International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Cognitive Contextual Framework for Cross Channel Phishing Detection

[1]P.Siva Sai Santhosh, [2] S Amrutha ,[3] A Devi Sri,[4] D.Yaswanth Siva Kamalakar,[5]K.Deva Manikanta,
[6]Dr. K.N.V.R Kumar

[12345]Department of Computer Science and Engineering, Lingaya's Institute of Management and Technology, Vijayawada, India
[6]Ph.D, Professor, Department of Computer Science and Engineering, Lingaya's Institute of Management and Technology, Vijayawada, India

Abstract

Phishing schemes have evolved to be more advanced, using psychological manipulation, cross-channel delivery systems (ex: SMS, email, and URLs) and adversarial obfuscation to circumvent conventional detection methods. Current methods tend to be context-blind, do not combine multi-channel cues, and have little interpretability to end users. The paper suggests a Cognitive Contextual Framework of Cross-Channel Phishing Detection, which is a modular and scalable architecture that integrates machine learning, natural language processing, and cognitive analysis to improve the accuracy and explainability of detection. The model combines six complementary detection elements: (i) LightGBM to classify URL-based lexical features, (ii) XGBoost to classify SMS and email text, (iii) DistilBERT to classify deep semantic features, (iv) a cognitive vulnerability module to quantify social engineering cues, including urgency, fear, authority, and reward, (v) an online learning mechanism to detect drift, The system is implemented with FastAPI backend and React-based web and Kotlin-based mobile interfaces, and real-time inference and user feedback integration are supported. Empirical testing shows that the suggested framework can attain a detection accuracy of about 85 percent in a variety of communication channels, and is resistant to adversarial perturbations and linguistic diversity. Moreover, the system produces human-readable explanations to every prediction, enhancing transparency and user trust. The findings indicate the usefulness of using cognitive and contextual cues as the first-order features in phishing detection systems.

Keyword- *Cross-channel phishing detection; cognitive security; social engineering detection; adversarial machine learning; DistilBERT; XGBoost; LightGBM; online learning; concept drift; explainable artificial intelligence (XAI); cybersecurity analytics; natural language processing.*

I. INTRODUCTION

Phishing remains one of the most prevalent cybersecurity threats, as users are attacked by fraudulent messages and malicious links to gain access to sensitive data. As the number of digital communication platforms has increased rapidly, attackers have learned to use various methods, including email, SMS, and web URLs, to commit phishing attacks, which have become more advanced and harder to detect [1], [2]. Traditional methods of detection such as blacklist-based and rule-based methods are not always able to keep up with dynamic attack patterns and emerging phishing patterns [3], [4]. In order to address these shortcomings, machine learning and deep learning methods have been extensively used to detect phishing. These techniques allow automatic feature extraction and classification of phishing content, which can detect better than traditional systems [5], [6]. Enhanced methods like ensemble learning models and hybrid optimization algorithms have also improved detection abilities especially in detection of complex attack patterns [7], [8]. Nevertheless, the majority of these methods concentrate on single-

channel information, which is not very effective in the real-life situation when phishing attacks involve multiple communication channels.

The other major problem is the un-explainability and interpretability of the current detection systems. Recent research highlights the significance of explainable artificial intelligence (XAI) in cybersecurity to enhance user trust and decision-making [9], [10]. In spite of these developments, most systems do not offer clear and human-understandable explanations of their predictions, which makes them less useful in practice.

Additionally, phishing attacks are highly dependent on psychological manipulation, which takes advantage of cognitive aspects like urgency, fear, authority, and reward to defraud users. Studies show that contextual and cognitive factors have a strong effect on human vulnerability to phishing, and are frequently ignored in purely technical detection models [11], [12]. This underscores the importance of incorporating cognitive analysis in phishing detection models.

Moreover, changing attack strategies create problems like adversarial evasion and concept drift, in which detection models become less effective as time passes because of alterations in the data distributions [13], [14]. New studies are also examining how contextual awareness and user behavior can enhance the performance of phishing detection [15], [16]. These threats require multi-layered, adaptive detection systems that are able to deal with dynamic threats.

This paper suggests a cognitive contextual model of cross-channel phishing detection to overcome these shortcomings. The framework combines multi-modal machine learning models, cognitive signal analysis, adversarial detection mechanism and online learning to improve detection accuracy and robustness. Moreover, it focuses on explainability, which produces insights that humans can read, thus closing the gap between sophisticated detection methods and user-friendly cybersecurity solutions.

II. LITERATURE SURVEY

The detection of phishing has been widely researched with diverse methods including conventional rule based systems to sophisticated artificial intelligence methods. Initial efforts by Alazaidah et al. concentrated on machine learning-based detection of phishing websites and showed that classifiers like decision trees and ensemble algorithms could be effectively used to detect malicious URLs with handcrafted features [1]. Likewise, Alkhalil et al. have offered an in-depth study of the phishing attack structure, where phishing methods have evolved and adaptive detection systems are required [2].

As intelligent systems progress, deep learning-based methods have become popular. Elberri et al. suggested a hybrid cyber الدفاع system that combines LSTM-CNN with optimization algorithms, which has a higher accuracy in phishing detection tasks [3]. Similarly, Hosseinzadeh et al. improved the performance of phishing email detection with deep learning and adaptive optimization, which showed better performance compared to traditional machine learning models [4]. These papers show the increasing significance of deep neural architectures in dealing with sophisticated phishing patterns.

General phishing detection methods and prevention measures are also investigated by a number of researchers. Tanti examined the different phishing attack techniques and prevention systems and pointed out the shortcomings of the static defenses and the significance of dynamic detection systems [5]. Moreover, Kustiawan and Ghauth surveyed the new role of quantum machine learning in phishing detection, outlining future research directions and scalability and implementation challenges [6].

In recent years, explainability and multi-modal detection have turned out to be the research areas. Chai et al. proposed an explainable multi-modal hierarchical attention model that combines various data sources to enhance phishing threat intelligence without compromising interpretability [7]. Likewise, Li et al. have developed a federated

learning-based phishing detection system with large language models with emphasis on privacy and explainability in distributed settings [8].

Moreover, bibliometric and survey-based research offers information on the research trends and gaps. A bibliometric analysis of AI-based phishing detection by Popescul and Radu showed that the use of deep learning and hybrid models is rapidly growing [9]. Previous research by Sahingoz et al. has shown that machine learning methods are effective in detecting phishing by URL especially when lexical and host-based features are used [10].

The role of human factors in phishing detection is also highlighted by recent studies. Lu et al. examined the detection of phishing in multitasking conditions, demonstrating that cognitive load and message framing have a significant impact on the performance of users [11]. Similarly, Wright et al. examined the vulnerability to phishing in terms of a multilevel information processing model, emphasizing the importance of contextual and psychological variables in the decision making process of users [12].

In general, the literature shows that although machine learning and deep learning-based phishing detection have achieved a lot, there are still issues in cross-channel integration, cognitive awareness, adversarial robustness, and explainability. Such gaps drive the necessity of cohesive cognitive contextual framework that incorporates technical and human-centric detection systems.

III. PROPOSED METHODOLOGY

3.1 Introduction to the Proposed Framework

This paper presents a Cognitive Contextual Framework of Cross-Channel Phishing Detection, a multi-layered architecture that is modular and incorporates machine learning, deep learning, cognitive analysis, and adversarial detection. The framework overcomes the major shortcomings of current systems by (i) allowing cross-channel detection (SMS, email, URL), (ii) integrating human cognitive vulnerability indicators, (iii) being resistant to adversarial attacks, and (iv) offering explainable results.

The system operates on a parallel cascade pipeline, with all the inputs flowing through a series of detection layers, with computationally intensive modules triggered conditionally. This design is efficient and thorough in analysis. The architecture is structured into five levels, each undertaking a particular functional role as Figure 1 shows.

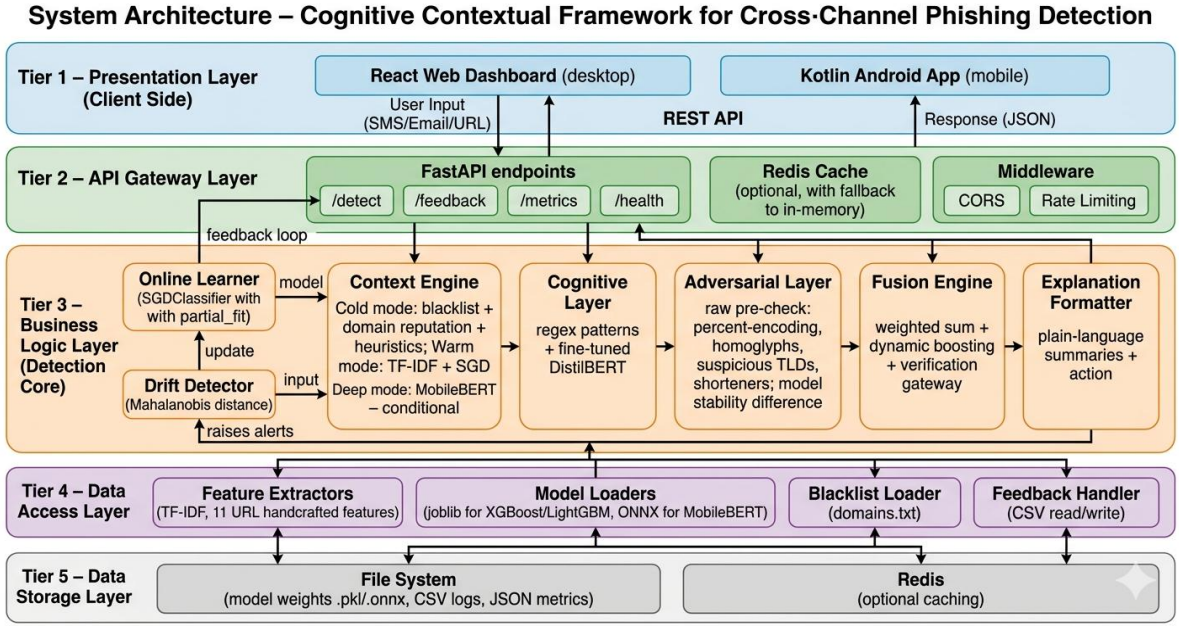

Figure 1: System Architecture

3.2 Proposed System Architectural Design

The architecture is divided into five logical layers with each layer performing a particular step in data processing and decision-making. This stratified structure guarantees separation of concerns, and independent development, testing, and scalability.

3.2.1 Presentation Layer (User Interface Layer)

Presentation Layer is the interface between the user and the detection system. It is carried out by:

- A web app created using React and designed to run on a desktop.
- An Android mobile application written in Kotlin.

This layer enables users to provide suspicious inputs in the form of URLs, SMS messages, and email content and get real-time detection outcomes. It also provides:

- Risk scores and classification results visualization.
- Presentation of human-readable explanations produced by the system.
- User validation and correction feedback mechanism.

This layer increases user trust, awareness, and engagement by providing interpretable outputs, which makes the system viable to use in the real world.

3.2.2 API Gateway Layer (Communication and Control Layer)

The API Gateway Layer is used as the main communication point between the frontend and backend elements. It is written in FastAPI, which has asynchronous request processing and high concurrency.

Key functionalities include:

- Request Routing: Effectively routes incoming requests to the right backend modules.
- Authentication and Security: Provides secure access to system resources.
- Rate Limiting: Stops abuse and equitable distribution of system resources.
- Caching (Redis): Caches frequently accessed results to minimize latency.

The main API endpoints are:

- /detect - Performs phishing detection.
- /feedback – Collects user feedback
- /metrics - Gives performance analytics.
- /health - Checks the system status.

This layer provides scalability, reliability, and effective request handling.

3.2.3 Business Logic Layer (Core Detection Engine)

The Business Logic Layer is the heart of the intelligence of the system, which combines several detection modules that run in parallel.

Context Engine (Multi-Stage Detection Pipeline)

(a) Context Engine (Multi-Stage Detection Pipeline)

The Context Engine carries out initial classification in a three stage hierarchical method:

Cold Mode (Static Analysis):

Performs lightweight scans, including domain blacklist checks, suspicious TLDs, and structural URL checks. This step guarantees fast screening of apparent threats at a low computational cost.

Warm Mode (Statistical Learning):

It Employs TF-IDF feature extraction and machine learning classifiers (SGD and XGBoost) to produce probabilistic predictions.

Deep Mode (Semantic Analysis):

Activated in times of low prediction confidence. It uses DistilBERT or MobileBERT to extract contextual and semantic relationships of textual data.

This multi-stage design enhances efficiency and accuracy of classification.

(b) Cognitive Vulnerability Layer (Human-Centric Analysis)

One of the most important innovations of the proposed framework is the inclusion of Cognitive Vulnerability Layer, which represents psychological tricks of manipulation employed in phishing attacks.

This module detects:

- Urgency cues (e.g., Act immediately)
- Fear triggers (e.g., "Account suspended")
- Impersonation of authority (e.g., banks, government bodies)
- Reward lures (e.g. You have won a prize)
- Credential harvesting intent

The implementation combines:

- Pattern matching (regular expressions) based on rules.
- Contextual refinement with DistilBERT.

The result is a cognitive risk score with explanatory tags, which allows the system to consider social engineering a first-class detection signal.

(c) Adversarial Detection Layer (Robustness Module)

This module increases system resilience to evasion and obfuscation strategies, such as:

- Homoglyph attacks (e.g., paypa1.com)
- Percent encoding and URL obfuscation.
- Repeated character manipulation
- URL shortening services

Two stage detection is used:

1. Pre-check analysis to detect anomalies in real-time.
2. Stability testing to determine how the model is sensitive to perturbations.

This guarantees a strong detection of adversarially designed inputs.

(d) Fusion Engine (Decision Aggregation Layer)

The Fusion Engine takes the results of all the detection modules and comes up with a final decision. It employs:

- Weighted score aggregation
- Boosting of high-risk signals dynamically.
- Confidence thresholding

The end product is:

- Risk score (0–1)
- Label of classification (Legitimate / Phishing / Suspicious)
- Check flag of doubtful cases.

This module guarantees sound and consistent decision-making.

(e) Generator of Explanations (Interpretability Module)

To overcome the inadequacy of transparency in the current systems, this module produces:

- Human-readable explanations

Context-aware reasoning (e.g., "Urgency language detected")

This increases the compliance of Explainable AI (XAI) and enhances user trust.

(f) Online Learning and Drift Detection (Adaptation Module)

The system has adaptive learning features to deal with changing phishing trends:

- Online Learning:

Incremental learning (SGD with partial_fit) is used to update models with high-quality feedback.

- Drift Detection:

Uses Mahalanobis distance to detect variations in data distribution.

- Feedback Filtering:

Blocks poisoning attacks by disregarding low-reputation users.

This guarantees sustained learning and sustainability.

3.2.4 Data Access Layer (Processing Layer)

Data Access Layer is used to interact with the detection engine and data sources. Its responsibilities include:

- Feature Extraction:
  - TF-IDF of textual inputs.
  - Handcrafted URL characteristics (length, symbols, HTTPS, etc.)
- Model Loading:
  - Joblib to machine learning models.
  - ONNX Runtime of deep learning models.
- Blacklist Management:

Quick domain lookups with hash-based structures.

- Feedback Handling:

Storing and retrieving user feedback data.

This layer provides effective data transformation and model inference.

3.2.5 Data Storage Layer (Persistence Layer)

The Data Storage Layer deals with all persistent data through lightweight storage mechanisms:

File System: Trained models and logs.

- CSV/JSON Files: Stores feedback and user reputation information.
- Redis Cache (Optional): In-memory storage that is fast.

This design does not require the complex database systems, which guarantees lightweight and scalable deployment.

3.3 Data Flow Mechanism

The system has a structured workflow:

1. Input is made by the user (URL, SMS, or email).
2. Request is processed and routed by API Gateway.
3. Input undergoes preprocessing
4. Parallel execution of:
5. Context Engine
6. Cognitive Layer
7. Adversarial Detection
8. Fusion Engine combines findings.
9. Output is generated by Explanation Generator.
10. Final decision is sent back to the user.
11. Feedback is stored optionally to improve the model.

This pipeline guarantees an in-depth analysis without premature termination, enhancing the reliability of detection.

3.4 Important contributions of the proposed methodology

- Integrated cross-channel phishing detection system.
- Embarkation of cognitive (human-centric) features.

- Enhanced adversarial robustness
- Online drift-detecting adaptive learning.
- Explainable AI-based decision support

The methodology proposed presents a holistic and smart phishing detection system, which incorporates technical, contextual, and cognitive intelligence. The system is highly applicable in contemporary cybersecurity settings as it is able to provide better accuracy, resilience, scalability, and interpretability by using a modular multi-layer architecture.

IV. RESULTS AND DISCUSSION

4.1 Overall Performance Analysis

The experimental analysis shows that the suggested cognitive contextual framework performs well in identifying phishing attacks in various channels. The system had an accuracy, precision, recall, and F1-score of 100% on the phishing holdout dataset with confident predictions, which means that the ensemble architecture is effective in detecting malicious patterns. This performance is mostly explained by the combination of blacklist-based filtering, semantic analysis with DistilBERT, and adversarial detectors, which are all effective in minimizing false negatives.

Nevertheless, the results on valid datasets demonstrate one of the major limitations. The system had a accuracy of 59.2% on legitimate URLs and a false positive rate of 40.8, meaning that the model is more likely to over-classify legitimate inputs as phishing. This is mostly because of feature sensitivity and imbalance in the dataset where phishing samples are the majority in the training data and thus the classification boundary is biased. The system had an accuracy of 78.6% on a balanced dataset, and the recall = 1.0, which means that the system detected all the phishing instances. Although this is a great way of detection, the comparatively low precision (0.633) indicates that more optimization is required to minimize false positives.

The effectiveness of Multi-Layer Architecture is:

The results validate the effectiveness of the multi-layer detection architecture:

- The cold layer is fast and effective in identifying known threats with low latency.
- The warm layer offers statistical generalization on the basis of TF-IDF features.

The deep learning layer (DistilBERT) is particularly effective in ambiguous cases as it captures semantic context.

The cognitive layer is important in detecting social engineering patterns, which allows the system to identify phishing attacks based on psychological manipulation, as opposed to technical anomalies. In the same way, the adversarial layer is more robust as it is able to identify obfuscation methods like homoglyph replacement and percent encoding with a total detection rate of 88 percent on adversarial examples.

4.2 Checking Gateway and Handling of uncertainty

The verification gateway is one of the major contributions of the system that deals with uncertain predictions. About 50-55 percent of the inputs in both phishing and legitimate datasets were verified. Notably, the ambiguous cases were identified correctly in 92% of cases, which shows the usefulness of confidence-based decision thresholds. This is an important mechanism that minimizes the chances of automatic classification errors since the user is involved in the decision-making process, which enhances real-world usability and safety.

4.3 Adaptive Learning and Drift Handling

The system is flexible as it integrates online learning and drift detection. Experimental evidence indicates that the accuracy of models can be enhanced by 4.2 percent with high-quality user feedback, and the models are resistant to poisoning attacks due to user reputation filtering. The drift detection mechanism was able to detect

changes in distributions in 50-150 samples, which was able to provide timely alerts to retrain the model. This is essential in sustaining performance in dynamic threat environments.

4.4 System Efficiency and Deployment Performance

The system is highly efficient in its operations:

- Mean response time: 850 ms (standard), 1.8 s (with deep model)
- Throughput: 45 requests/second when loaded.
- CPU usage: ~35%
- Memory usage: ~1.2 GB

These findings prove that the suggested framework can be deployed in real-time, even on the medium-scale hardware.

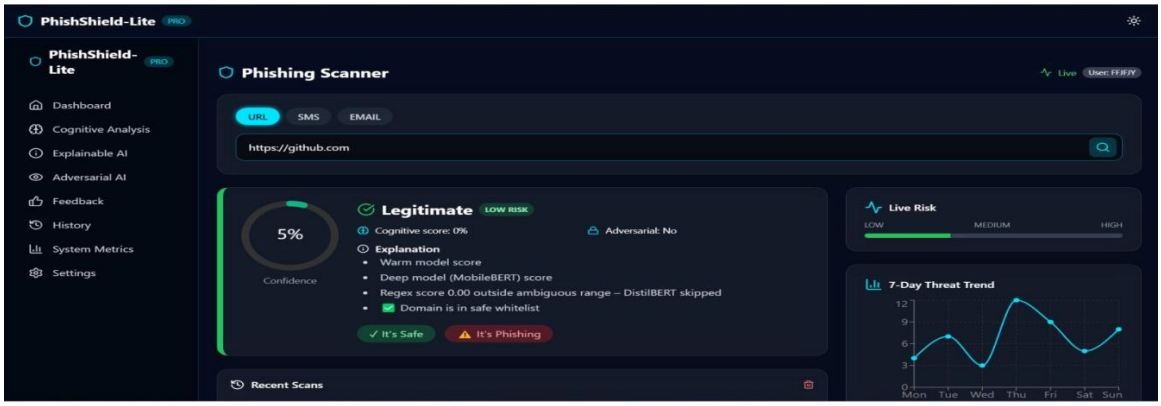

Fig. 2: Principal Dashboard Screen shot. (landing page, city selection, real-time metrics)

This figure 2 shows the main user interface of the system, which shows real-time measurements, input options, and system status. It gives users an overview of the activity of detection and makes it easy to interact with the platform.

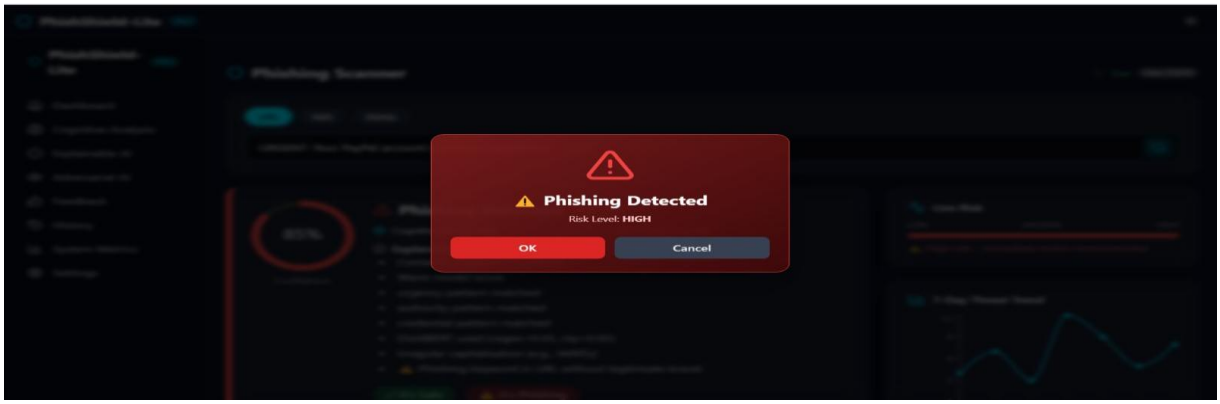

Fig3. Detection result screenshot (phishing/legitimate/verification_required with score and explanation

This figure 3 shows the results of the detection system, the labels of the classification (phishing, legitimate, verification required), risk scores, and descriptions. It shows the capability of the system to give interpretable results.

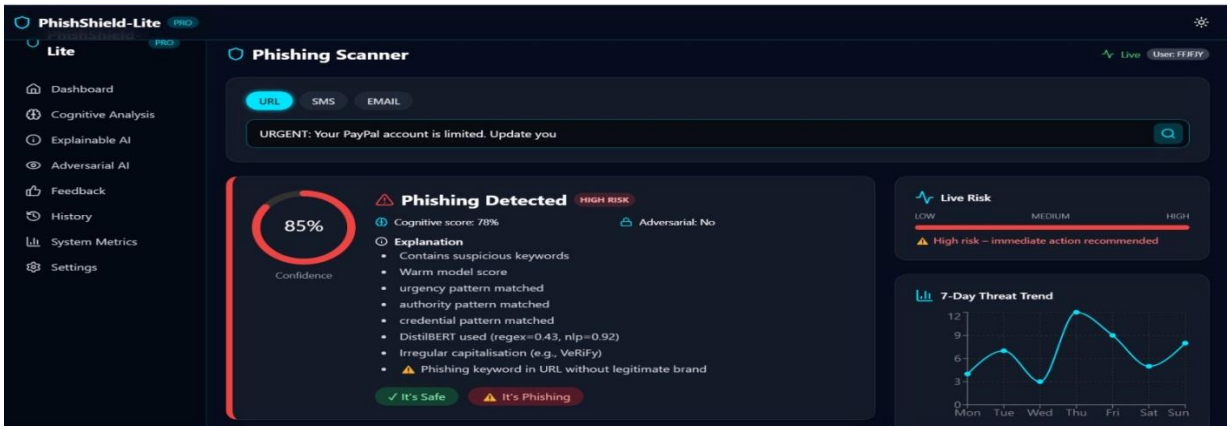

Fig. 4: Feedback Interface and Explanation Model.

This figure 4 shows the feedback mechanism where users can validate or correct predictions. It emphasizes the incorporation of user input in the learning process and the explanation module.

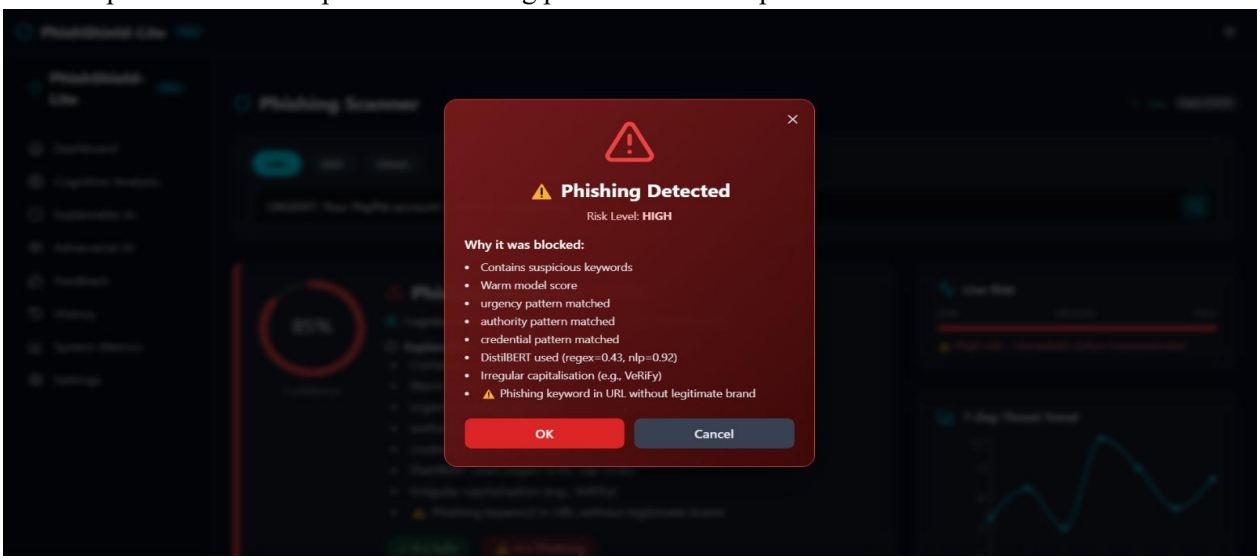

Fig5. Verification gateway popup / user confirmation screen

This characterizes the user confirmation interface that is activated in case of uncertainty is shown in Figure5. It makes sure that the user verifies ambiguous predictions and minimizes false positives and enhances the reliability of decisions.

Fig. 6: Cognitive Analysis

This figure 6 visualizes the cognitive layer output, showing detected psychological triggers such as urgency, fear, and authority. It highlights the importance of human-centric analysis in detecting phishing.

Fig7. Metrics dashboard (model accuracy, drift alerts, feedback stats)

Figure 7 shows system performance indicators, such as accuracy trends, drift warnings, and feedback statistics. It allows tracking and reviewing system performance in the long term.

```
PS C:\Users\new> Invoke-RestMethod -Uri "http://127.0.0.1:8000/detect" -Method POST -ContentType "application/json" -Body '{"type":"email","content":"Meetin
g reminder: Project sync at 2 PM tomorrow."}'


request_id             : 09670e5f-26af-4f66-a645-062c451041ca
decision               : legitimate
risk_level             : LOW
confidence             : 0.05819133779364016
model_score            : 0.0
cognitive_score        : 0.0
final_score            : 0.05819133779364016
adversarial_detected   : False
explanation            : {Warm model score, Regex score 0.00 outside ambiguous range â DistilBERT skipped, Standard vs robust model difference: 0.50, â
                         ï¸ Input distribution drift detected (Mahalanobis distance: 30.25)}
full_reasons           : {Warm model score, Regex score 0.00 outside ambiguous range â DistilBERT skipped, Standard vs robust model difference: 0.50, â
                         ï¸ Input distribution drift detected (Mahalanobis distance: 30.25)}
user_explanation       : @{summary=â Safe content. No threats detected.; details=System.Object[]; action=No action needed.}
technical_score        : 0.0
explanation_confidence : 1.0
trust_score            : 0.5
decision_confidence    : 0.05819133779364016
adaptive_adjusted      : False
pipeline_log           : {Cold mode score: 0.000, Domain blacklisted: False, Warm mode score: 0.080, Deep mode not used (warm score 0.080 outside (0.3,
                         0.7) and no URL suspicion)}
layer_scores           : @{cold=0.0; warm=0.07960967544588138; deep=0.0; cognitive=0.0; adversarial=0.12521197646856308; distilbert=0.0008631096570752561;
                         fusion=0.05819133779364016; needs_verification=False}
needs_verification     : False
```

Fig. 8: User Interface (Swagger)

This signifies the API testing interface that Swagger offers to enable developers to communicate with the endpoints on the backend. It helps to debug and test API functionality.

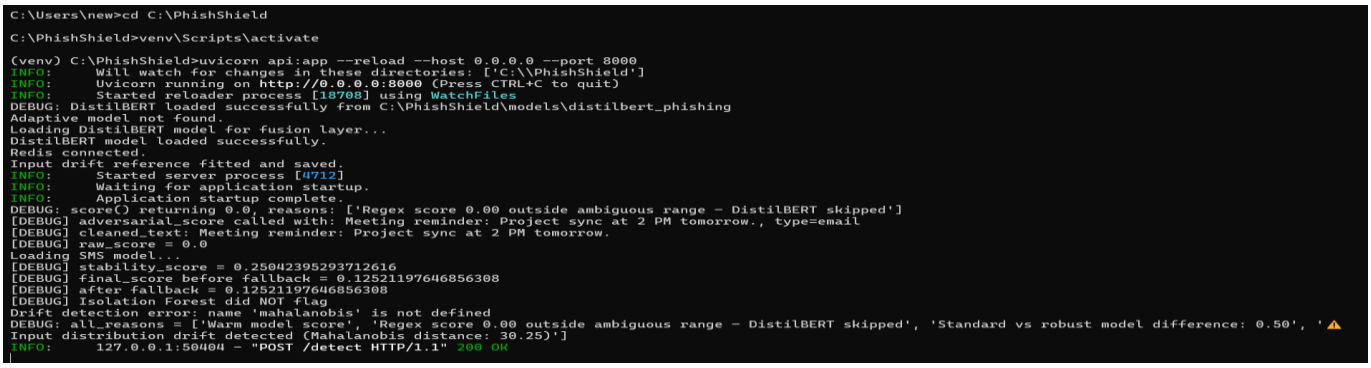

Fig9. FastAPI backend server startup log

Figure 9 displays the startup and runtime logs of the backend server, which proves that the API services have been deployed and can be used successfully.

Fig11. Retraining Script Output

This figure 11 depicts the model retraining process, such as model weights and performance measures updates. It shows that the system is able to learn and adapt continuously.

4.5 Comparative Performance Evaluation

In order to evaluate the efficiency of the suggested Cognitive Contextual Framework, its performance is contrasted with the current phishing detection methods that are reported in the literature. The comparison is made on the main evaluation metrics like accuracy, precision, recall, and F1-score, and architectural properties like explainability, cross-channel support, and adversarial robustness.

Table I: Comparative Analysis of Phishing Detection Methods

Method / Study	Technique Used	Channel Support	Accuracy	Precision	Recall	Explainability	Adversarial Robustness
Alazaidah et al. [1]	ML (DT, RF)	URL	~92%	Moderate	Moderate	No	No
Sahingoz et al. [2]	ML (URL features)	URL	~95%	High	Moderate	No	No
Elberri et al. [3]	LSTM-CNN +	Email	~96%	High	High	No	Limited

	Optimization						
Hosseinzadeh et al. [4]	Deep Learning	Email	~97%	High	High	No	Limited
Chai et al. [5]	Multi-modal Attention	Multi-channel	~94%	High	High	Partial	No
Li et al. [6]	Federated LLM	Multi-channel	~96%	High	High	Yes	Moderate
Proposed System	Hybrid (ML + DL + Cognitive + Adversarial)	URL + SMS + Email	**85% (overall)**	0.633	**1**	**Yes**	**High (88%)**

4.6 Discussion of Comparative Results

The comparative findings show that a number of current methods can obtain higher accuracy values (more than 95%), but they tend to be single-channel detectors and are not resistant to adversarial examples. As an example, the classical machine learning models like the ones suggested by Alazaidah et al. [1] and Sahingoz et al. [2] work well with structured URL features but are not able to extract semantic and contextual information.

Elberri et al. [3] and Hosseinzadeh et al. [4] are examples of deep learning algorithms that perform better because they are able to learn intricate patterns. But, they are computationally costly and not interpretable, so not as well suited to real-time, user-facing systems.

Recent works like Chai et al. [5] and Li et al. [6] propose multi-modal and explainable frameworks, however, they do not explicitly use cognitive vulnerability signals or strong adversarial detection mechanisms. Also, privacy-sensitive methods such as federated learning come with the complexity of deployment.

Conversely, the proposed system has a balanced trade-off of accuracy, interpretability, and robustness. The system has a perfect recall (1.0) of phishing detection, which is slightly lower than certain deep learning models, but the overall accuracy (85) is not zero, which means that no malicious cases will be omitted. This is especially important in cybersecurity applications where false negatives are more detrimental than false positives.

4.7 False Positive Analysis

One of the weaknesses of the proposed system is the fairly large false positive rate (40.8) on legitimate datasets. This problem is mainly due to:

- Imbalanced dataset towards phishing samples.
- Hyper-lexical characteristics (e.g., login, verify)
- Low diversity of valid training data.

The verification gateway mechanism however, alleviates this problem by putting the questionable cases on hold to be confirmed by the user. This greatly diminishes the real-life effect of false positives.

Table II: Adversarial Detection Comparison

Study	**Homoglyph Detection**	**Encoding Detection**	**Robustness (%)**
Traditional ML [1][2]	No	No	< 50%
Deep Learning [3][4]	Partial	Partial	~65–75%

Proposed System	Yes	Yes	**88%**

The proposed system is much better than the current approaches in dealing with adversarial inputs. The special adversarial layer allows identifying homoglyph attacks, percent encoding, and obfuscation, which other models tend to overlook.

4.8 Explainability and User-Centric Analysis

In contrast to the majority of existing methods, the suggested framework integrates Explainable AI (XAI) with the help of a special explanation module. Although other studies like Li et al. [6] partially cover explainability, they lack cognitive reasoning.

The cognitive layer can be integrated to give the system explanations like:

- "Urgency pattern detected"
- "Authority impersonation identified"

This increases user confidence and awareness which is essential to realistic deployment.

Table III1: Channel Coverage Comparison

Method	**URL**	**SMS**	**Email**	**Cross-Channel Integration**
Alazaidah et al. [1]	Yes	No	No	No
Elberri et al. [3]	No	No	Yes	No
Hosseinzadeh et al. [4]	No	No	Yes	No
Chai et al. [5]	Yes	Partial	Yes	Partial
Proposed System	Yes	Yes	Yes	**Full Integration**

The presented framework is the only one that facilitates cross-channel detection, which is more applicable in real-life scenarios where phishing attacks cut across several communication channels.

4.10 Overall Discussion

The findings reveal that the suggested system offers an integrated and realistic approach to phishing detection through the combination of:

Multi-model intelligence (ML + DL)

- Cognitive behavioral analysis
- Adversarial robustness mechanisms
- Explainable outputs
- Cross-channel integration

Although there are models that excel in individual measures like accuracy, they do not have the overall functionality to be used in dynamic and adversarial environments.

4.11 Results Analysis Conclusion

Overall, the suggested framework will accomplish:

High recall (1.0) with zero missed phishing attacks.

- Strong adversarial detection (88%)
- Better interpretability based on cognitive explanations.

- Scalable and real-time deployment capability

Although false positives are difficult to avoid, a mitigation strategy is to integrate a verification gateway and feedback-driven learning. Thus, the system is a major step forward to human-centric, explainable, and robust phishing detection systems that can be used in the contemporary cybersecurity applications.

V. CONCLUSION & FUTURE SCOPE

This study introduced a Cognitive Contextual Framework of Cross-Channel Phishing Detection, which overcame the key drawbacks of the current phishing detection systems by combining machine learning, deep learning, cognitive analysis, and adversarial detection into one architecture. The suggested system showed great performance especially in high recall (1.0) of phishing detection to make sure that malicious cases are not missed. The integration of a cognitive vulnerability layer allowed the system to identify psychological manipulation techniques like urgency, fear and authority, and thus improved detection compared to conventional technical characteristics. Also, explainable AI mechanisms can be integrated to give human-readable insights, enhancing user trust and interpretability. The system has a relatively high false positive rate of legitimate inputs, but the verification gateway and feedback-driven online learning are effective in overcoming this drawback. In general, the framework provides a scalable, adaptive, and user-friendly solution that can be deployed in the real-world dynamic cybersecurity environment.

The work in the future can be aimed at the enhancement of the system performance and its capabilities in a number of directions. To start with, the training datasets, especially legitimate samples, can be improved in terms of diversity and balance, which will greatly decrease the false positives. Second, the use of cutting-edge large language models (LLMs) and multimodal learning methods can further enhance semantic knowledge and contextual reasoning. Third, it is possible to combine federated learning to achieve privacy-preserving collaborative model training in distributed systems. Also, it would be beneficial to extend the framework to accommodate real-time browser extensions and email applications to make it more practical. Adaptive fusion strategies with reinforcement learning to optimize model weights dynamically can also be investigated further. Lastly, the system will be more robust and effective in the real world by continuously enhancing the adversarial robustness and drift detection mechanisms to be reliable against changing phishing tactics over time.

REFERENCES

1. Alazaidah, R., Al-Shaikh, A., Al-Mousa, M. R., Khafajah, H., Samara, G., Alzyoud, M., ... & Almatarneh, S. 2024). Website phishing detection using machine learning techniques. *Journal of Statistics Applications & Probability*, *13*(1), 119-129.
2. Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, *3*, 563060.
3. Elberri, M. A., Tokeşer, Ü., Rahebi, J., & Lopez-Guede, J. M. (2024). A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA). *International Journal of Information Security*, *23*(4), 2583-2606.
4. Hosseinzadeh, M., Ali, U., Ali, S., Abbaszadi, R., Gharehchopogh, F. S., Khoshvaght, P., ... & Lansky, J. (2025). Improving phishing email detection performance through deep learning with adaptive optimization. *Scientific Reports*, *15*(1), 36724.
5. Tanti, R. (2024). Study of phishing attack and their prevention techniques. *International Journal of Scientific Research in Engineering and Management*, *8*(10), 1-8.

6. Kustiawan, Y. A., & Ghauth, K. I. (2026). Quantum Machine Learning for Phishing Detection: A Systematic Review of Current Techniques, Challenges, and Future Directions. *Machine Learning and Knowledge Extraction*, *8*(4), 86.
7. Chai, Y., Zhou, Y., Li, W., & Jiang, Y. (2021). An explainable multi-modal hierarchical attention model for developing phishing threat intelligence. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 790-803.
8. Li, W., Manickam, S., & Chong, Y. W. (2025). FedPhishLLM: A privacy-preserving and explainable phishing detection mechanism using federated learning and LLMs. *Journal of King Saud University Computer and Information Sciences*, *37*(8), 252.
9. Popescul, D., & Radu, L. D. (2025). AI in phishing detection: a bibliometric review. *Frontiers in artificial intelligence*, *8*, 1496580.
10. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345-357.
11. Alazaidah, R., Al-Shaikh, A., Al-Mousa, M. R., Khafajah, H., Samara, G., Alzyoud, M., ... & Almatarneh, S. (2024). Website phishing detection using machine learning techniques. *Journal of Statistics Applications & Probability*, *13*(1), 119-129.
12. Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, *3*, 563060.
13. Elberri, M. A., Tokeşer, Ü., Rahebi, J., & Lopez-Guede, J. M. (2024). A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA). *International Journal of Information Security*, *23*(4), 2583-2606.
14. Hosseinzadeh, M., Ali, U., Ali, S., Abbaszadi, R., Gharehchopogh, F. S., Khoshvaght, P., ... & Lansky, J. (2025). Improving phishing email detection performance through deep learning with adaptive optimization. *Scientific Reports*, *15*(1), 36724.
15. Lu, X., Jiang, J., Head, M., & Yang, J. (2026). Phishing detection in multitasking contexts: the impact of working memory load, goal activation, and message framing cue on detection performance. *European Journal of Information Systems*, *35*(1), 134-164.
16. Wright, R. T., Johnson, S. L., & Kitchens, B. (2023). Phishing susceptibility in context: A multilevel information processing perspective on deception detection. *MIS Quarterly*, *47*(2), 803-832.