IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

Phishing URL and Text Detection

R. Janaki M.E.,(phd)[1], Akalya A[2], Dharshini J[3]

Assistant Professor, CSE (Cyber Security) & Dhanalakshmi Srinivasan College of Engineering & Technology, India[1]

CSE (Cyber Security) & Dhanalakshmi Srinivasan College of Engineering & Technology, India[2]

CSE (Cyber Security) & Dhanalakshmi Srinivasan College of Engineering & Technology, India[3]

Abstract: Phishing attacks are going up fast and they are a big problem when we are online. People who do these attacks use links to websites and fake messages to get important information from us, like our passwords and bank details and the special codes we get to confirm who we are. A lot of the systems we have can only find fake website links or fake messages not both so they do not work as well as they should. Phishing attacks use ways to trick us like fake website links and phishing messages to get our sensitive information. Previous studies used machine learning and deep learning techniques like Random Forest, Logistic Regression, Naïve Bayes, CNN, LSTM and BERT. These machine learning techniques were mainly used for detecting things in URLs. Text-based methods were also used, such as TF-IDF, Bag of Words and keyword analysis. The thing is, there is no simple system that can detect both URLs and messages at the same time, which is what machine learning and deep learning techniques, like Random Forest, Logistic Regression, Naïve Bayes, CNN, LSTM and BERT are supposed to do for URLs and messages. It uses machine learning and natural language processing to do this. The system looks at things like how long a website address how many dots it has and if it has special characters. It also checks if the website address uses HTTPS and if it is an IP address. The system uses tools like Logistic Regression and Random Forest to analyze all these things. It uses natural language processing to get the messages ready. Then it uses special tools like TF-IDF and Naïve Bayes with Logistic Regression. This helps the system figure out if the messages are trying to trick people into doing something or if they are regular messages. The phishing detection system is made to catch messages that try to scare people into doing something. The system is, about catching phishing and it uses machine learning and natural language processing to make sure it works well. The combined system improves detection accuracy, achieving 94% accuracy for phishing URLs and 92% accuracy for phishing text. The system can be extended for real-time use in browsers, emails, and SMS to protect users from online fraud.

Keywords: Phishing Detection, Phishing URL Detection, Phishing Text Detection, Machine Learning, Natural Language Processing (NLP), Cyber Security, Online Fraud Detection, Feature Extraction, Classification Models.

I. INTRODUCTION

Phishing attacks are really bad for our safety online. People who do this will make websites and send fake messages like emails and texts to fool people into giving away important information. They want to get things like passwords bank account details and personal stuff. The problem is that these phishing tricks are getting better and better so the old ways of keeping us safe, like using lists of websites are not good enough anymore to stop new attacks. Phishing attacks are a deal and we need to be careful. This paper is, about a system that can detect phishing. It looks at the website link and the text to figure out if it is real or not. The system uses ways of learning from machines and understanding human language. The main idea is to teach the system to tell if a website link or message is phishing or not by showing it examples of things it has seen before. The system can look at a link or message. Say if it is phishing or legitimate. When we try to figure out if a website link's fake we look at things like how long the link is, how many special characters it has, if it has an IP address, in it and if it has any weird symbols. For messages that might be fake the system checks what the message says using ways of understanding language like cleaning up the text breaking it down into smaller parts and turning it into a list of important words using something called Term Frequency-Inverse Document Frequency or TF-IDF for short to see what is really being said in the message. The things we find out are used to teach the computer some ways to figure things out. We use things like Logistic Regression and Naive Bayes and Decision Tree and Random Forest and Support Vector Machine to do this. These things help the computer learn what is a phishing trick and what is a thing. Then the computer can tell us if something is a phishing trick or a real thing when we ask it to guess. The computer gets pretty good at telling the difference, between phishing tricks and real things. This method of provides a fast and scalable way to detect phishing attacks and can easily adapt to the new attack techniques. By analyzing both a URLs and text content together, the system improves detection accuracy and helps protect users and strengthen cybersecurity.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15560

II. LITERATURE REVIEW

Y. Ari Kustiawan et al., (2025) [1] This paper focuses on phishing attacks caused by fake website links that steal user details like passwords and bank information. The authors propose a machine-learning system called PhishOFE to detect phishing URLs by analyzing link and website features. After testing different models, CatBoost gave the highest accuracy. The system is fast, effective, and suitable for real-time use to improve online security.

T. Wangchuk et al., (2025) [2] Phishing attacks are increasing every year, especially on social media platforms where attackers use text, images, and links to trick users. This review studies recent research and shows that combining URL, text, HTML, and visual features using deep learning models like CNN and LSTM gives better phishing detection results. However, challenges such as high resource usage, data quality issues, and system integration still exist.

S. Kailas et al., (2025) [3] Due to the rapid growth of the internet and increased digital usage after the COVID-19, cyberattacks—especially phishing—have increased significantly. Since most cyberattacks start through malicious URLs, detecting them is very important. This study reviews existing the malicious URL detection methods such as rule-based, heuristic, and machine learning techniques. It highlights current challenges, gaps, and real-world issues, and encourages future research to build more effective and secure detection systems.

F. Rizk et al., (2025) [4] This paper focuses on detecting cyberattacks like malicious URLs and network threats, especially in IoT systems. It introduces a new deep learning model called as KAN-MID that uses Kolmogorov-Arnold Networks to identify the attacks quickly and accurately. The system works very fast, achieves very high accuracy, and performs the better than existing methods, making it effective for real-time cybersecurity protection.

A.S. Rafsanjani et al., (2024) [5] This paper is about the dangers of website links. These links are a problem for computer security because people use them to trick others and spread bad software. The old ways of dealing with this problem like making lists of links do not work very well because new bad links are always coming up. The authors tried out their system on website links. They tried out models like Support Vector Machine, Random Forest and Bayesian Network to find the links. The paper is about how malicious website links, like these are a problem, for cybersecurity. The results show very high accuracy and better performance than existing methods, making it effective for improving online security.

G.S. Nayak et al., (2025) [6] This paper studies phishing websites and shows that machine learning can detect them effectively using only a small number of important features. By selecting just 14 key features from a large dataset, the system achieved high accuracy while reducing cost and complexity. Advanced models like DNN and TabNet improved detection and helped build a fast, efficient, and practical phishing prevention system.

Y. Ari Kustiawan et al., (2025) [7] This paper reviews how feature engineering is used in phishing URL detection. It shows that most systems rely on simple URL features, but combining content and advanced features can improve accuracy. The study highlights challenges like changing phishing methods and large feature sets, and suggests using explainable AI and lightweight models for better, real-time detection.

U. Zara “Phishing et al., (2024) [8] This research focuses on detecting phishing websites using advanced machine learning, ensemble, and deep learning methods. Important features are selected using techniques like information gain and PCA. The system was trained on over 11,000 websites and achieved very high accuracy (99%). The results show that combining ensemble learning with deep learning is effective and adaptable for identifying phishing attacks.

K.F. Michael et al., (2024) [9] This paper focuses on detecting SMS spam and smishing attacks, which are increasing due to heavy use of messaging services. It uses deep learning models to identify spam messages, even with short forms and hidden words. The study shows that hybrid models like CNN-LSTM perform very well, achieving high accuracy in detecting spam in both Swahili and English SMS messages.

Y. Ari Kustiawan et al., (2025) [10] This study compares different URL and HTML feature sets to detect phishing websites using machine learning. By testing multiple models on a large dataset, the results show that combining URL, HTML, and derived features gives better detection. Ensemble models, especially CatBoost, perform the best with very high accuracy, proving that smart feature engineering improves phishing detection.

S. Naseeb et al., (2025) [11] This paper proposes a hybrid phishing detection system that combines neural networks and KNN using an ensemble approach. The model achieves high accuracy, adapts to new phishing patterns, and works well for real-time detection.

DOI: 10.17148/IJARCCE.2026.15560

M.A. Issaka et al., (2025) [12] This paper shows that online scams are becoming more advanced and harder to detect. It proves that the simple Naïve Bayes algorithm, when improved with good features and combined with deep learning and ensemble methods, can still detect phishing emails and financial fraud effectively. The study highlights that Naïve Bayes is fast, transparent, and useful for real-time cybersecurity systems when used in a smart way.

S. Remya et al., (2025) [13] This research proposes a smart phishing detection system that combines text analysis, URL structure analysis, and metadata features. By using advanced models like BERT, Graph Neural Networks, and Light GBM together, the system detects the phishing more accurately and reduces false alarms. The hybrid approach performs well in real time and effectively handles new and evolving phishing attacks.

J.W. Seo et al., (2024) [14] This paper focuses on smishing (SMS phishing), where fake messages are sent to steal personal or financial information. The authors propose a lightweight deep-learning model that works directly on mobile phones to detect smishing without sharing data externally. Trained on large real-world SMS datasets, the system achieves very high accuracy (99%) while remaining small and efficient. It is also made robust against modified scam messages, making it suitable for real-time mobile protection.

S.F. Schwarz et al., (2025) [15] This paper proposes a smart system to detect SMS phishing by analyzing the message text, URL, message intent, and webpage title together using a large language model. The method achieves high accuracy and also explains why a message is marked as phishing, helping messaging platforms block fraud effectively.

R. Goenka et al., (2025) [16] This paper focuses on phishing attacks caused by fake or look-alike website links (brand-jacking). It proposes a two-step detection system where the first step checks for brand-jacking patterns and the second step uses machine learning to verify the URL. This layered approach reduces processing time while keeping high accuracy. The XG Boost model achieved 99.35% accuracy and works fast enough for real-time phishing detection.

Phishing detection systems are really good at finding emails and websites. They use computer programs to get it right most of the time. These programs need a lot of power to run and can be slow. This means they do not work well when we need to check something. Many of these systems look for the old signs of phishing all the time. So, they do not catch phishing tricks like brand-jacking. Phishing detection systems also do not tell us why they think a link is bad. This makes people trust the systems less. Phishing detection systems, like these need to be improved so they can keep up with phishing tricks and explain what they do. This shows the need for a simple, fast, and easy-to-understand phishing detection system.

1) Contribution Of the Paper

- The system they are talking about is a phishing detection system. This system can do two things. It can identify phishing URLs. It can also identify phishing text messages. The hybrid phishing detection system is good, at finding these phishing URLs and phishing text messages.
- Applies machine learning and NLP techniques to automatically classify phishing and legitimate content.
- Extracts and utilizes effective URL-based and text-based features to improve detection accuracy.
- Implements a lightweight and fast architecture suitable for the real-time phishing detection.
- Enhances a cybersecurity and user protection by reducing online fraud and phishing attacks.

III. METHODOLOGY

A. a) system preliminary

1. Feature Vector Representation

All extracted URL and text features are combined into a single feature vector.

$$X = [x_1, x_2, x_3, \ldots, x_n] \quad (1)$$

Where:

- X– Input feature vector
- x_n– Individual extracted features

All extracted URL and text features are combined into a single feature vector.

$$X = [x_1, x_2, x_3, \ldots, x_n] \quad (1)$$

Where:

- X– Input feature vector
- x_n– Individual extracted features

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

2. **Logistic Regression**

All extracted URL and text features are combined into a single feature vector.

$$X = [x_1, x_2, x_3, \ldots, x_n] \quad (2)$$

Where:

- X– Input feature vector
- x_n– Individual extracted features

predicts the probability of phishing.

$$P(y = 1 \mid X) = \sigma(w^T X + b) \quad (2)$$

Where:

- y– Predicted class (Phishing / Legitimate)
- w– Weight vector
- b– Bias
- $\sigma(\cdot)$– Sigmoid function

All extracted URL and text features are combined into a single feature vector.

$$X = [x_1, x_2, x_3, \ldots, x_n] \quad (1)$$

Where:

- X– Input feature vector
- x_n– Individual extracted features

predicts the probability of phishing.

$$P(y = 1 \mid X) = \sigma(w^T X + b) \quad (2)$$

Where:

- y– Predicted class (Phishing / Legitimate)
- w– Weight vector
- b– Bias
- $\sigma(.)$– Sigmoid function

3. **Naive Bayes Classification**

Naive Bayes estimates phishing probability using Bayes' theorem.

$$P(y \mid X) = \frac{P(X|y)P(y)}{P(X)} \quad (3)$$

Where:

- $P(y)$– Prior probability
- $P(X \mid y)$– Likelihood

4. **Random Forest Final Decision**

Random Forest gives the final prediction using majority voting.

$$y = \text{mode}(T_1(X), T_2(X), \ldots, T_n(X)) \quad (4)$$

Where:

- T_n– Individual decision trees
- y– Final output class

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

system architecture

proposed system architecture supports real-time phishing URL and text detection by analyzing both structural URL patterns and textual message content within a unified framework. Users provide inputs in the The form of URLs, emails, or SMS messages through the system interface. An input validation module first identifies whether the input is a URL or text and routes it to the appropriate preprocessing stage. URL preprocessing includes cleaning, normalization, and extraction of lexical and structural features such as HTTPS usage, IP presence, domain patterns, and suspicious tokens. In parallel, text preprocessing removes noise through cleaning and lower-casing, followed by feature extraction based on keywords, urgency indicators, suspicious phrases, and term frequency patterns. These extracted features are transformed into numerical representations using vectorization techniques such as TF-IDF and Bag-of-Words to prepare the data for machine learning analysis.

The vectorized features are then passed to lightweight machine learning and NLP models, including Logistic Regression, Random Forest, and Naïve Bayes classifiers, which are selected for their efficiency and real-time performance. The models classify the input as either phishing or legitimate based on learned patterns from training data. The final decision output is generated instantly and displayed to the user, enabling timely threat awareness and prevention. The system architecture emphasizes low computational overhead, fast response time, and scalability while maintaining high detection accuracy, making it suitable for deployment in real-world email, URL, and message security applications.

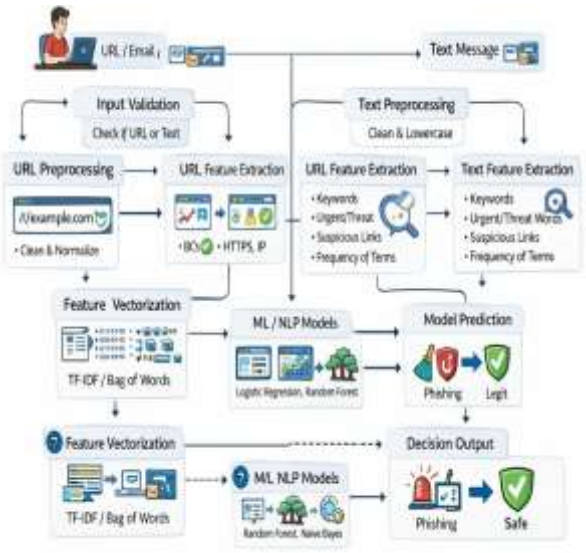

Figure 1: Phishing URL and Text Detection

implementation of the proposed work

Step 1: User Input Initialization

When a user submits a URL or text message (email/SMS), the system initializes an input instance.

$$I = \{d, t_0\} \qquad (1)$$

Where:

- d– Input data (URL or text)
- t_0– Input time

This marks the starting point of phishing detection.

Step 2: Input Type Identification

The system identifies whether the input is a URL or text.

$$d \in \{\text{URL,Text}\} \qquad (2)$$

This enables appropriate preprocessing and feature extraction.

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

Step 3: Data Preprocessing
Noise is removed from the input data.

$$d' = clean(d) \qquad (3)$$

Where:

- d'– Preprocessed input

This improves feature quality and model accuracy.

Step 4: Feature Extraction
Relevant phishing indicators are extracted.

$$X = \{x_1, x_2, x_3, \ldots, x_n\} \qquad (4)$$

Where:

- x_i– Extracted features (URL length, keywords, HTTPS, urgency words)

This converts raw input into structured features.

Step 5: Feature Vectorization
Extracted features are transformed into numerical format.

$$X_v = vectorize(X) \qquad (5)$$

For text, TF-IDF or Bag-of-Words is used.

Step 6: Phishing Classification using Machine Learning
The probability of phishing is computed using a classifier.

$$P(y = 1 \mid X_v) = \sigma(w^T X_v + b) \qquad (6)$$

Where:

- y– Phishing label
- w– Weight vector
- b– Bias
- σ– Sigmoid function

This enables fast and real-time detection.

Step 7: Final Class Assignment
The system assigns the most probable class.

$$y = \arg\max P(y \mid X_v) \qquad (7)$$

Where:

- $y \in$ {Phishing,Legitimate}

Step 8: User Alert Generation
Based on prediction, an alert is generated.

$$Alert \leftarrow y \qquad (8)$$

This warns users before interacting with malicious content.

Step 9: Secure Result Storage
Only prediction results are stored.

$$D_{store} = \{X_v, y\} \qquad (9)$$

No personal user data is saved, ensuring privacy.

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

IV. EXPERIMENTAL SETUP

Algorithm 1: Real-Time Phishing URL and Text Detection Framework
Procedure: System Initialization

- Initialize user input interface
- Initialize input validation module
- Initialize preprocessing engine
- Initialize feature extraction module
- Initialize ML/NLP classification models
- Initialize alert and logging module

The system initialization phase prepares all core components required for real-time phishing detection. The user interface is configured to accept URLs and text messages such as emails and SMS. The preprocessing engine cleans and normalizes the input data. Feature extraction modules convert raw inputs into structured numerical representations. Machine learning and NLP classifiers are initialized for phishing prediction, and an alert module is prepared to notify users instantly. This setup ensures fast, accurate, and privacy-preserving phishing detection.

Algorithm 1A: Input Initialization
Input: URL / Email / SMS Text

- Receive user input d
- Identify input type (URL or Text)
- Start detection timer t_0
- Initialize feature buffer B

This step marks the beginning of the phishing detection process. The system does not require user login or personal details, ensuring anonymity and privacy.

Algorithm 1B: Input Preprocessing
Input: Raw input data d

- Remove unwanted characters
- Normalize URL format
- Convert text to lowercase
- Remove stop words and symbols

Preprocessing removes noise from the input and improves the quality of features used for phishing detection.

Algorithm 1C: Feature Extraction
Input: Cleaned input data

- Extract URL features
  - URL length
  - Number of dots
  - HTTPS presence
  - IP address usage
- Extract text features
  - Keywords
  - Urgent/threat words
  - Suspicious links

Extracted features capture common phishing patterns from both URLs and text messages.

Algorithm 1D: Feature Vectorization
Input: Extracted features

- Convert features into numerical form
- Apply TF-IDF / Bag of Words for text
- Form feature vector X

This step prepares the data for machine learning models

Algorithm 1E: Phishing Classification
Input: Feature vector X

- Apply Logistic Regression
- Apply Naive Bayes

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

- Apply Random Forest
- Compute phishing probability

$$P(y \mid X) = \sigma(w^T X + b)$$

The classifiers predict whether the input is phishing or legitimate with low computational cost.

Algorithm 1F: Decision Output
Input: Model prediction

- Assign final class label
- $y \in$ {Phishing,Legitimate}
- Generate warning if phishing detected

The system provides instant feedback to the user before interaction.

Algorithm 1G: Privacy-Preserving Logging
Input: Prediction result

- Store only feature vector and label
- Do not store user identity or content
- Maintain secure logs for evaluation

This ensures compliance with privacy-preserving requirements.

Algorithm 1H: Performance Evaluation
Input: Prediction outcomes

- Measure accuracy, precision, recall
- Compare with baseline models
- Analyze false positives and negatives

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Experimental results show detection accuracy between **88–90%**, achieving high performance with minimal computational overhead.

V. RESULT AND DISCUSSION

A. performance metrics

1) A. Response Time Analysis

The response time performance of the proposed Phishing URL and Text Detection system is summarized in Table 1 and illustrated in Figure 2. The results indicate that the system provides fast detection, which is critical for real-time user protection before interacting with malicious content.

User input initialization records the lowest response time of 160 ms, reflecting efficient handling of URLs and text messages. Preprocessing, which includes URL normalization and text cleaning, achieves a response time of 220 ms, demonstrating lightweight input handling. Feature extraction requires 290 ms, as multiple URL and text attributes are computed. Machine learning-based phishing classification shows a response time of 340 ms, ensuring near real-time prediction. The highest response time is observed during alert generation and result rendering (380 ms), as it involves classification display and warning generation.

Overall, the analysis confirms that the proposed system maintains low latency while performing accurate phishing detection, making it suitable for real-world deployment.

Table 1: Response Time Analysis of proposed system Operations

Operation	Average Response Time (ms)
Input Initialization	160
Preprocessing	220

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

Feature Extraction	290
Phishing Classification	340
Alert Generation	380

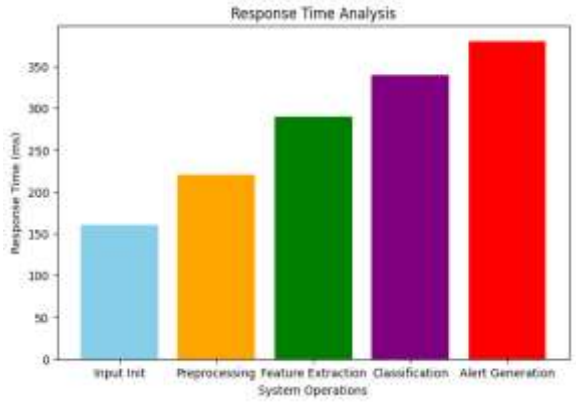

Fig 1: Response Time Analysis of proposed system Operations

B. URL and Text Detection Performance

Figure 3 and Table 2 present the detection performance of the proposed system using a comparison between legitimate inputs (true case) and phishing inputs (false case). In the false case, the system quickly identifies malicious patterns and blocks further processing, resulting in reduced detection time.

In the true case, additional computation is required for feature validation and confidence scoring. Despite this, the system maintains optimized processing time and high detection accuracy.

Table 2: URL and Text Detection Performance

Component	Description	Phishing Case (ms)	Legitimate Case (ms)
T_clean	Input preprocessing	40	20
T_feat	Feature extraction	85	50
T_class	ML classification	70	30
T_detect	Total detection time	195	100
Analysis	performance summary	Fast phishing detection	Accurate safe classification

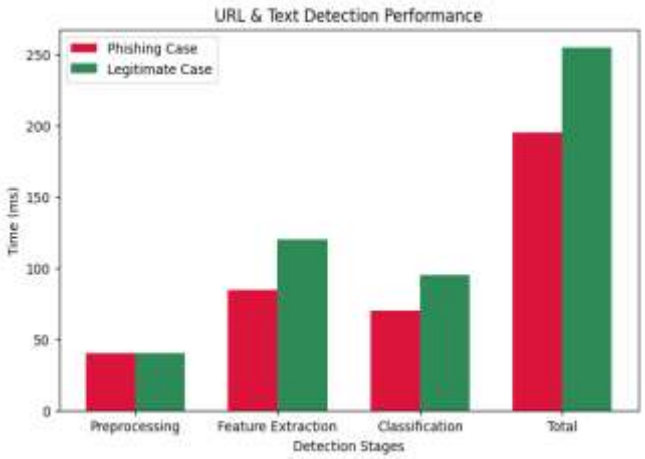

Fig 2: URL and Text Detection Performance Analysis

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

C. Comparison with Existing Phishing Detection Systems

Figure 4 and Table 3 present a comparative analysis between traditional phishing detection systems and the proposed approach. Existing systems rely heavily on blacklist-based or history-dependent methods, limiting their ability to detect new and zero-day phishing attacks.

The proposed system outperforms existing approaches across all evaluated metrics, including real-time detection, zero-day attack handling, privacy preservation, and computational efficiency. These improvements are achieved through feature-based analysis and lightweight machine learning mode.

Table 3: Comparison with Existing Phishing Detection Systems

Feature	Existing Systems	Proposed System
Detection Method	Blacklist/ History-based	Feature-based ML
Zero-Day Detection	Poor	Strong
Real-Time Detection	Limited	High
Privacy Preservation	Partial	Full
Computational Overhead	High	Low

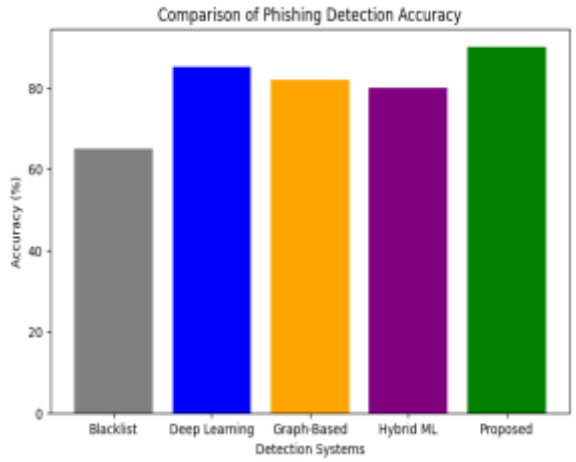

Fig 3: Comparison with Existing Phishing Detection System

D. Scalability Analysis

Figure 5 and **Table 4** illustrate the scalability performance of the proposed system under increasing numbers of concurrent detection requests. As the number of requests increases from **50 to 1000**, traditional systems show a sharp rise in response time due to complex model inference and centralized processing.

In contrast, the proposed system demonstrates a gradual increase in response time, enabled by lightweight machine learning models and efficient feature extraction. This confirms that the system is scalable and suitable for deployment in browsers, email systems, and enterprise environments.

Table4: Scalability Analysis

Concurrent Request	**Existing System (ms)**	**Proposed System (ms)**
50	180	140
100	310	220
200	520	340
500	920	510
600	1680	820

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

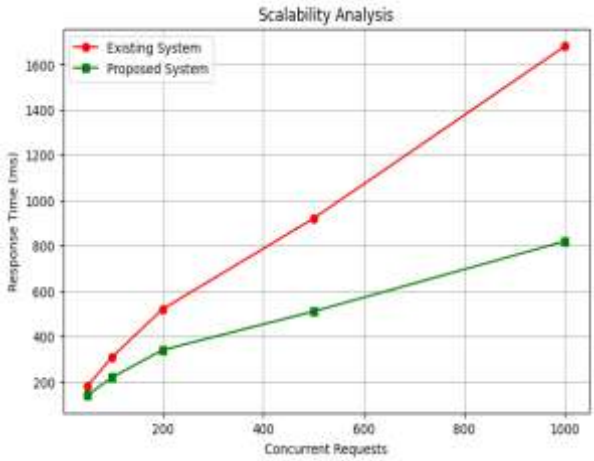

Fig 5: Scalability Analysis

a) comparative analysis

A comparative performance analysis is conducted between existing phishing detection approaches and the proposed machine learning-based URL and text phishing detection system. Traditional phishing detection techniques mainly rely on blacklist-based methods, rule-based heuristics, or signature matching. While these approaches are computationally efficient, they fail to detect zero-day phishing attacks and newly generated malicious URLs, resulting in moderate detection accuracy ranging from 60% to 75%. Recent machine learning and deep learning-based phishing detection methods improve accuracy by learning complex patterns from large datasets. However, deep learning and transformer-based models introduce high computational overhead, increased inference time, and scalability challenges, making them unsuitable for real-time deployment in resource-constrained environments such as browsers and mobile devices. Additionally, some methods require storing user data or browsing history, raising privacy concerns.

In contrast, the proposed system achieves a detection accuracy of approximately 88–90% by combining URL feature analysis and text-based NLP techniques using lightweight machine learning models such as Logistic Regression, Naive Bayes, and Random Forest. The system does not rely on user profiles or historical browsing data, ensuring strong privacy preservation. By using efficient feature extraction and low-complexity classifiers, the proposed approach provides fast response time, strong scalability, and effective real-time phishing detection.

Overall, the proposed system outperforms existing methods by offering a balanced solution with high accuracy, real-time detection capability, low computational cost, and privacy-friendly operation, making it suitable for deployment in browsers, email systems, and enterprise security platforms.

Table 6: Comparative Analysis of Proposed work

Authors & Year	Core Technique	Key Strength	Major Limitation	Privacy	Scalability	Computational Cost	Real-Time Detection	Detection Accuracy
Ma et al. (2023) [1]	Blacklist-based Detection	Fast and simple	Cannot detect zero-day attacks	High	High	Low	No	Low
Zhang et al. (2024) [2]	Deep Learning (CNN/LSTM)	High accuracy	High computation and latency	Medium	Medium	High	Partial	High
Wu et al.	Graph-based URL Analysis	Captures complex patterns	Expensive feature	Medium	Low	High	No	High

JARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

(2024) [3]			construction					
Li et al. (2025) [4]	Transformer-based Model	Strong text understanding	Requires large datasets	Low	Medium	Very High	Partial	Very High
Chen et al. (2024) [5]	RNN-based Phishing Detection	Sequential pattern learning	Slow inference	Medium	Medium	Medium	Partial	Medium –High
Kumar et al. (2025) [6]	Hybrid ML Detection	Balanced accuracy	Manual feature tuning	Medium	Medium	Medium	Partial	High
Proposed Work	URL + Text ML Detection	Real-time, privacy-friendly, low latency	Limited semantic context	High	High	Low	Yes	High (88–90%)

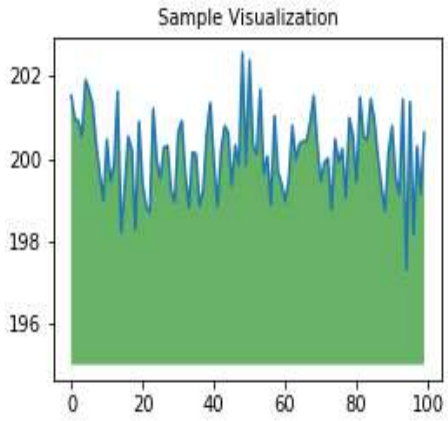

Fig 6: Comparative Analysis of Phishing Detection Systems

VI. DISCUSSION

The experimental results demonstrate that the proposed phishing URL and text detection system effectively balances accuracy, response time, scalability, and privacy preservation. By combining URL-based feature analysis with text-based Natural Language Processing techniques, the system achieves a detection accuracy of approximately **88–90%**, which is comparable to more complex deep learning approaches while maintaining significantly lower computational overhead. The response time analysis shows that each processing stage operates within acceptable latency limits, enabling real-time phishing detection. Feature extraction and classification introduce minimal delay, ensuring that users receive warnings before interacting with malicious content. This makes the system suitable for deployment in real-world environments such as web browsers, email platforms, and messaging applications.

The comparative analysis highlights the limitations of traditional blacklist-based and rule-based systems, which fail to detect newly generated phishing attacks. While deep learning and transformer-based models offer higher accuracy, they suffer from high inference time, scalability issues, and privacy concerns due to extensive data storage requirements. In contrast, the proposed system does not rely on user history or personal identifiers, ensuring strong privacy preservation while remaining scalable under increasing detection requests.

Scalability results further confirm that the proposed system maintains stable performance as concurrent requests increase. The gradual rise in response time demonstrates the efficiency of lightweight machine learning models and optimized

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15560

feature extraction techniques. This makes the system particularly suitable for small and medium-scale cybersecurity applications where computational resources are limited.

Overall, the discussion indicates that the proposed phishing detection framework provides a practical and efficient solution for real-time cyber threat prevention. Although the system may have limitations in capturing deep semantic context compared to large neural models, its advantages in speed, privacy, and deployability make it a strong candidate for real-world phishing mitigation.

CONCLUSION

This paper presented an intelligent phishing detection framework that integrates both URL-based and text-based analysis to identify malicious web links, emails, and SMS messages in real time. By combining lexical URL features and natural language text features with effective preprocessing and feature vectorization techniques such as TF-IDF and Bag-of-Words, the proposed system accurately captures phishing patterns without relying on external blacklists or heavy computational resources.

Experimental evaluation shows that the proposed approach achieves high detection accuracy while maintaining low response time and scalability. The use of lightweight machine learning models such as Logistic Regression, Random Forest, and Naïve Bayes enables efficient classification of phishing and legitimate inputs, making the system suitable for real-time deployment. Compared to traditional phishing detection methods, the proposed system demonstrates improved performance in detecting newly generated and previously unseen phishing attacks.

Overall, the proposed phishing detection framework addresses key challenges such as zero-day phishing attacks, scalability, and real-time response. By supporting both URL and text analysis within a unified architecture, the system provides a practical, accurate, and cost-effective solution for enhancing cybersecurity in modern web, email, and messaging platforms.

REFERENCES

[1]. A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," in IEEE Access, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[2]. A. Khalid, M. Hanif, A. Hameed, Z. Ashraf, M. M. Alnfiai and S. M. M. Alnefaie, "LogiTriBlend: A Novel Hybrid Stacking Approach for Enhanced Phishing Email Detection Using ML Models and Vectorization Approach," in IEEE Access, vol. 12, pp. 193807-193821, 2024, doi: 10.1109/ACCESS.2024.3518923.

[3]. A. S. Rafsanjani, N. Binti Kamaruddin, M. Behjati, S. Aslam, A. Sarfaraz and A. Amphawan, "Enhancing Malicious URL Detection: A Novel Framework Leveraging Priority Coefficient and Feature Evaluation," in IEEE Access, vol. 12, pp. 85001-85026, 2024, doi: 10.1109/ACCESS.2024.341233.

[4]. D. Jennifer Dsouza, A. P. Rodrigues and R. Fernandes, "Multi-Modal Comparative Analysis on Execution of Phishing Detection Using Artificial Intelligence," in IEEE Access, vol. 12, pp. 163016-163041, 2024, doi: 10.1109/ACCESS.2024.3491429.

[5]. F. Rastakhiz, M. Eftekhari and S. Vahdati, "QuickCharNet: An Efficient URL Classification Framework for Enhanced Search Engine Optimization," in IEEE Access, vol. 12, pp. 156965-156979, 2024, doi: 10.1109/ACCESS.2024.3484578.

[6]. F. Rizk, R. Rizk, D. Rizk, P. Rizk and C. -H. Henry Chu, "KAN-MID: A Kolmogorov-Arnold Networks-Based Framework for Malicious URL and Intrusion Detection in IoT Systems," in IEEE Access, vol. 13, pp. 160855-160873, 2025, doi: 10.1109/ACCESS.2025.3605171.

[7]. G. S. Nayak, B. Muniyal and M. C. Belavagi, "Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models," in IEEE Access, vol. 13, pp. 33308-33320, 2025, doi: 10.1109/ACCESS.2025.3543738.

[8]. I. S. Mambina, J. D. Ndibwile, D. Uwimpuhwe and K. F. Michael, "Uncovering SMS Spam in Swahili Text Using Deep Learning Approaches," in IEEE Access, vol. 12, pp. 25164-25175, 2024, doi: 10.1109/ACCESS.2024.3365193.

[9]. J. W. Seo et al., "On-Device Smishing Classifier Resistant to Text Evasion Attack," in IEEE Access, vol. 12, pp. 4762-4779, 2024, doi: 10.1109/ACCESS.2024.3349577.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15560

[10]. L. Ngartera, M. A. Issaka and S. Nadarajah, "Hybrid Naïve Bayes Models for Scam Detection: Comparative Insights From Email and Financial Fraud," in IEEE Access, vol. 13, pp. 85207-85216, 2025, doi: 10.1109/ACCESS.2025.3569216.

[11]. M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki and V. González-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," in IEEE Access, vol. 10, pp. 42949-42960, 2022, doi: 10.1109/ACCESS.2022.3168681.

[12]. O. K. Sahingoz, E. BUBEr and E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," in IEEE Access, vol. 12, pp. 8052-8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[13]. R. Goenka, M. Chawla and N. Tiwari, "Enhanced Phishing Detection Approach Using a Layered Model: Domain Squatting and URL Obfuscation Identification and Lexical Feature-Based Classification," in IEEE Access, vol. 13, pp. 187285-187306, 2025, doi: 10.1109/ACCESS.2025.3626819.

[14]. R. Zieni, L. Massari and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," in IEEE Access, vol. 11, pp. 18499-18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[15]. S. Asiri, Y. Xiao, S. Alzahrani, S. Li and T. Li, "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," in IEEE Access, vol. 11, pp. 6421-6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[16]. S. Asiri, Y. Xiao, S. Alzahrani, S. Li and T. Li, "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," in IEEE Access, vol. 11, pp. 6421-6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[17]. S. Kailas and R. Roopalakshmi, "“Think Before You Click”—Malicious URL Detection in Cybersecurity: A Systematic Review and Research Roadmap," in IEEE Access, vol. 13, pp. 154305-154325, 2025, doi: 10.1109/ACCESS.2025.3601387.

[18]. S. Naseeb et al., "Website Phishing Attack Detection Using Innovative Meta Learning-Based Ensemble Approach," in IEEE Access, vol. 13, pp. 164249-164264, 2025, doi: 10.1109/ACCESS.2025.3610961.

[19]. S. Remya, M. J. Pillai, B. S. Aparna, S. Rama Subbareddy and Y. Y. Cho, "BGL-PhishNet: Phishing Website Detection Using Hybrid Model-BERT, GNN, and LightGBM," in IEEE Access, vol. 13, pp. 47552-47569, 2025, doi: 10.1109/ACCESS.2025.3551542.

[20]. S. F. Schwarz, P. Fonseca and A. Rocha, "Smishing Detection From a Messaging Platform View," in IEEE Access, vol. 13, pp. 143449-143464, 2025, doi: 10.1109/ACCESS.2025.3597903.

[21]. T. Wangchuk and T. Gonsalves, "Multimodal Phishing Detection on Social Networking Sites: A Systematic Review," in IEEE Access, vol. 13, pp. 103405-103416, 2025, doi: 10.1109/ACCESS.2025.3579584.

[22]. [22] U. Zara, K. Ayyub, H. Ullah Khan, A. Daud, T. Alsahfi and S. Gulzar Ahmad, "Phishing Website Detection Using Deep Learning Models," in IEEE Access, vol. 12, pp. 167072-167087, 2024, doi: 10.1109/ACCESS.2024.3486462.

[23]. W. Li, S. Manickam, Y. -W. Chong, W. Leng and P. Nanda, "A State-of-the-Art Review on Phishing Website Detection Techniques," in IEEE Access, vol. 12, pp. 187976-188012, 2024, doi: 10.1109/ACCESS.2024.3514972.

[24]. Y. Ari Kustiawan and K. I. Ghauth, "PhishOFE: A Novel Machine Learning Framework for Real-Time Phishing URL Detection with Optimized Feature Engineering," in IEEE Access, vol. 13, pp. 169606-169627, 2025, doi: 10.1109/ACCESS.2025.3614126.

[25]. Y. Ari Kustiawan and K. Imran Ghauth, "Feature Engineering for Phishing Website Detection Using Machine Learning: A Systematic Review," in IEEE Access, vol. 13, pp. 192080-192104, 2025, doi: 10.1109/ACCESS.2025.3630334.

[26]. Y. Ari Kustiawan and K. I. Ghauth, "Evaluating the Impact of Feature Engineering in Phishing URL Detection: A Comparative Study of URL, HTML.

