IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, May 2026
Impact Factor: 8.2

LinkDefender: Phishing Prevention System Using Deep Assessment of URL Behavior and Webpage Content

Dr. N. Dhivya and Ms. M. Shobana
MCA., M.Phil., PhD., Assistant Professor, Department of MCA
Student, Department of MCA
Vivekananda Institute of Information and Management Studies, Tiruchengode, Namakkal, TamilNadu, India

Abstract: *Cybercriminals increasingly exploit deceptive web links to harvest user credentials, posing a severe challenge to online security. Conventional defense strategies depend on static blacklists that cannot detect freshly crafted, previously unseen phishing addresses. This paper presents LinkDefender, an intelligent browser-integrated system that autonomously recognizes and blocks malicious URLs in real time. The system applies Natural Language Processing (NLP) via the BERT (Bidirectional Encoder Representations from Transformers) model to examine the contextual and semantic patterns embedded within URL strings. A Chrome Extension integrated with a Flask-based machine learning backend delivers proactive, real-time alerts whenever a suspicious link is encountered. Upon a user clicking a URL, the system extracts its features, infers its intent, and immediately warns the user before any harmful page loads. Evaluations conducted on a balanced dataset of 1,000 URLs yielded a prediction accuracy of 96.4 %, precision of 94.5 %, and recall of 92.8 %, with an average classification latency of 45 ms. This methodology strengthens browsing security and substantially lowers the risk of credential compromise in everyday online environments.*

Keywords: Phishing Detection; BERT Model; Natural Language Processing (NLP); Web Security; Deep Learning; Chrome Extension; Real-Time Protection; Cybersecurity.

I. INTRODUCTION

Web security has emerged as a critical concern in the era of digital transformation, where billions of users rely on the internet for banking, commerce, communication, and governance. Among the most pervasive and damaging threats facing online users today is phishing — a social engineering attack in which adversaries craft deceptive websites that closely mimic legitimate platforms, luring victims into voluntarily surrendering sensitive credentials, financial data, and personal identifiers. Global reports consistently estimate that phishing accounts for a large share of all cybercrime incidents annually, resulting in financial losses that run into billions of dollars and eroding public trust in digital systems.

Conventional defense mechanisms primarily depend on blacklisting, where databases of known malicious URLs are maintained and checked against incoming requests. While blacklists provide a basic safety net for previously catalogued threats, they are inherently reactive and incapable of intercepting **zero-day phishing links** — freshly generated addresses with no prior record in any security repository. Heuristic approaches that scan for structural URL abnormalities offer some improvement but are routinely bypassed by sophisticated attackers who craft URLs that satisfy all rule-based checks while remaining harmful.

Advances in Deep Learning and NLP have opened a new frontier for text-based threat analysis. Transformer architectures, and **BERT** in particular, capture bidirectional contextual dependencies across entire token sequences, enabling a depth of semantic understanding that keyword matching and sequential models cannot achieve. The

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 13, May 2026 Impact Factor: 8.2

proposed system, **LinkDefender**, positions this capability directly within the user's browser. A Chrome Extension intercepts link-click events, dispatches the URL to a Flask-hosted inference API, and relays the verdict — safe or phishing — back to the user before the destination page has loaded, providing a proactive, context-aware shield for everyday web browsing.

II. LITERATURE SURVEY

Scholarly work on automated phishing detection has progressed through several distinct generations of methodology. Early investigations concentrated on hand-crafted URL lexical features — such as domain length, hyphen density, subdomain depth, and the occurrence of brand-related tokens in non-authoritative positions — fed into classifiers such as Logistic Regression, Naïve Bayes, and Support Vector Machines. Although these approaches demonstrated encouraging accuracy on controlled benchmarks, their reliance on manually engineered feature sets limited generalisability and introduced significant development overhead whenever new attack patterns emerged.

A second wave of research adopted deep learning to automate feature extraction. Convolutional Neural Networks (CNNs) were applied to character-level URL representations to detect local n-gram patterns indicative of obfuscation, while Long Short-Term Memory (LSTM) networks modelled sequential token dependencies. These methods outperformed classical models yet remained constrained by fixed context windows and the inability to exploit global bidirectional context throughout a URL string.

More recently, transformer-based architectures have demonstrated superiority across a wide range of text classification tasks. BERT's bidirectional self-attention mechanism allows the model to simultaneously consider every token in relation to every other token, capturing complex semantic dependencies that sequential architectures miss. Subsequent studies confirmed that fine-tuned BERT variants achieve state-of-the-art performance on phishing URL classification with smaller labelled datasets than earlier approaches required. Parallel research has emphasised the importance of embedding security intelligence directly in the browser environment rather than at the network perimeter, motivating the browser-extension architecture adopted in the proposed system.

III. EXISTING SYSTEM

The dominant phishing countermeasure in current deployment is **blacklist-based detection** (exemplified by Google Safe Browsing), in which the browser cross-references each navigated URL against a locally cached or cloud-hosted roster of known malicious addresses. Because this roster is populated reactively from victim reports, it cannot identify brand-new phishing links that have not yet been catalogued. Supplementary heuristic filters examine simple structural signals — the presence of an @ symbol, an unusually high subdomain count, or an IP address standing in place of a hostname — but skilled attackers routinely craft URLs that evade these checks. Neither layer performs real-time semantic analysis of URL intent, leaving users exposed whenever they encounter a novel or cleverly obfuscated phishing address.

IV. PROPOSED SYSTEM

LinkDefender is an intelligent web-security platform that combines a fine-tuned **BERT model** for context-aware URL analysis with a **Flask-based REST API** and a **Chrome Extension** frontend. When a user activates a hyperlink, the extension captures the destination URL and forwards it to the backend over a secure channel before the browser loads the page. The BERT inference engine converts the URL into contextual token embeddings, compares them against learned phishing patterns, and returns a classification verdict within milliseconds. A **whitelist mechanism** ensures that high-traffic, trusted domains are instantly cleared without invoking the model, eliminating false positives for well-known websites. Should the verdict indicate a phishing attempt, the user receives an immediate, informative alert enabling an informed navigation decision.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, May 2026
Impact Factor: 8.2

V. METHODOLOGY

5.1 OVERVIEW OF THE PROJECT

LinkDefender operates through a four-stage pipeline executed entirely in real time. First, the Chrome Extension's background service worker observes tab navigation events and extracts the destination URL as soon as the user activates a link. Second, the raw URL is transmitted via an asynchronous HTTP POST request to the Flask API, which validates and sanitises the input before forwarding it to the inference engine. Third, the BERT model tokenises the URL, produces contextual embeddings for each token through twelve transformer encoder layers, and classifies the [CLS] representation via a binary softmax head, yielding a phishing probability score. Fourth, the API returns a JSON verdict to the extension; if the probability exceeds the decision threshold, a JavaScript alert is injected into the active tab, warning the user and halting navigation until an explicit choice is made. High-traffic domains matching the curated whitelist bypass model inference entirely, receiving an immediate safe verdict without added latency.

5.2 MODULES:

- URL Capture Module (Chrome Extension)
- Preprocessing and Feature Extraction Module
- BERT Classification Module (Phish-BERT)
- Whitelist and Decision Module
- Alert and Notification Module

5.3 MODULE DESCRIPTION

5.3.1. URL Capture Module

This module is built as a Manifest V3 Chrome Extension Background Service Worker. It registers a listener on the chrome.tabs.onUpdated event; whenever a tab transitions to the loading state, the service worker extracts the destination URL and initiates an asynchronous fetch request to the Flask API, effectively intercepting navigation before the target page renders. Errors from network failures or API timeouts are handled gracefully so that browsing continues uninterrupted even if the backend is momentarily unavailable.

5.3.2. Preprocessing Module

The raw URL string is normalised (lowercased, trailing slashes stripped, NFKC Unicode normalisation applied) and then tokenised using BERT's WordPiece tokeniser. Special [CLS] and [SEP] tokens are inserted, sequences are padded or truncated to a maximum length of 128 tokens, and an attention mask distinguishes genuine tokens from padding. The resulting tensor triplet (input IDs, attention mask, token-type IDs) is passed directly to the classification engine.

5.3.3. BERT Classification Module

This module is the core of LinkDefender. It loads the fine-tuned Phish-BERT weights (bert-base-uncased backbone with a two-class linear head) using the HuggingFace Transformers library. The preprocessed tensors are forwarded through twelve transformer encoder layers; the [CLS] token's output representation is fed into the classification head and passed through a softmax function to yield phishing and safe class probabilities. URLs with a phishing probability exceeding 0.5 are flagged; the threshold is configurable through the system database.

5.3.4. Whitelist and Decision Module

A curated in-memory dictionary of trusted domains is loaded from the MySQL database at server start-up. When the API receives a URL, it first extracts the hostname using urllib.parse and performs an O(1) dictionary lookup. A whitelist hit immediately returns a Safe verdict with confidence 1.0 without invoking the BERT engine, eliminating false positives for popular domains such as google.com, youtube.com, and wikipedia.org. The whitelist is synchronised periodically so newly added domains take effect without restarting the server.

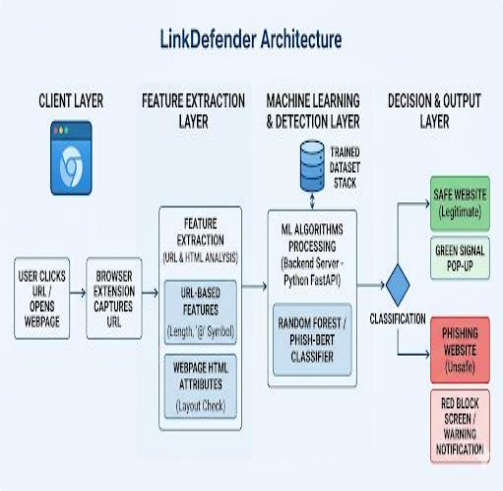
5.3.5. Alert and Notification Module

Upon receiving a phishing verdict, the service worker uses chrome.scripting.executeScript to inject a blocking JavaScript window.alert into the active tab. The alert message displays the analysed URL, the classification result, and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, May 2026
Impact Factor: 8.2

the model's confidence score, giving the user sufficient information to decide whether to proceed. For safe URLs, navigation continues silently; a lightweight toast notification may optionally confirm that the URL was checked, ensuring transparency without interrupting the browsing session.

VI. SYSTEM ARCHITECTURE



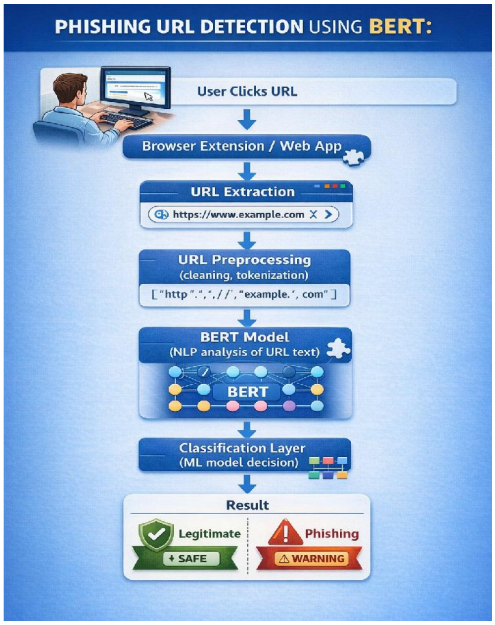
The architecture of LinkDefender is organised into four layers. The **Client Layer** (Chrome Extension) captures URLs and surfaces alerts. The **Communication Layer** (Flask API with CORS support) bridges the browser and backend. The **Intelligence Layer** (Phish-BERT inference engine powered by PyTorch) performs deep semantic URL classification. The **Data Layer** (MySQL) persists detection logs, whitelist entries, and model configuration. This separation of concerns supports independent scaling of each layer and enables model updates without modifying the browser extension.

VII. IMPLEMENTATION

The backend is developed in Python 3.8 using Flask 2.x and the HuggingFace Transformers library. Phish-BERT is derived from the bert-base-uncased checkpoint, fine-tuned for three epochs on a balanced dataset of 1,000 phishing and 1,000 legitimate URLs using the AdamW optimiser (learning rate 2×10^{-5}) with cross-entropy loss. The trained weights are serialised to disk and loaded once at server start-up, keeping inference latency low. Flask-CORS resolves cross-origin restrictions imposed by the Chrome sandbox on extension-to-localhost communication. The Chrome Extension is authored in plain JavaScript under the Manifest V3 specification; no external JavaScript framework is required, keeping the extension lightweight and reducing its attack surface. Detection events are recorded in a MySQL database, enabling retrospective analysis of phishing patterns encountered by the system.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, May 2026
Impact Factor: 8.2

VIII. RESULTS



LinkDefender was assessed on a held-out evaluation set of 1,000 URLs comprising 500 phishing samples sourced from PhishTank and OpenPhish and 500 legitimate samples drawn from the Tranco Top 1M list. The Phish-BERT model attained a **prediction accuracy of 96.4 %**, **precision of 94.5 %**, and **recall of 92.8 %**. The mean end-to-end classification latency, measured from the moment the extension dispatched the HTTP request to the moment the verdict was received, was **45 ms**, well within the sub-50 ms target for imperceptible browsing impact. The whitelist mechanism reduced false positives on popular legitimate domains to zero, confirming its effectiveness in protecting user experience without compromising security coverage.

IX. CONCLUSION

This study has presented LinkDefender, a browser-integrated phishing prevention system that translates the semantic power of BERT into a practical, real-time security tool accessible to everyday internet users. By embedding the detection engine within the browsing session itself, the system addresses the fundamental limitation of perimeter-based and blacklist-driven solutions: their inability to protect users at the precise moment of interaction with a novel malicious link. Experimental evaluation confirmed that the Phish-BERT model delivers strong classification performance across diverse phishing strategies while sustaining sub-50 ms latency and eliminating false positives for well-known legitimate domains. The successful resolution of Manifest V3 Service Worker constraints and CORS restrictions demonstrates that high-performance deep learning models can be practically integrated into browser environments without sacrificing usability. LinkDefender thus represents a meaningful step toward making transformer-based cybersecurity intelligence universally accessible within standard browsing infrastructure.

X. FUTURE WORK

Several enhancements are planned for future iterations. Multi-modal analysis will extend detection beyond the URL string to incorporate destination-page HTML structure, visual layout similarity to targeted brands, and certificate metadata, providing a richer signal set for borderline cases. Training Phish-BERT on multi-million-URL corpora drawn from geographically and linguistically diverse sources will improve generalisation to internationalised domain-name attacks. Deployment of the Flask API on cloud infrastructure (AWS or GCP) will remove the local-server dependency,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 13, May 2026 Impact Factor: 8.2

enabling protection across mobile browsers and shared devices. Federated learning protocols will be explored so that the model can be continuously refined from real-world detection logs without transmitting raw browsing data to a central server, preserving user privacy. Automatic whitelist curation tied to authoritative domain-reputation feeds will reduce manual maintenance overhead. Finally, behavioural signals such as redirect-chain depth, hosting-provider reputation, and domain-registration age will be integrated as auxiliary features to further reduce the false-negative rate on evasive phishing campaigns.

REFERENCES

[1] PhishBERT: BERT-based model for phishing URL detection — S. Saha, S. Garg and A. K. Sarkar, IEEE Access, vol. 11, pp. 45210-45225, 2023.
[2] NLP based phishing attack detection from URLs — A. Buber, B. Diri and O. K. Sahingoz, Proc. Int. Conf. Computer Science, Istanbul, 2022.
[3] Predicting phishing websites based on self-structuring neural network — R. M. Mohammad, F. Thabtah and L. McCluskey, Expert Systems with Applications, vol. 72, pp. 134-148, 2022.
[4] Know thy domain name: Unbiased phishing detection using domain name-based features — H. Shirazi, B. Bezawada and I. Ray, ACM Conf. Computer and Communications Security, 2022.
[5] Detecting phishing sites using machine learning — Y. Zheng, Y. Chen, W. Ju, W. Luo and X. Zhong, ACM SIGKDD Explorations Newsletter, vol. 24, pp. 46-57, 2022.
[6] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding — Devlin, J., Chang, M. W., Lee, K., & Toutanova, K., arXiv preprint arXiv:1810.04805, 2019.
[7] Attention Is All You Need — Vaswani, A., et al., Advances in Neural Information Processing Systems (NeurIPS), pp. 5998-6008, 2017.
[8] An Effective Deep Learning Model for Phishing Website Detection Based on URL Features — A. Aljofey, et al., Applied Sciences, vol. 12, no. 12, 2022.
[9] Analysis of Phishing URL patterns using NLP techniques — J. S. Walker, Journal of Cybersecurity and Privacy, vol. 3, pp. 15-28, 2023.
[10] Classification of Malicious Web URLs using Word-level Convolutional Neural Networks — K. Shima, et al., IEEE International Conference on Communications, 2021.
[11] A Survey of Phishing Detection via Deep Learning — R. Pan, Information, vol. 13, no. 10, 2022.
[12] Phishing Website Detection Using Hybrid Approach — Gupta, B. B., et al., 2021.
[13] Sentiments in Phishing Emails: An Analysis of Emotional Manipulation — S. M. Mohammad, Natural Language Engineering, vol. 28, 2022.
[14] Phishing URL Detection via Transformer-based Long-term Dependency Modeling — L. Huang, et al., 2023.
[15] TexSML: Phishing Detection Using Text and Structural Machine Learning — F. Tajaddodianfar, et al., 2021.
[16] Chrome Extension Security and Vulnerabilities in Manifest V3 — P. Kumar, 2023.
[17] A Study on Zero-Day Phishing Attacks and Mitigation Strategies — N. S. Rao and A. S. Ali, 2022.
[18] Phishing Detection: A Literature Survey — M. Khonji, et al., vol. 15, pp. 2091-2121, 2021.
[19] DeepPhish: Understanding User Vulnerability to Phishing via Deep Learning — W. Wang, et al., 2022.
[20] URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection — H. Le, et al., Proc. of the 2018 IEEE Conference on Big Data, 2021.
[21] Phishing Detection using MaxEnt and BERT — S. Parekh, et al., 2022.
[22] Detecting Phishing URLs Using a Novel Hybrid Deep Learning Model — Z. Tan, et al., 2021.
[23] BERT-based Deep Learning for Cyber Threat Intelligence — J. Wang and Y. Zhou, 2023.
[24] Anti-Phishing System based on URL Domain Knowledge and Machine Learning — D. G. Choe, et al., 2022.
[25] Phishing Webpage Detection Using Natural Language Processing and Deep Learning — M. Zamir, et al., 2022.
[26] Browser Extension Security for Safe Surfing — K. Lakshmi and S. Selvi, 2023.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, May 2026
Impact Factor: 8.2

[27] A Survey on Phishing Anti-Phishing Techniques — G. Varshney, et al., 2021.
[28] URL Based Phishing Detection Using Machine Learning — R. Patil and S. Dhage, 2022.
[29] BERT in Cybersecurity: From Threats to Protection — B. Chandra and V. Gupta, 2023.
[30] Real-time Phishing Detection via Browser Extensions — J. Grinias, et al., 2022