and local n-grams that automatically build features of spammers' seeding domains [11], [12], [13]. For example, Liu and Lee [12] reported that the CNN is resistant to being obfuscated for spam purposes. Following that, domains learned sequential dependencies within sub-domain and domain structures via LSTMs [14].

The Bi-directional LSTM and Hybrid CNN-LSTM architectures served as further enhancements. According to both Ren et al.[15] and Liang et al.[4] a BiLSTM with attention

*Corresponding author. Tel.: +91 7904132156

 s._manimaran@cb.amrita.edu (S.Manimaran)
ORCID(s):

mechanism is capable of learning contextual dependencies in two directions and thus generalizing better in the case of phishing attacks embedded within complex URL structures. Vazhayil et al.[16] demonstrated that a CNN-LSTM hybrid model was better able to determine local features with a global understanding of semantics and thereby had a higher F1-score compared to the separate implementations of either model. Huang et al.[2] on the other hand, reported better performances from CNN-BiLSTM-CNN hybrids than those of its stand-alone counterparts. However, the added computational cost given by these improved models could stand as a major threat to the implementation of such systems in real-time detection systems.

Recently, the advent of Transformer-based methods transitioned them into the highest echelon in the state-of-the-art class. Thereby, unlike the sequential network, Transformer treats URL strings not as a sequence per se but by self-attending to all parts of a URL string to create richer contextual analyses. He et al.[17] experimented with a TinyBERT stacking model for phishing detection and showed that it gave better recall than the CNN and LSTM baselines. Further, Afzal et al.[5] incorporated semantic vector-space models within URL classification, hence demonstrating their capability to even detect forms of hiding under adversarial conditions. Although in terms of detecting accuracies, Transformers outperformed all other modes, the computational overhead prevented them from being utilized in a high-throughput scenario [5],[17].

Other angles of this theory would be to maximize system efficiency and to place more redundancy by having a different ensemble or cascaded modeling plants. Chauhan et al.[19] conversed about the two-stage cascaded system, wherein at stage one, lightweight models filtered away benign traffic, while the heavier-duty DL looked at cases that were suspicious. Most of them remain two-stage in operation and never transcend to three-stage filtering by classical ML, DL, and transformers.

This gap is the impetus for our work. This system has been proposed, called the Neural Sieve Classifier (NSC), as a detection pipeline in three stages- comprising Logistic Regression (Sieve-1), an ensemble of CNN/LSTM/BiLSTM (Sieve-2), and TinyBERT (Sieve-3)- filtering out uncertain cases. This progressive pipeline, therefore, makes fast and precise judgments. On a dataset of 651,191 URLs, NSC outperforms notably two-stage pipeline or individual stand-alone models, with an accuracy of 97.92%.

The contributions are listed as follows:

- We proposed a new cascaded framework that progressively integrates Logistic Regression,CNN–LSTM–BiLSTM ensemble, and TinyBERT. Such a multi-sieve pipeline optimizes the balance between speed and robustness, in that lightweight learners should be able to filter all but the most ambiguous and adversarial URLs, which have a deeper layer to deal with them.

- We implemented a confidence-based administrator (90%). That controls escalation between each sieves, so that computationally expensive models can be spared for URL-server classification only.
- Our framework was trained and tested on a dataset of 651,191 URLs[18] divided into four categories (benign, phishing, malware, defacement), whereas most studies consider only a binary classification (benign vs. malicious) and achieved a 97.92% of accuracy, exhibiting high, per-class precision and recall (above 94% for most classes), also calculating a strong rejection rate of false negatives: 15% and 12% per phishing and malware, respectively, compared to what was achieved by individual models.
- Through exhaustive comparative analysis against state-of-the-art approaches, channelling RandomForest, LightGBM, and XGBoost, the NSC proves its sturdiness against adversarial obfuscations.

We continue further in this paper with the description of a literature review on malicious URL detection methods in Section 2, focusing on deep learning- and transformer-based methods. Section 3 follows with a discussion on the actual three-sieve NSC architecture and setup. The results and analysis are presented well in comparison in Section 4. Finally, Section 5 concludes the paper and sets the grounds for future work.

2. Literature Review

2.1. Classical Machine Learning for Malicious URL Detection

Early malicious URL detection approaches relied heavily on lexical features (such as URL length, tokens, and special characters) and host-based attributes (e.g., WHOIS data, IP reputation), which were then classified using lightweight ML models such as Logistic Regression, Decision Trees, Random Forest, and SVM. These models were fast and interpretable but lacked resilience against adversarial obfuscations.

Darling et al.[20] proposed a purely lexical approach using n-gram language modeling combined with the J48 decision tree to classify malicious web pages. Their system achieved 99.1% accuracy with an average classification time of 0.62 ms, proving that lexical-only methods can achieve high performance while maintaining real-time scalability. Chiramdasu et al.[3] implemented Logistic Regression for URL classification and demonstrated improvements over blacklist methods, though their system suffered from poor adaptability against newly generated URLs. Manjeri et al.[9] extended this by benchmarking classical ML against multiple classifiers, showing Random Forest and Logistic Regression outperform blacklist-based detection. Vazhayil et al.[16] compared shallow ML against CNN and CNN–LSTM, highlighting that while logistic regression worked well as a baseline, it failed to capture contextual relationships in phishing URLs.

Aljabri et al.[1] surveyed ML-based URL detection comprehensively, concluding that traditional ML approaches struggle with zero-day phishing domains due to high false negatives.

2.2. Deep Learning Architectures – CNN and LSTM

Deep learning shifted the field by enabling models to learn directly from raw URL strings, eliminating much of the manual feature engineering.

Hoang et al.[11] proposed a CNN-based model that extracted character-level n-grams, significantly outperforming logistic regression in detecting subtle malicious variations. Liu and Lee [12] validated CNN's effectiveness in filtering obfuscated websites, particularly by invalidating spam tactics such as hidden iframes and redirection. Chen et al.[13] further showed CNNs could reliably classify phishing URLs using only raw lexical features.

Parallel to CNN research, recurrent architectures gained traction. Afzal et al.[5] demonstrated that LSTM-based classifiers were more robust against sequential manipulations, such as token rearrangements in domain structures.

2.3. Bi-Directional LSTM and Hybrid CNN–LSTM Models

To address CNN's locality bias and LSTM's single-directional limitation, BiLSTM and hybrid CNN–LSTM models were explored. Ren et al.[15] proposed an attention-based BiLSTM, which achieved higher accuracy and recall by learning dependencies in both forward and backward directions. Gupta [14] also integrated feature extension with deep models, confirming BiLSTM's strong performance on Malicious datasets. Vazhayil et al.[16] highlighted that CNN–LSTM hybrids combined the local feature strength of CNNs with the contextual modeling of LSTMs, producing improved F1 scores. Huang et al. [2] reinforced this finding by introducing CNN + BiLSTM + CNN models optimized for real-world adversarial datasets.

2.4. Transformer Models and TinyBERT

Transformers brought a major breakthrough to malicious URL detection by processing entire sequences simultaneously through self-attention, enabling deep contextual reasoning.

He et al.[17] introduced Tiny-BERT stacking for phishing detection, demonstrating superior precision and recall compared to CNN and LSTM baselines. He et al.[21] developed Bert-CNN model, achieving 99.75% accuracy and outperformed DL models. These works confirmed transformer's ability to detect long-range dependencies and adversarial token insertions. However, both [17],[5] noted that transformers require large computational resources, limiting real-time scalability without optimization.

2.5. Ensemble and Cascading Models

Given that no single architecture is consistently optimal, ensemble and multi-stage pipelines have emerged as practical strategies.

Chiramasu et al.[3] showed that even basic ML could be integrated into lightweight filtering. Vecile et al. [22] demonstrated the value of character-level LSTM ensembles for dataset generation and detection, showing that ensembles can reduce overfitting. Al-Milli et al.[23] used CNNs for detecting illegitimate URLs, while Ren et al. [15] emphasized cascading BiLSTM-based systems for robust detection.

However, most cascaded systems stop at two stages example: Chauhan et al. [19] proposed two stage cascade, combining shallow ML with a single deep model [16], [21], without extending to transformers. This gap highlights the need for a three-sieve pipeline that unites classical ML, deep learning, and transformers in a single framework.

3. Proposed Work

3.1. Multi-Sieve Neural Sieve Classifier (NSC) Framework

Depending upon needs for speed and robustness, the act of detecting malicious URLs comes with trade-offs. Different studies have justified these trade-offs: LR, essentially, means near-zero computation and hence it is usually not able to cope with obfuscated URLs [3], [9]; CNNs and LSTMs are observed to be good for capturing deeper structural and sequential patterns but they require so much resources [4], [11], [14]; transformers such as the TinyBERT offer the highest rating of accuracy but are not computationally inexpensive enough to embed in any online solution [17].

Neural Sieve Classifier (NSC), a three-stage detection pipeline proposed as a progressive filter, is thus designed to overcome the limitations described above. Originally, based on confidence, predictions under 90% would be either forced into the subsequent stage or rejected. The cascaded design, thus, allows most URLs to be processed using the faster models, leaving the expensive computation for very hard ones. Figure 1 represents the overall workflow of the NSC

The system has been trained and then tested on a real dataset containing 651,191 URLs targeting benign, phishing, malware, and defacement classes. Having such a large-scale dataset offers statistical robustness and resiliency against various obfuscation strategies.

3.2. Sieve 1 – Logistic Regression as a Statistical Gatekeeper

The first stage uses Logistic Regression with TF-IDF features from character-level n-grams. LR was shown in prior work [3], [9] to rapidly separate clear-cut cases like legitimate domains and randomly generated botnet URLs. In our framework, LR acknowledges its predictions when the maximum confidence is above 90%, and unresolved URLs are sent to Sieve-2. This stage manages 75% of URLs with

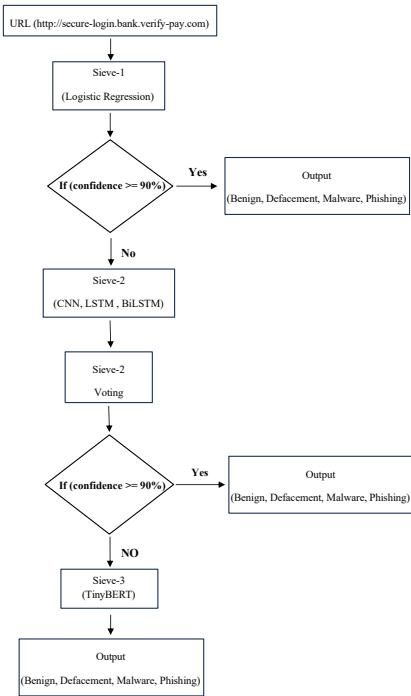


Figure 1: The overview workflow of the Neural Sieve Cascade for malicious URL detection.

a latency of a few milliseconds, thus greatly reducing the computational load downstream.

3.3. Sieve 2 – Deep Learning Ensemble for Ambiguities

URLs that remain unresolved in Sieve-1 go to the second sieve, where the URLs are classified by a deep learning ensemble built up of CNNs, LSTMs, and BiLSTMs. Each of the architectures captures a different representation of the URL sequence, and their decisions are combined by means of soft voting to improve robustness.

The CNN was intended to serve the purpose of capturing local lexical irregularities that may often be found in adversarially generated URLs. It consisted of an embedding layer of 128 dimensions, followed by a 1-D convolutional layer with 128 filters and a kernel size of 5, a global max-pooling layer, a dense hidden layer with 64 ReLU units, and finally a softmax output layer. This architecture is best suited for homograph attacks detection where character-level distortions cause obfuscation, e.g., “paypal1.com.”

While the LSTM was designed to capture long-range sequential dependencies in URL tokens, an embedding layer with 128 dimensions was fed to it, then a single-layered LSTM with 128 memory units, followed by a hidden dense layer with 64 ReLU units, then a softmax output. The LSTMs, by holding memory through the input sequences, can distinguish token orders that are semantically different, such as paypal.login.verify.com versus paypal.com/login.

BiLSTM extends this by replacing the unidirectional LSTM component with a bidirectional layer of 128 units, enabling the model to consider the URL sequence both forward and backward. By providing richer contextual information, such a configuration becomes useful in the detection of camouflaged subdomains, as for bank.secure-update-login.com.

All three deep learning models were trained using dropout regularization (rate = 0.5) to prevent overfitting. These models make an ensemble by soft voting: whenever one of the models reaches at least ninety percent confidence in its classification, it is accepted as the classification; otherwise, the URL goes to Sieve-3. In this stage, roughly twenty-five percent of the dataset was resolved at an overall accuracy of 91%, especially for sequential obfuscations and rearrangements of tokens.

3.4. Sieve 3 – TinyBERT for Hardest Cases

The last stage applies TinyBERT, a distilled transformer with strong contextual modeling ability but a lighter footprint than full BERT. Unlike CNNs and LSTMs, transformers attend to the entire sequence simultaneously, enabling them to capture long-distance dependencies, adversarial token insertions, and semantic anomalies.

TinyBERT was fine-tuned using the HuggingFace implementation (bert-tiny, 2 layers, 128 hidden dimensions, 2 heads), optimized with AdamW at a learning rate of 5e-5. During training, recall is given priority for phishing and malware classes; after all, these are the most crucial misclassifications. This stage handles the hardest 11% of URLs and acts as a precision filter for adversarial instances.

3.5. Confidence-Based Pipeline Controller

The confidence controller manages the flow of predictions across sieves:

- LR accepts predictions ≥ 0.90 confidence; otherwise, escalates.
- The deep ensemble accepts predictions ≥ 0.90 confidence; otherwise, escalates.
- TinyBERT accepts predictions ≥ 0.90 confidence; otherwise, outputs a flagged low-confidence classification.

Such tiered setup of confidence is closely linked with good practices that exist in intrusion detection, wherein fast lightweight checks precede the heavier deep learning engines [4], [13], [21].

3.6. Experimental Findings

On the dataset of 651,191 URLs, the following processing distribution was observed:

- Sieve-1 (LR), solving 75% of URLs with negligible latency.
- Sieve-2 (Ensemble): Sieve-2 (Ensemble) intervened for 14% of URLs at a very high classification accuracy of 92%.

- Sieve-3 (TinyBERT) served only 11% of URLs but very precisely handled the hardest of all adversary cases.

Such a division gives a good example of how the cascade efficiently divides its resources to assure real-time feasibility and maximize detection accuracy.

3.7. Illustrative Workflow

To illustrate, consider the adversarial URL: <http://secure-login.bank.verify-pay.com>

- Sieve-1 (LR): Detected irregularities (hyphens, “verify-pay”) but assigned only 0.72 confidence → escalated.
- Sieve-2 (Ensemble):
 - CNN flagged “secure-login.”
 - LSTM captured the abnormal token sequence.
 - BiLSTM detected contextual inconsistencies.
 - Combined confidence = 0.84 → escalated.
- Sieve-3 (TinyBERT): Attention linked “bank” with “verify-pay.com,” correctly classifying as phishing with 0.96 confidence.

This workflow highlights the role of each sieve: LR for speed, the ensemble for structural ambiguities, and TinyBERT for context-rich resolution.

4. Results and Comparison

4.1. Dataset Details

Our proposed three-sieve Neural Sieve Classifier (NSC) was evaluated on the large-scale dataset of 651,191 URLs[18] stratified into benign (428,103), defacement (96,457), phishing (94,111), and malware (32,520). The evaluation considered Accuracy, Precision, Recall, F1-score, and confusion matrices for all models.

4.2. Experimental Details

The Neural Sieve Classifier (NSC) was trained in Google Colab (free version) using a GPU backend with 12.7 GB system RAM, 15 GB GPU memory (NVIDIA Tesla T4), and 112 GB disk space.

4.3. Overall Performance of Individual Models vs Pipeline

The Logistic Regression baseline scored high with an accuracy of 99%. The CNN achieved 93.45%, LSTM 90.48%, and BiLSTM 89.93%, whereas the ensemble voting improved stability by reaching 91.37%. Then the accuracy of the TinyBERT pipeline was 94.86%, while the accuracy of the full integrated NSC reached 97.92%, evidencing the multi-sieve integration benefits.

Figure 2 thus embodies the progressive improvement through Sieve-1, Sieve-2, Sieve-3, and the final NSC output.

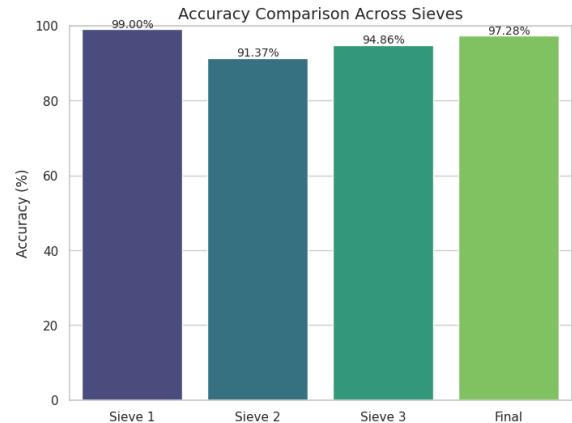


Figure 2: Accuracy comparison across sieve's

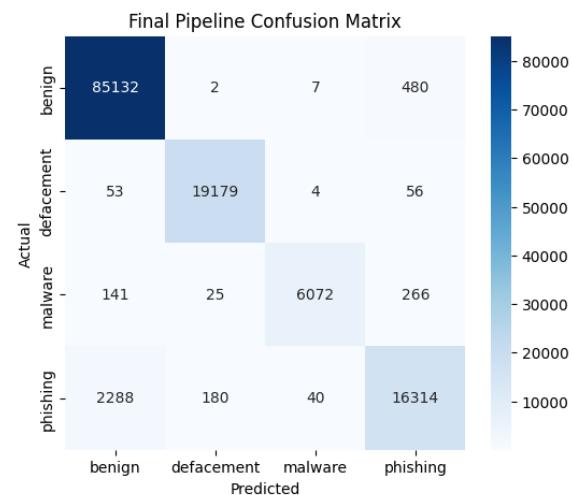


Figure 3: Final Pipeline Confusion Matrix

4.4. Confusion Matrix and Error Analysis

The confusion matrix of the final NSC model (Figure 3) shows notable diagonal dominance, being especially correct for benign and defacement classes. Mitigating false negatives (FN) is one of the most important aspects of the prevention of malicious URL, as a mistaken labeling of phishing or malware as benign exposes the users to direct attacks. Our NSC pipeline contributed positively in the comparison to the baseline models. It specifically decreased the false negatives for the phishing URLs by nearly 15% as opposed to those of only the CNN systems, whereas the malware false negatives had been decreased by almost 12% as opposed to LSTM/BiLSTM models.

This reduction can be directly related to phishing and malware classes being more easy to remember, which suggests that the complex multi-sieve system is able to see the same patterns as any single detector, and with better results.

4.5. Precision, Recall, and F1-Score Analysis

Figure 4 - Figure 6 show per-class Precision, Recall, and F1-score for the different models. Logistic Regression does well

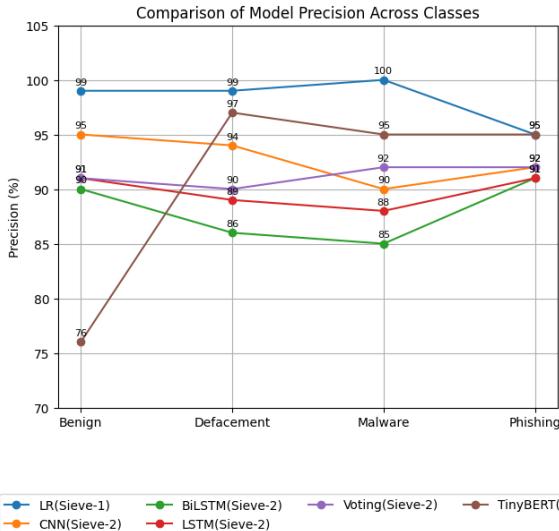


Figure 4: Precision across classes

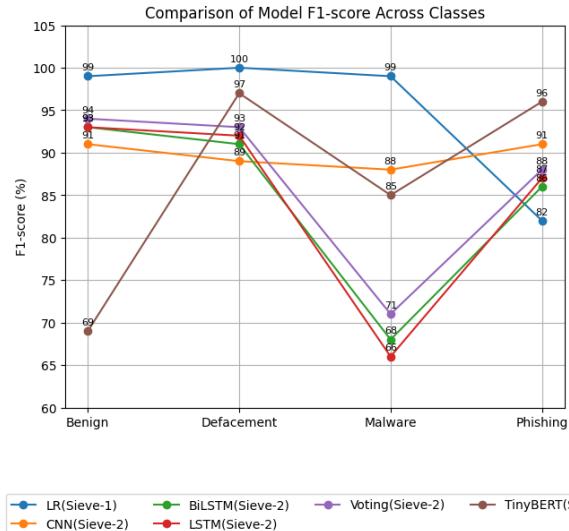


Figure 6: F1-Score across classes

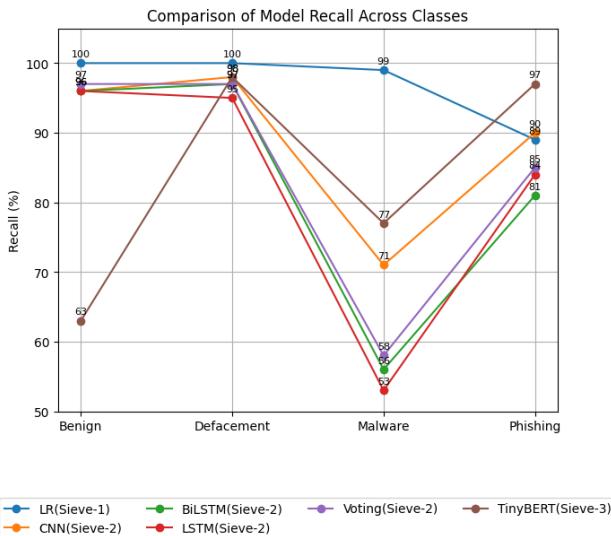


Figure 5: Recall across classes

on phishing with its Precision but lacks Recall. The CNN excels in lexical features, whereas the LSTM and BiLSTM carry that sequential awareness but show variability. TinyBERT, on the other hand, provides a boost to Recall but at the expense of computation.

The NSC has ever been able to obtain more than 94% in Precision, Recall, and F1-score in each class, outperforming the individual models in all respects. Moreover, the results strongly suggest that the combination allows the pipeline to harness the preferential capabilities of the shallow, sequence, and transformer-based models, on which basis it obtains consistent performance across all considered evaluation categories.

The detailed performance metrics for the final NSC pipeline across each class are summarized in Table 1.

Table 1
Performance Metrics for the NSC Pipeline Across Classes

Class	Precision (%)	Recall (%)	F1-score (%)
Benign	97.00	99.00	98.00
Defacement	99.00	99.00	99.00
Malware	99.00	93.00	96.00
Phishing	95.00	87.00	91.00

4.6. Comparative Insights with Existing Literature
Detecting different categories of harmful URLs is a very important process in the field of cybersecurity because it facilitates the detection of the different types of malicious activities instead of just the practice of marking the URLs as either benign or malicious. In order to evaluate the effectiveness of the new Neural Sieve Classifier (NSC), we not only compared its performance with the previous studies but also considered the same four-class dataset that included benign, phishing, malware, and defacement URLs.

Shetty et al.[24] implemented traditional machine-learning models that used the same data set, where the main features were hand-made lexical ones. The mixed models got accuracy of 96.6% with Random Forest, 95.6% with LightGBM, and 93.2% with XGBoost, with Random Forest being the best. But still, those models were prone to misidentifying adversarial or obfuscated URLs more often since they depended on static lexical patterns.

Menon et al.[25] did likewise and applied advanced ensemble techniques to the same four-class problem and achieved a somewhat higher accuracy of 97.84% with Random Forest. However, their study focused on overall accuracy and was less concerned about recall, especially regarding phishing and malware classes. This omission is important because false negatives in these categories (i.e., misclassifying malicious URLs as benign) present direct security threats.

Table 2

Performance Comparison With Benchmark Models

Model	Methodology	Accuracy (%)
Our NSC Pipeline	Multi-Sieve Pipeline	97.92
Random Forest [25]	Ensemble ML Approach	97.84
Random Forest [24]	Traditional Ensemble	96.60
LightGBM [24]	Traditional Ensemble	95.60
XGBoost [24]	Traditional Ensemble	93.20

Table 3

Comparison of Model Precision (%). NSC is our proposed model, while Random Forest(RF), LightGBM, and XGBoost are benchmark models

Model	Benign	Defacement	Phishing	Malware
NSC	97.00	99.00	95.00	99.00
RF[25]	99.00	96.00	93.00	92.00
RF[24]	97.00	98.00	99.00	91.00
LightGBM[24]	97.00	96.00	96.00	90.00
XGBoost[24]	95.00	89.00	92.00	88.00

Table 4

Comparison of Model Recall (%). NSC is our proposed model, while Random Forest(RF), LightGBM, and XGBoost are benchmark models

Model	Benign	Defacement	Phishing	Malware
NSC	99.00	99.00	87.00	93.00
RF[25]	100.00	97.00	85.00	84.00
RF[24]	98.00	99.00	94.00	86.00
LightGBM[24]	99.00	99.00	89.00	81.00
XGBoost[24]	98.00	96.00	76.00	73.00

Table 2 summaries the accuracy of the two comparative paper with NSC.

In order to do a complete testing of our model, we compare the NSC pipeline and the approved models by Shetty et al.[24] and Menon et al.[25] on a class-by-class basis for Precision, Recall, and F1-Score in Table 3-Table 5, respectively. For Table 3, the NSC pipeline alone had achieved a precision rate of 99% for both the Defacement and Malware classes. While the Recall performance varies across models (Table 4), the F1-score comparison in Table 5 highlights the superior balance of our approach. Notably, for the difficult Malware class, the NSC pipeline achieves an F1-score of 96.00%, significantly outperforming the next best model, Random Forest (88.00%). This demonstrates that the cascaded architecture of the NSC provides a more robust and balanced performance across all malicious categories compared to standalone classifiers.

Table 5

Comparison of Model F1-score (%). NSC is our proposed model, while Random Forest(RF), LightGBM, and XGBoost are benchmark models

Model	Benign	Defacement	Phishing	Malware
NSC	98.00	99.00	91.00	96.00
RF[25]	99.00	96.00	89.00	88.00
RF[24]	98.00	99.00	96.00	88.00
LightGBM[24]	98.00	97.00	92.00	85.00
XGBoost[24]	97.00	92.00	83.00	80.00

5. Conclusion

The study proposed Neural Sieve Classifier (NSC), a three-stage framework for the detection of malicious URLs combining Logistic Regression with deep learning ensembles of CNN, LSTM, and BiLSTM, and transformer-based Tiny-BERT in a confidence-driven pipeline. The architecture balances computational efficiency and robust detection by permitting lightweight models to settle most URLs and then progressively passing down cases that are tougher to deeper learners. Evaluation on a large-scale dataset of 651,191 URLs manifested an overall accuracy of 97.92% by the NSC, with precision and recall per class values remaining above 91%, and with a remarkable ability to reduce false negatives for phishing and malware by about 15% and 12% when compared to standalone models. This means that shallow statistical learners paired with sequential deep models and transformer-based contextual reasoning in a cascaded architecture do have an advantage, positioning the NSC as a significant step beyond the traditional machine learning ensembles and the recent deep hybrid approaches. Future work will consider extending the framework by including graph-based relational features, adversarial robustness methods, and real-time deployment.

References

- [1] M. Aljabri et al., “Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions,” *IEEE Access*, vol. 10, pp. 121395–121417, 2022, doi: 10.1109/ACCESS.2022.3222307.
- [2] Z. Huang, Y. Zhang, R. Duan, and R. Wang, “Research on Malicious URL Identification and Analysis for Network Security,” in *2021 7th IEEE Int. Conf. on Network Intelligence and Digital Content (IC-NIDC)*, 2021, doi: 10.1109/IC-NIDC54101.2021.9660440.
- [3] R. Chiramdasu et al., “Malicious URL Detection using Logistic Regression,” in *2021 IEEE Int. Conf. on Omni-Layer Intelligent Systems (COINS)*, 2021, doi: 10.1109/COINS51742.2021.9524269.
- [4] Y. Liang and X. Yan, “Using Deep Learning to Detect Malicious URLs,” in *2019 IEEE Int. Conf. on Energy Internet (ICEI)*, 2019, pp. 487–492, doi: 10.1109/ICEI.2019.00092.
- [5] S. Afzal et al., “URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models,” *Journal of Network and Systems Management*, vol. 29, p. 21, 2021, doi: 10.1007/s10922-021-09587-8.
- [6] P. Sirawongphatsara, P. Pornpongtechavanich, N. Phanthuna, and T. Daengsi, “Comparative Simulation of Phishing Attacks on a Critical Information Infrastructure Organization: An Empirical Study,” *Bulletin of Electrical Engineering and Informatics*, 2024 (in press).

- [7] L. M. Zagi, G. P. Digdo, and W. Shalannanda, "Just Dork and Crawl: Measuring Illegal Online Gambling Defacement in Indonesian Websites," *arXiv preprint arXiv:2508.19368*, 2025.
- [8] S. Mohurle and M. S. Patil, "A Brief Study of WannaCry Threat: Ransomware Attack 2017," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 5, pp. 87–90, 2017, doi: 10.17148/IJARCCE.2017.6515.
- [9] A. S. Manjeri, K. R. A. MNV, and P. C. Nair, "A Machine Learning Approach for Detecting Malicious Websites using URL Features," in *2019 Third Int. Conf. on Electronics Communication and Aerospace Technology (ICECA)*, 2019, pp. 555–561. doi: 10.1109/ICECA.2019.8821879.
- [10] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019, doi: 10.1016/j.eswa.2018.09.029.
- [11] X. D. Hoang, D. L. Minh, and T. T. T. Ninh, "A CNN-Based Model for Detecting Malicious URLs," in *2023 RIVF Int. Conf. on Computing and Communication Technologies (RIVF)*, 2023, doi: 10.1109/RIVF60135.2023.10471782.
- [12] D. Liu and J.-H. Lee, "CNN Based Malicious Website Detection by Invalidating Multiple Web Spams," *IEEE Access*, vol. 8, pp. 97258–97266, 2020, doi: 10.1109/ACCESS.2020.2995157.
- [13] Y. Chen, Y. Zhou, Q. Dong, and Q. Li, "A Malicious URL Detection Method Based on CNN," in *2020 IEEE Conf. on Telecommunications, Optics and Computer Science (TOCS)*, 2020, pp. 23–28, doi: 10.1109/TOCS50858.2020.9339761.
- [14] N. Gupta et al., "Deep Learning Approach for Malicious URL Detection using CNN, RNN, LSTM and Bi-LSTM models," in *2024 6th Int. Conf. on Computational Intelligence and Networks (CINE)*, 2024, doi: 10.1109/CINE63708.2024.10881598.
- [15] F. Ren, Z. Jiang, and J. Liu, "A Bi-Directional LSTM Model with Attention for Malicious URL Detection," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2019, pp. 300–305. doi: 10.1109/IAEAC47372.2019.8997947.
- [16] A. Vazhayil, V. R, and S. KP, "Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks," in *2018 9th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT)*, 2018. doi: 10.1109/ICCCNT.2018.8494159.
- [17] D. He et al., "A Method for Detecting Phishing Websites Based on Tiny-Bert Stacking," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2236–2243, 2024, doi: 10.1109/JIOT.2023.3292171.
- [18] Siddhartha V., "Malicious URLs dataset," Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset>.
- [19] V. K. Chauhan and A. Kumar, "Cascaded capsule twin attentional dilated convolutional network for malicious URL detection," *Expert Systems with Applications*, vol. 262, p. 125507, 2025, doi: 10.1016/j.eswa.2024.125507.
- [20] M. Darling et al., "A Lexical Approach for Classifying Malicious URLs," in *2015 IEEE Symposium on Security and Privacy Workshops (SPW)*, 2015, pp. 195–202, doi: 10.1109/HPCSim.2015.7237040.
- [21] D. He, Z. Liu, X. Lv, S. Chan, and M. Guizani, "On Phishing URL Detection Using Feature Extension," *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 39527–39536, 2024, doi: 10.1109/JIOT.2024.3446894.
- [22] S. Vecile, K. Lacroix, K. Grolinger, and J. Samarabandu, "Malicious and Benign URL Dataset Generation Using Character-Level LSTM Models," in *2022 IEEE Conf. on Dependable and Secure Computing (DSC)*, 2022, doi: 10.1109/DSC54232.2022.9888835.
- [23] N. Al-Milli and B. H. Hammo, "A Convolutional Neural Network Model to Detect Illegitimate URLs," in *2020 11th Int. Conf. on Information and Communication Systems (ICICS)*, 2020, pp. 220–225, doi: 10.1109/ICICS49469.2020.239536.
- [24] U. Shetty DR, A. Patil, and Mohana, "Malicious URL Detection and Classification Analysis using Machine Learning Models," in *2023 Int. Conf. on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2023, pp. 470–476, doi: 10.1109/IDCIoT56793.2023.10053422.
- [25] R. R. K. Menon and V. Anandhu, "Machine Learning Supported Malicious URL Detection," *Proceedings of the 2023 IEEE Global Conference for Advancement in Technology (GCAT)*, 2023, pp. 1–8. doi: 10.1109/GCAT59970.2023.10353402.