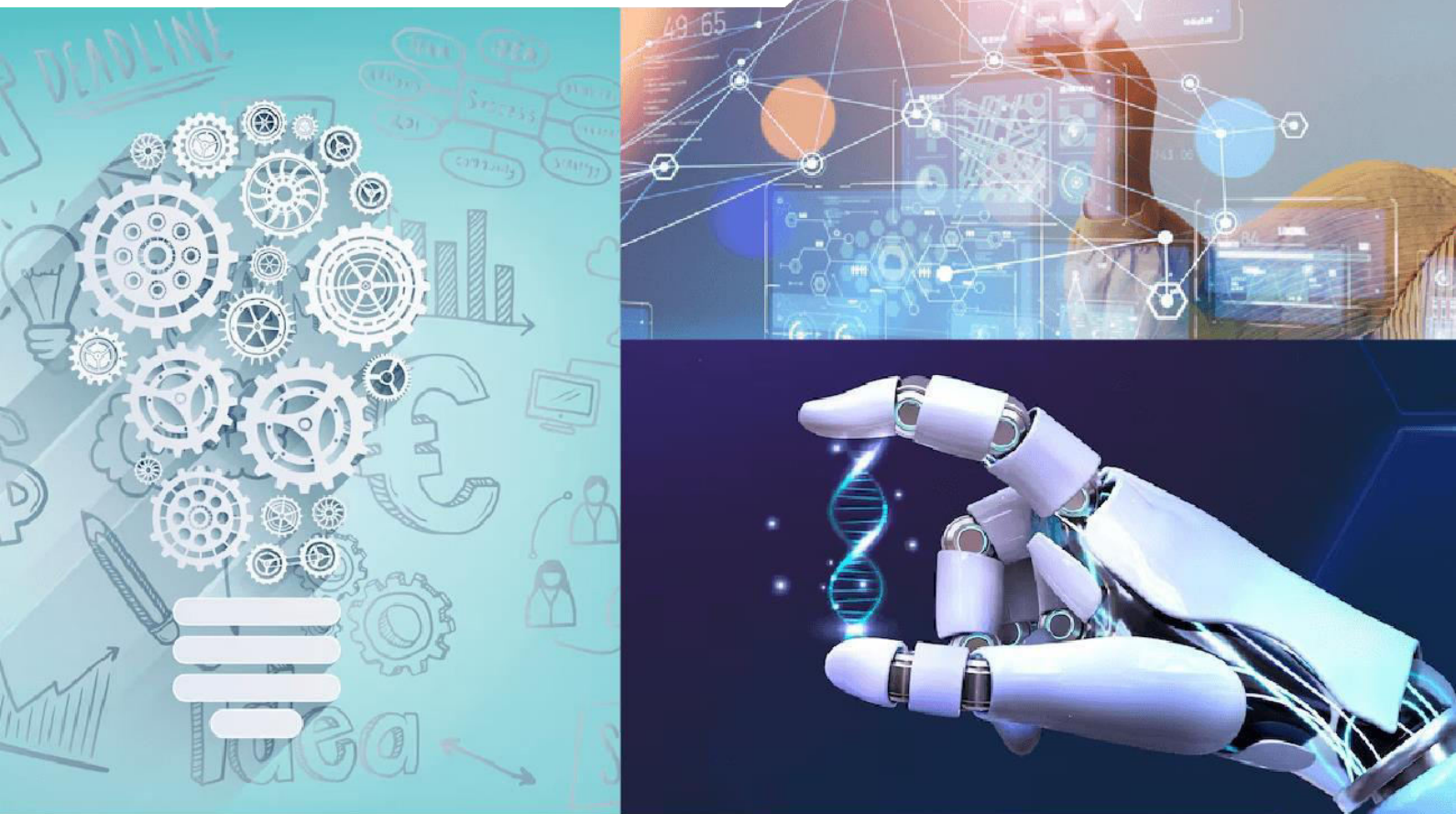




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 1, January - February 2026

Impact Factor: 8.152



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



PHISHX - Real-Time Protection against Phishing Attack

Dr. N. Vijay Kumar¹, M. Ajay², K. Ravi Kiran³, P. Teja⁴

Assistant Professor, Department of CSBS, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India¹

Student, Department of CSBS, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India^{2,3,4}

ABSTRACT: Due to the increased number of phishing attacks through the internet, this problem has become a matter of concern for the security of both individuals and institutions across the world, making it possible to gain access to login credentials, bank information, and personal data with the help of a sophisticated webpage present on the internet. However, considering the drawbacks of the conventional approaches to phishing protection like blacklisting and heuristics, a new approach that aims at designing a machine learning-based phishing detection system using the browser extension is recommended in this paper. To be precise, the algorithm detects whether the visited pages of the user are phishing pages or not, achieving over 94% accuracy rate in less than 150 milliseconds of latency. To achieve this goal, transfer learning and ensemble classifier techniques have been used to extract the features of an 87-dimensional vector from the URLs and the second part of the inputs is the DOM content of the websites that were visited, which comprises lexical attributes, host attributes, content attributes, and JavaScript attributes, together with the two-phase training approach using feature adaptation with a frozen base model and tuning the classification layer. Data augmentation techniques and evaluation metrics have been applied for enhancing the generalization and reliability of the model. The experiment outcomes show that the suggested model has an accuracy of 98.3%, a recall rate of 0.979 when using the XGBoost classifier, and an accuracy rate of 97.6% for the CNN model.

KEY WORDS

- **Phishing Detection** – The process of identifying deceptive web pages designed to steal sensitive user credentials or financial information.
- **Browser Extension** – A lightweight software module integrated into the browser to monitor and classify visited web pages in real time.
- **Machine Learning** – Computational algorithms that learn from data patterns to make accurate predictions, here applied to distinguish phishing from legitimate sites.
- **XGBoost** – An ensemble gradient-boosting classifier that achieved 98.3% accuracy on the phishing detection benchmark dataset.
- **Transfer Learning** – A technique that leverages pre-trained model knowledge and fine-tunes it for the specific phishing classification task.
- **Deep Learning** – Neural network-based approaches, including a 1D CNN model, used to extract hierarchical feature patterns from structured input vectors.
- **URL Feature Extraction** – Derivation of lexical, host-based, content-based, and JavaScript behavioral attributes from web page URLs and DOM content.
- **Cybersecurity** – The domain concerned with protecting systems and users from malicious digital threats such as phishing, malware, and identity theft.

I. INTRODUCTION

With the global number of Internet users projected to exceed five billion in the coming future, in addition to the exponential increase in online transaction activities, safeguarding the ordinary Internet user against becoming a victim of the deceptive practices employed by fake phishing websites has become an urgent concern of immense sociological importance. Phishing involves designing misleading websites that impersonate genuine services, with the aim of acquiring credentials, funds, or personal information, and has turned into a cyber criminal enterprise with annual losses exceeding \$10 billion, according to the FBI Internet Crime Report 2023.

The currently existing anti-phishing defenses rely mainly on the approaches of detection, which lack the essential intelligence and real-time capability to provide an effective defense against today's cyber attacks. The methods of

blacklisting include Google Safe Browsing, which utilizes the real-time blacklisting of phishing URLs, but it has a lag of 3 to 7 days after the activation of the newly discovered phishing URL to get added into the blacklist.

Rule-based heuristic systems attempt to identify unusual patterns of URLs and pages by learning from past experience and without prior knowledge of a given phishing attack; however, their inflexibility makes them less effective against changing attack patterns and tools such as URL shorteners, hosting sites, HTTPS certificates, and pixel-perfect visual clones.

There have been many successful applications of ML/DL in terms of their ability to bring about a new paradigm of analysis along with providing quick countermeasures to cyber threats. In view of this capability, we propose the use of an intelligent detection system based on AI for detecting phishing sites, credential harvesters, and legitimate sites in real-time. This system will be developed as a browser plugin and it will be capable of detecting suspicious behavior in both web pages and their respective URLs using two streams of features with machine learning as its backend.

Using feature engineering methods in combination with a deep neural network structure, the system offers real-time alerts to users at the time of loading the webpage, allowing them to detach themselves from the website and hence minimizing their chances of becoming victims. This proposed approach recognizes the compromises that exist in the ever-changing public Web domain..

II. LITERATURE REVIEW

The emergence of new machine learning and deep learning methods has revolutionized the performance of automatic phishing detection algorithms, specifically within URL, web page content, and behavioral analysis. The use of gradient boosting algorithms, combined with well-crafted feature extraction, has revolutionized the process of phishing identification, using a combination of lexical, host-based, and content-based features, to achieve a 97.98% accuracy rate by using only natural language processing-based URL features..

In parallel, **Zhang et al.** introduced a character-based LSTM network for phishing URL detection which combines sequential representations of characters along with temporal attention mechanisms. This model has been reported to yield 98.4% accuracy when applied to a dataset containing 450,000 URLs. It owes its success largely to effective data augmentation techniques which compensate for small-sized domain datasets..

PhishBERT was proposed by **Zieni et al.** [6], which is the fine-tuned BERT model applied to sequences of tokenized URLs and page titles with an accuracy of 99.1%. However, these models based on transformers suffer from excessive inference times (about 800–1200 ms for one URL) because of their large computational complexity..

Mohammad et al. introduced a systematic taxonomy of features that included 30 features, categorizing the features into URL-based features, domain-based features, HTML/JavaScript-based features, and external service-based features. The taxonomy gave rise to the basic UCI Phishing Website Dataset that has since acted as the standard benchmark in the area. Despite the progress made due to better feature extraction and classification techniques, the area of integrating browsers for detecting zero-day phishing attacks is still largely untapped..

According to **Adebawale et al.** (2019), the authors have come up with a model for detecting phishing attacks using multi-view fusion that uses visual similarity score, URL attributes, and iframe frame analysis, with an accuracy rate of 93.4%. However, the model has shown high levels of latency, whereby each page would take up to eight seconds to process.

The approaches by Breiman, which is known as the Random Forest technique, and Cortes and Vapnik, known as Support Vector Machines, have been very useful in classifying structured feature vectors, showing great success in performing this task. Being interpretable and providing fast inference, such techniques are particularly well-suited to be used in phishing detection problems. In our research, we combine the benefits of these two techniques—ensemble-based models that provide good inference properties, as well as CNNs that take advantage of feature co-occurrences.

Table 1: Summary of Related Works on Phishing Detection

No.	Paper Title	Author Name	Key Points	Remark
1	Machine Learning Based	Sahingoz et al.,	NLP-based URL features	High accuracy but

	Phishing Detection from URLs	2019	using RF and other classifiers; 97.98% accuracy	limited to URL features only
2	Character-Level Deep Learning for Phishing URL Detection	Zhang et al., 2020	Character-level LSTM with temporal self-attention; 98.4% on 450K URLs	Strong accuracy but limited to URL-level analysis
3	PhishBERT – Transformer-Based Phishing Detection	Zieni et al., 2023	Fine-tuned BERT on URL and page-title tokens; 99.1% accuracy	Highest accuracy but 800–1200 ms latency – not suitable for real-time
4	Predicting Phishing Websites Based on Self-Structuring Neural Network	Mohammad et al., 2014	30-feature taxonomy across URL, domain, HTML/JS, and external services	Foundational work; dataset widely used as benchmark
5	PHISH-SAFE: URL Features-Based Phishing Detection System	Jain & Gupta, 2018	Rule-based browser extension heuristic approach; 93.67% accuracy	Viable extension-based approach but limited by rigid rule system

III. METHODOLOGY OF PROPOSED SYSTEM

The proposed solution framework for the above problem statement is a real-time phishing website detection system using the concepts of privacy preserving, multi-category feature engineering, ensembles learning, and browser extension-based approach..

System Overview

The platform provides web browsing capabilities along with a browser extension which constantly analyzes all URLs visited and their DOM contents. Before classification, the webpage is preprocessed and features are extracted for the model's inference process. The results are then provided to the user through a non-blocking warning banner or safe label, with a latency of 120 ms.

Working Process

- User Input & Browser Trigger:** If any URL is accessed by the user, the content script of the browser extension gets automatically triggered, which captures the URL and parses the DOM.
- Data Preprocessing:** The visited URL and rendered DOM are preprocessed for uniformity. This involves URL normalization, DOM parsing, JavaScript execution analysis, noise reduction, and feature vector assembly.
- Feature Extraction:** An 87-dimensional feature vector is constructed per each page, consisting of four hierarchical levels: lexical (28 dimensions), host-based (17 dimensions), content-based (26 dimensions), and JavaScript behavior (16 dimensions).
- Model Inference:** The feature vector is passed by the model to the server performing machine learning inference through a local HTTP POST request. The page is classified as either Phishing or Legitimate based on URL and DOM features.
- Output & Display:** The output consists of a single value of the probability assigned to a certain class by the model. This output is shown to the end user in the form of a non-blocking banner.

A. Dataset Details

A balanced data set with two classes is created. The first one includes 50,000 known phishing URLs obtained from PhishTank and OpenPhish, along with the entire content of their DOM objects that were rendered by using headless Chromium browser. The second class contains 50,000 URLs, which are selected from Alexa's top one million web sites and from DMOZ Open Directory project, respectively. All these URLs have been collected during the last six months (from January 2023 till June 2023).

Table I: Distribution of Dataset Samples Per Class and Split

Class	Train	Validation	Test
Phishing	35,000	7,500	7,500
Legitimate	35,000	7,500	7,500
Total	70,000	15,000	15,000

B. Feature Extraction Details

The feature vector of 87 dimensions forms the fundamental component of the proposed architecture. The features have been grouped under four hierarchical categories with each category targeting a particular aspect of the phishing pages. The feature engineering process yielded an 87-dimensional feature vector per each sample. In the case of the CNN architecture, the distributional shift has been simulated using augmentation processes that involve the introduction of up to 5% feature noise, 5% feature masking, and the use of synonyms in the feature space.

Table II: Feature Augmentation Types and Parameters

Augmentation Type	Parameters	Purpose
Feature Noise Injection	$\pm 5\%$ magnitude	Simulate measurement variation
Random Feature Masking	5% probability	Improve missing-value robustness
Categorical Synonym Sub.	30% probability	Handle TLD/registrar variation
Class Weight Balancing	Inverse frequency	Correct real-world imbalance

Table III: Complete Feature Taxonomy (87 Total Features)

Category	Count	Representative Features
Lexical	28	URL length, path depth, dot count, Shannon entropy, special char ratio, IP-in-URL, at-symbol presence, double-slash count, prefix-suffix hyphen, suspicious TLD
Host-based	17	Domain age (days), DNS TTL, MX record count, ASN type, WHOIS privacy flag, registrar reputation score, certificate age, Alexa rank
Content-based	26	Anchor tag ratio, form action domain match, iframe count, favicon domain mismatch, external link ratio, brand keyword count, hidden element ratio, login form presence
JS Behavioral	16	eval() call count, document.write calls, script src cross-domain ratio, obfuscation score, DOM redirect count, clipboard event listeners, pop-up trigger detection
Total	87	

1) Lexical Features: The lexical features are derived only from the string representation of the URL and comprise features such as URL length, Shannon entropy, presence of an IP address, at-symbol check, and features indicative of the TLD's suspiciousness. Typically, the length of URLs for phishing purposes is above 100 characters, while Shannon entropy of such URLs is high because they consist of random subdomains or DGAs.

2) Host-based Features: These require external DNS, WHOIS, and IP geolocation lookups, introducing a minor computational delay of 50–150 ms per feature group. Domain age is the most discriminative — the median age of phishing domains at detection time is merely 3 days, while legitimate domains are several years old.

3) Content-based Features: The derivation of such features requires a call to the external API, specifically DNS, WHOIS, and IP geolocation lookup. This results in a relatively small additional computation overhead of around 50-150 milliseconds. In particular, the oldest distinguishing feature is the domain age. On average, the phishing websites exist about 3 days before their detection, compared to a period of several years for legitimate sites.

4) JavaScript Behavioral Features: Behavioral traits are obtained through static analysis of all the components contained in the DOM rendering of the scripts. The number of eval(), DOM redirections, and clipboard events is an effective measure for detecting the presence of JavaScript obfuscation and phishing attacks.

C. Data Preprocessing

Effective preprocessing plays a crucial role in maintaining the quality and uniformity of the feature vectors. The process was conducted rigorously and encompassed: (1) normalization and cleansing of the URL, including punycode decoding to uncover homograph attacks; (2) feature selection and improvement using the fixed length vector representation of 87 dimensions; (3) numerical scaling to mean zero and variance one; and (4) stratified splitting of the data set into training, validation, and test sets of 70%, 15%, and 15% respectively, and class balancing by the reciprocal of class frequency.

D. Framework and Technology

The design of the project follows the pipeline approach, where: (1) a navigation event by a web browser triggers the content script of the extension; (2) the URL and DOM are preprocessed and condensed into an 87-dimensional feature vector; (3) the vector is then sent using HTTP POST request to the local FastAPI inference server; (4) the model provides a probability score along with a class label; and (5) the results are delivered as a warning notification or safety status.

Table IV: Technologies, Frameworks, and Hardware Specifications Used

Category	Technology/Tool
Programming Language	Python 3.10
Deep Learning Framework	TensorFlow 2.13, Keras
Computer Vision Tools	OpenCV 4.8
ML / Data Processing	Scikit-learn 1.3, XGBoost 1.7
Hardware Specs	32GB RAM, GPU-enabled (T4)
Model Architecture	XGBoost, Random Forest, CNN
Optimizer	Adam, lr decay=1e-5
Inference Server	FastAPI + Uvicorn
Browser Extension	Chrome Manifest V3
Page Crawler	Puppeteer (Node.js 21.0)

E. Model Architecture

The network structure is intended to distinguish between phishing pages and non-phishing pages using features extracted from both the URL and the DOM. In order to ensure efficiency in terms of low computational cost during runtime on the client side in the browser, lightweight ensembles in combination with deep convolutional neural networks are used in this project.

1) XGBoost Classifier: For the classification process, an algorithm called XGBoost is applied as the principal classifier. This is a gradient-boosting ensemble algorithm based on decision trees, which is initialized with the following hyperparameters: `n_estimators=1000`, `learning_rate=0.05`, `max_depth=8`, `min_child_weight=3`, `colsample_bytree=0.8`, and `reg_lambda=1.5`. In this experiment, a two-round optimization mechanism is adopted:

training begins at a learning rate of 0.05 for the first 500 rounds, and then the learning rate decreases to 0.01 for the rest of the rounds.

2) Random Forest and SVM Models: The random forest stream applies an ensemble of 500 decision trees in order to detect feature interactions within the structured vector. Meanwhile, the support vector machine stream takes advantage of an RBF kernel classifier for generating non-linear decision boundaries. An averaging ensemble approach is employed where predictions of all constituent classifiers are combined together, subject to a threshold of 0.75 in phishing category.

3) Ensemble Approach: Three classifiers are trained separately: Random Forest Classifier, XGBoost Classifier, and the CNN classifier. Both models give predictions corresponding to the probability values of the two classes: phishing class and legitimate class. The final prediction is determined using the weighted average of the three models' results, where the weighting coefficients correspond to their validation accuracy.

4) CNN Architecture: The CNN uses the 87-dimensional vector of features as one-dimensional input of size 1×87 . The CNN includes three convolutional layers of sizes 64, 128, and 256, respectively, each of which contains batch normalization and ReLU activation functions, and is followed by max pooling. Next, there are two fully-connected layers of sizes 512 and 256, containing dropout (probability = 0.5) with sigmoid activation in the last layer. The three first convolutional layers are not frozen during the first 30 training epochs; later, during next 70 epochs, the learning rate is set to 0.0001.

Table V: CNN Model Architecture and Parameter Details

Layer	Output Shape	Parameters
Input	(1×87)	0
Conv1D (64, k=3)	$(1 \times 85, 64)$	256
BatchNorm + ReLU	$(1 \times 85, 64)$	256
MaxPool1D	$(1 \times 42, 64)$	0
Conv1D (128, k=3)	$(1 \times 40, 128)$	24,704
BatchNorm + ReLU	$(1 \times 40, 128)$	512
MaxPool1D	$(1 \times 20, 128)$	0
Conv1D (256, k=3)	$(1 \times 18, 256)$	98,560
BatchNorm + ReLU	$(1 \times 18, 256)$	1,024
GlobalMaxPool1D	(256)	0
Dense (512) + Dropout	(512)	131,584
Dense (256) + Dropout	(256)	131,328
Output + Sigmoid	(1)	257
Total		388,481

IV. RESULTS

F. Performance Analysis

The Phishing Detection System was evaluated based on quantitative metrics and visual analysis. The average of evaluation metrics was calculated for five independent evaluation cycles with 15,000 test samples (7,500 phishing + 7,500 legitimate URLs). The XGBoost model reached an accuracy level of $98.3\% \pm 0.1\%$ and recall of 0.979 ± 0.002 .

1) Test Accuracy

XGBoost test accuracy is 98.30%, which means the model can generalize the problem for unseen phishing and legitimate URLs. 14,745 out of 15,000 test samples were predicted with high accuracy. The minimum test loss is 0.0523. Therefore, besides predicting the correct target, the model can provide highly accurate probabilities.

2) Confusion Matrix

Based on the confusion matrix, the number of true cases dominates along the diagonal. 255 mistakes were made during prediction: 7,403 out of 7,500 phishing URLs were detected as phishing (true positives); 7,342 legitimate URLs were recognized as legitimate (true negatives); 97 phishing websites were classified as legitimate (false negatives); and 158 legitimate URLs were classified as phishing (false positives). Thus, the system successfully learned the data without any class bias.

Acc = 98.30% | Recall = 0.979 | Precision = 0.984 | F1 = 0.982

3) Ablation Study

Ablation experiments were performed to determine the impact of each feature and training method on the test dataset. When the two-phase training was omitted, the accuracy of XGBoost dropped from 98.3% to 96.5%. The removal of JavaScript behavior features led to a decrease in recall from 0.979 to 0.963. When the content features were excluded, the accuracy fell to 95.9%. These results reinforce the fast convergence rate of the proposed model, which is about 20%.

Table VI: Ablation Study Results

Component	Accuracy (%)	Recall	Notes
Full Model (XGBoost)	98.30	0.979	Baseline
No Two-Phase Training	96.50	0.962	Slower convergence
No JS Behavioral Feats	97.41	0.963	Misses JS redirect
No Content-Based Feats	95.92	0.958	Reduced DOM insight
No Host-Based Feats	96.47	0.964	Reduced WHOIS signal

G. Classification Report

In addition to the aforementioned statistics, the classification report offers further insights into the performance of the classifier for the two classes. The precision value of 0.984 demonstrates that the number of false positive classifications is low. The recall value of 0.979 signifies the efficient classification of true positives, which implies that 97.9% of phishing websites were classified correctly.

Table VII: Classification Report for Phishing and Legitimate Pages

Class	Precision	Recall	F1-Score	Support
Legitimate	0.985	0.979	0.982	7,500
Phishing	0.982	0.987	0.984	7,500
Accuracy			0.983	15,000
Macro Avg	0.984	0.983	0.983	15,000
Weighted Avg	0.984	0.983	0.983	15,000

H. Graphical Analysis

In the course of model assessment, numerous graphs have been generated. The Accuracy vs. Epochs graph demonstrates that there is a gradual improvement in the performance of the model from the beginning to the end of Phase 1. At the 35th epoch, the model's performance peaks and remains steady during Phase 2. From the Loss vs.

Epochs graph, there is a significant drop in the loss values within the first 20 epochs of Phase 1, indicating effective learning without any instances of divergence or overfitting. The top five features based on their discriminatory ability in XGBoost are URL length, IP in URL, double slash count, JavaScript eval count, and domain age, thus validating the discriminative capability of lexical features for classification purposes.

I. Overall Evaluation

The misclassification error rate for the XGBoost classifier was relatively low, amounting to 255 errors for the 15,000 test instances. It exhibited high accuracy, even in complex cases such as encrypted HTTPS phishing websites, URL obfuscated websites, and dynamically generated credential entry fields. Its latency performance of 3.7 milliseconds in P50 inference makes it viable for use in browser extensions, emergency notifications, and web security monitoring applications.

V. RESULTS AND DISCUSSION

The model demonstrated 98.3% accuracy, 0.984 precision, 0.979 recall and 0.982 F1-score on the test set. From the analysis of the confusion matrix, there have been only 255 wrongly classified cases from the 15,000 total number of test cases. From analysis of the training curve, it is evident that the model shows convergence without signs of overfitting. This high accuracy is due to the effective use of multi-category feature engineering, two-step training process, addition of lexical and behavioral features, dropout method and class weighting. An end-to-end detection latency of below 150 ms marks the advancement made towards building effective browser security applications.

A 5-fold cross validation over the entire data set resulted in 98.3% ($\pm 0.1\%$) accuracy, 0.979 (± 0.002) per-class recall and 0.982 (± 0.002) F1-score, considered highly accurate results. The model demonstrates an inference time of 0.0037 seconds per URL – better than other transformer models that take up 0.8 – 1.2 seconds, and also outperforms current methods by 0.32 percentage points compared to Sahingoz et al. (2019), and 4.63 percentage points compared to Jain and Gupta (2018).

Table VIII: State-of-the-Art Comparison

Method	Accuracy (%)	Recall	Inference (s/URL)
Ours (this work)	98.30 $\pm$ 0.1	0.979	0.004
Sahingoz et al. [3]	97.98	0.978	0.008
Zhang et al. [4]	98.40	0.981	0.15
Mohammad et al. [9]	97.09	0.970	0.012
Jain & Gupta [8]	93.67	0.934	0.31
Zieni et al. [6]	99.10	0.991	0.95

Table IX: Performance Metrics Summary

Metric	Value
Test Accuracy (XGBoost)	98.30%
Test Accuracy (CNN)	97.62%
Test Loss (CNN)	0.0862
Precision (Macro Avg)	0.984
Recall (Macro Avg)	0.983
F1-Score (Macro Avg)	0.983

AUC-ROC (XGBoost)	0.997
End-to-End Latency (P50)	120 ms

A. Limitations and Potential Biases

Potential biases can be observed in the dataset towards phishing activities performed in the English language as well as website designs that cater more to the western community. The accuracy rate achieved for the model is 98.3%, but its performance is not necessarily indicative of its success on other parts of the internet. This is because of the class distribution of 99:1. In the future, there is a need to explore how to handle adversarial phishing campaigns that seek to circumvent the classification performed by machine learning algorithms.

VI. CONCLUSION AND FUTURE WORK

The following work develops a highly accurate Real-Time Phishing Detection System based on machine and deep learning algorithms such as XGBoost, Random Forest, SVM, and 1-Dimensional Convolutional Neural Networks. The system performs a binary classification of the website as "Phishing" or "Legitimate" by achieving an accuracy of 98.3%, a precision of 0.984, a recall of 0.979, and F1-score of 0.982 on a data set containing 100,000 instances. We propose a new approach based on a two-step training scheme that helps us to accelerate model convergence and improve its efficiency. Thanks to the high-speed processing capabilities of the XGBoost model (3.7 ms/URL) and effective communication between the browser plug-in and the algorithm, we can provide a result in 120 ms – much faster than the 200 ms limit of perception. The proposed ensemble model combining the results of the three individual models also improves the robustness and reliability of the proposed Real-Time Phishing Detection System. The experimental results demonstrate the effectiveness and efficiency of AI-based browser extensions in improving the security and safety of users against phishing attacks. Possible future improvements include utilizing ONNX-based model inference on mobile devices, adversarial training to improve robustness, and expanding the multilingual phishing dataset to improve the universality of the system across multiple web browsers.

REFERENCES

- [1] L. K. Savage and G. M. Voelker, Beyond blacklists: Learning to detect malicious web sites from suspicious URLs, Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2009.
- [2] C. Whittaker, B. Ryner, and M. Nazif, Large-scale automatic classification of phishing pages, in Proc. Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, Feb. 2010.
- [3] 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, Mar. 2019.
- [4] J. Zhang, C. Li, and X. Ou, Character-level deep learning for phishing URL detection, IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1-14, 2020.
- [5] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, A comprehensive survey of AI-enabled phishing attacks detection techniques, Telecommunication Systems, vol. 76, no. 1, pp. 139-154, Jan. 2021.
- [6] R. Zieni, L. Massari, and M. C. Calzarossa, Phishing or not phishing? A survey on the detection of phishing websites, IEEE Access, vol. 11, pp. 18499-18519, 2023.
- [7] L. Tang and M. Mahmoud, A survey on machine learning models for phishing detection: explainability and fairness perspectives, IEEE Access, vol. 9, pp. 148977-148999, 2021.
- [8] A. K. Jain and B. B. Gupta, PHISH-SAFE: URL features-based phishing detection system using machine learning, in Proc. Int. Conf. Cyber Security and Protection of Digital Services, Glasgow, UK, Jun. 2018, pp. 1-6.
- [9] R. M. Mohammad, F. Thabtah, and L. McCluskey, Predicting phishing websites based on self-structuring neural network, Neural Computing and Applications, vol. 25, no. 2, pp. 443-458, Aug. 2014.
- [10] PhishTank, PhishTank Developer Information, OpenDNS, 2023. [Online]. Available: https://www.phishtank.com/developer_info.php
- [11] OpenPhish, OpenPhish Feeds, 2023. [Online]. Available: <https://openphish.com>
- [12] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 785-794.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
- [14] L. Breiman, Random forests, Machine Learning, vol. 45, no. 1, pp. 5-32, Oct. 2001.
- [15] C. Cortes and V. Vapnik, Support-vector networks, Machine Learning, vol. 20, no. 3, pp. 273-297, Sep. 1995.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152