# Social Engineering and Phishing Protection using AI-Driven Methods

Afsana Mustafazade
Istanbul Gedik University, Istanbul, Turkey

Mohamed El-dosuky
Arab East Colleges, Saudi Arabia, and
Faculty of Computers and Information, Mansoura University, Egypt

Sherif Kamel
Arab East Colleges, Saudi Arabia, and
October University for Modern Sciences and Arts, Egypt

## ABSTRACT
Phishing and social engineering attacks remain major cybersecurity threats by exploiting human behavior rather than system vulnerabilities. This paper investigates artificial intelligence (AI)-driven approaches for detecting such attacks through a combination of literature review, dataset evaluation, system design, and experimental analysis. It examines traditional machine learning models, deep learning techniques, and transformer-based natural language processing methods. A hybrid detection framework is proposed, integrating semantic text analysis, deep learning, and URL-based features within a modular architecture. The system is implemented using a fine-tuned RoBERTa model for email classification and evaluated on multiple benchmark datasets. Results show that transformer-based models achieve high accuracy in identifying contextual and manipulative patterns, while URL-based methods enable fast real-time detection. The findings demonstrate that hybrid approaches significantly enhance detection performance and robustness, particularly against AI-generated phishing attacks, providing a scalable and adaptive solution for modern cybersecurity challenges.

## General Terms
Phishing Detection; Social Engineering; Natural Language Processing; Cybersecurity.

## Keywords
Phishing Detection; Social Engineering; Natural Language Processing; Cybersecurity.

## 1. INTRODUCTION
Phishing and social engineering attacks remain persistent and evolving threats within modern cybersecurity, largely because they exploit human behavior rather than technical vulnerabilities [1]. Unlike traditional system-level exploits, these attacks manipulate emotions, trust, and decision-making, making them significantly harder to detect with conventional security tools. Attackers use advanced strategies such as AI-generated content, deceptive communication patterns, and personalized targeting [2]. This raises the need for intelligent, adaptive defense mechanisms has become increasingly urgent.

Recent studies highlight major progress in applying machine learning, natural language processing, and deep analytical models to identify fraudulent messages and behavioral anomalies [3]. Techniques such as deep neural networks, transformer-based architectures, and linguistic analysis provide new ways to capture subtle indicators of deception that older rule-based systems often miss.

This research paper combines multiple complementary tasks: reviewing some contemporary academic studies on phishing detection, evaluating available phishing datasets, designing system architectures through UML diagrams, and implementing BERT-based classification models for email analysis. Together, these tasks form a structured exploration of modern defensive technologies and their practical capabilities. By integrating insights from academic literature, dataset evaluation, model design, and experimental testing, this work offers a clearer understanding of how AI-driven methods can strengthen defenses against phishing and social engineering threats.

The rest of the paper is presented as follows. Section 2 is for reviewing literature, before analysis and design in Section 3. Section 4 is for presenting the common datasets. Section 5 is for model selection. Section 6 is dedicated for proposed system details, while Section 7 is for results and discussion. Finally, the conclusion and future work are presented in the last section.

## 2. LITERATURE REVIEW
Phishing and social engineering continue to pose serious risks in cybersecurity because these attacks manipulate human decision-making instead of exploiting system vulnerabilities [4]. This makes conventional rule-based defenses increasingly ineffective, especially as attackers adopt more sophisticated psychological techniques. For this reason, recent academic studies place strong emphasis on AI-driven solutions that can recognize deceptive patterns in communication, URLs, and user behavior [5].

Classical machine learning models such as SVM, Random Forest, and Logistic Regression remain popular due to their speed and simplicity [6]. However, they rely heavily on predefined features and struggle to adapt when attackers alter their strategies or introduce new linguistic tricks. More advanced deep learning approaches address this limitation. Architectures like CNN and LSTM demonstrate high performance by learning deeper semantic and contextual relationships within text [7].

A further leap in progress comes from transformer-based models, including BERT [8]. These systems excel at understanding context, word dependencies, and emotional tone, enabling them to detect manipulative language that traditional ML systems often miss [9]. Despite their strong results, they demand considerable computational power and sometimes misinterpret legitimate messages. On the operational side, URL-focused detection models such as AdaBoost and fully connected neural networks provide rapid classification for real-time filtering [10]. However, because they concentrate on structural and lexical indicators, they lack the ability to interpret psychological or semantic manipulation.

Recent literature also warns about emerging threats driven by

generative AI, such as deepfakes [11] and AI-crafted phishing content [12]. Findings suggest that combining deep semantic NLP models like BERT with fast URL-based methods offers the most reliable multi-layered defense.

# 3. ANALYSIS AND DESIGN

Designing an effective phishing and social engineering detection system requires integrating insights from prior research while addressing practical limitations such as real-time applicability, computational efficiency, and adaptability to evolving threats. This system employs a modular, object-oriented architecture represented through UML diagrams, consisting of data acquisition, preprocessing, feature extraction, and classification components. Textual and URL-based datasets are preprocessed using tokenization, normalization, and embedding techniques such as Keras Embedding or GloVe to preserve semantic meaning.

Classification combines hybrid AI approaches. Recurrent Neural Networks capture sequential dependencies and semantic relationships, while transformer models like BERT provide contextual and emotional analysis to identify manipulative content. URL-based modules, using AdaBoost and fully connected neural networks, offer rapid detection suitable for real-time monitoring. Adaptive layers address generative AI threats, including GPT-generated phishing and deepfakes. The modular design ensures scalability, multilingual support, and future integration, providing a robust and extensible framework for phishing and social engineering protection.

The proposed phishing detection system is implemented as a Python-based application following the object-oriented design. The core classification module employs RoBERTa, which is based on a fine-tuned BERT-base-uncased model (Hugging Face Transformers) that analyzes the semantic and contextual properties of email text [13]. The implementation strictly respects the designed UML artifacts: the system context (Figure 1), use-case interactions (Figure 2), sequence of operations (Figure 3), and the email processing state machine (Figure 4).


**Fig 1: Context Diagram**

Email analysis begins when a user submits a message through the web interface. The text is tokenized with BERT's built-in tokenizer (maximum sequence length 512), fed into the fine-tuned model, and classified as either "Safe" or "Phishing." The prediction, together with confidence scores, is immediately stored in a PostgreSQL database and displayed to the user.

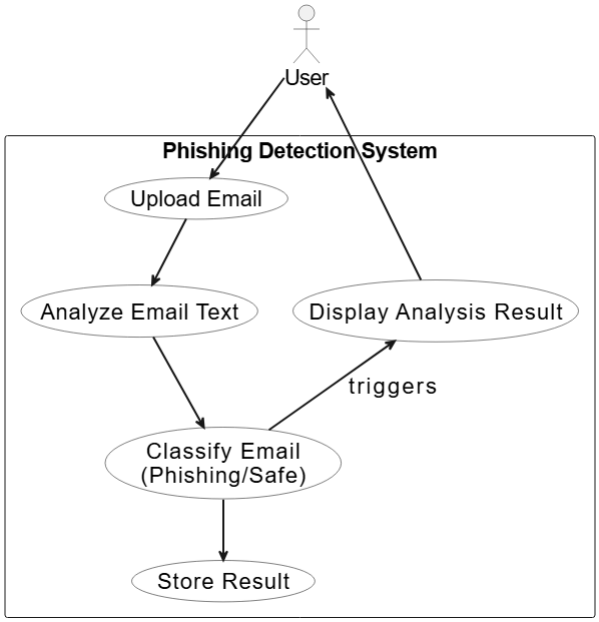


**Fig 2: Use-case interactions**

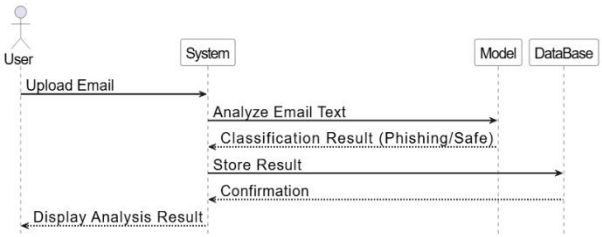


**Fig 3: Sequence Diagram**

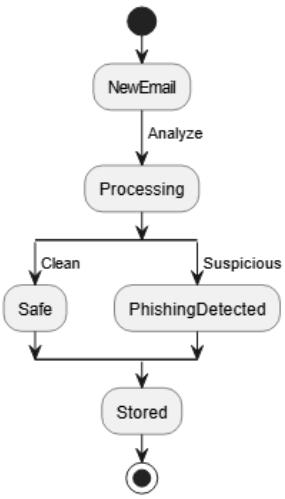


**Fig 4: Flowchart of processing a new email**

# 4. DATASETS

The datasets used in this study vary in size, source, and focus, as shown in Table 1. The Ling dataset, derived from the Linguist List online email communications in the early 2000s [14], consists of 458 spam and 2,401 legitimate emails, making it suitable for linguistics-related studies but limited in scope and size. The Enron dataset, extracted from the Enron email corpus around 2005 [15], contains 13,976 spam and 15,791 ham emails, offering extensive corporate communications, though with limited diversity. The Spam Assassin dataset, developed by the Apache Software Foundation between 2002 and 2006

[16], includes 1,670 spam and 3,928 ham emails, providing a diverse set of public spam messages, albeit with potentially outdated tactics. The CEAS_08 dataset comes from the CEAS 2008 Challenge Lab Evaluation Corpus [17] and contains 21,839 spam and 16,853 ham emails, capturing a broad range of communications but reflecting dated spam characteristics. Nazario is a balanced dataset with 1,469 phishing and 1,482 legitimate emails [18], useful for model testing but lacking comprehensive coverage of phishing tactics. Finally, the Nigerian dataset is publicly available and comprises 1,586 advance-fee scam emails and 2,967 legitimate emails from 1998 to 2007 [19], focusing on a specific type of fraud but with limited diversity.

**Table 1. Datasets used in this study**

| Dataset | Reference | Phishing Emails | Legitimate Emails |
|---|---|---|---|
| **Ling** | Linguist List online emails, early 2000s [14] | 458 | 2,401 |
| **Enron** | Enron email corpus, ~2005 [15] | 13,976 | 15,791 |
| **Spam Assassin** | Apache Software Foundation, 2002–2006 [16] | 1,670 | 3,928 |
| **CEAS_08** | CEAS 2008 Challenge Lab Evaluation Corpus [17] | 21,839 | 16,853 |
| **Nazario** | Balanced sources [18] | 1,469 | 1,482 |
| **Nigerian** | Publicly available, 1998–2007 [19] | 1,586 | 2,967 |

# 5. MODEL SELECTION

Table 2 lists the utilized Artificial Intelligence-driven methods. Several AI-driven methods have been widely employed for phishing detection, ranging from traditional machine learning models to advanced deep learning architectures. Naive Bayes, a probabilistic classifier grounded in Bayes' theorem, estimates posterior class probabilities using feature likelihoods to distinguish between phishing and legitimate emails [20].

Logistic Regression has also been extensively utilized as a statistical approach for binary classification, where input features are fitted to a logistic function, weighted optimally, and converted into probability scores for threshold-based decision making [21].

**Table 2. Utilized AI-driven methods**

| Classifier | Key Concept | How It Works |
|---|---|---|
| **Naive Bayes** [20] | Probabilistic classifier based on Bayes' theorem | Computes posterior probability of each class based on features; classifies emails as phishing or legitimate |
| **Logistic Regression** [21] | Statistical technique for binary classification | Fits data to a logistic function; assigns weights to features; computes probability score; classifies based on threshold |
| **Random Forest** [22] | Supervised ML algorithm for classification and regression | Constructs multiple decision trees from training data; combines predictions via ensemble voting or averaging; runs trees in parallel to reduce overfitting and improve accuracy |
| **CNN** [23] | Automatically extracts important features; works well for text classification | input layer (TF-IDF vectors), followed by dense layers with ReLU activation and dropout to prevent overfitting, and a final sigmoid output layer |
| **LSTM** [24] | Handles long- and short-term dependencies | Embedding layer for tokenized sequences; LSTM layer with many units and Tanh activation captures dependencies; dense layer with ReLU; final sigmoid neuron outputs. |
| **BERT** [25] | Transformer-based, bidirectional context-aware model | Pre-trained BERT (Bidirectional Encoder Representations from Transformers) encoder generates contextual embeddings; dense layer with ReLU processes embeddings; output sigmoid neuron |
| **DistilBERT** [26] | Lightweight, faster version of BERT | Encoder generates embeddings; fully connected layer processes embeddings; softmax output layer produces class probabilities |
| **RoBERTa** [8] | Optimized BERT with better training and generalization | Base BERT architecture with adjusted hyperparameters; pre-trained on multiple corpora; used for fine-tuning on specific tasks like phishing detection |

To enhance predictive performance and mitigate overfitting, Random Forest constructs multiple decision trees from bootstrapped samples and aggregates their outputs through ensemble voting or averaging mechanisms [22]. In the realm of deep learning, Convolutional Neural Networks (CNNs) have demonstrated strong capability in automatically extracting discriminative textual features by processing TF-IDF representations through convolutional and dense layers with ReLU activation and dropout regularization [23]. Long Short-Term Memory (LSTM) networks further improve sequence modeling by capturing both short- and long-term dependencies via embedding layers followed by LSTM units and dense transformations, making them particularly effective for contextual text classification [24].

More recently, transformer-based models have achieved superior performance due to their contextual understanding of language; BERT leverages bidirectional attention mechanisms to generate rich contextual embeddings that are refined through

dense layers for phishing classification tasks [25]. To address computational efficiency, DistilBERT offers a lightweight yet effective alternative by compressing the BERT architecture while maintaining high representational power for text embeddings and classification [26]. Furthermore, RoBERTa enhances the original BERT framework through optimized hyperparameters and large-scale pre-training strategies, resulting in improved generalization and robustness when fine-tuned for phishing detection applications [8].

# 6. PROPOSED SYSTEM

The proposed architecture is logically divided into blocks, as shown in Figure 5. Each of these blocks is described in the following subsections.

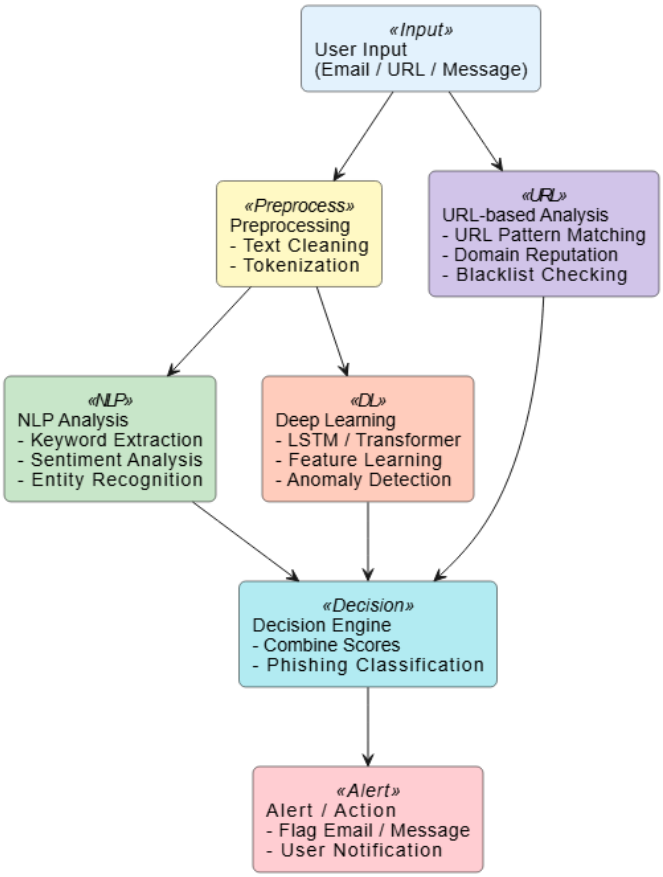


**Fig 5: Proposed System**

## 6.1 User Input

This block represents the raw data that enters the phishing detection system. It can include emails, URLs, instant messages, or any text-based communication that may contain social engineering attacks. The quality and format of the input data significantly affect the performance of downstream modules, as incomplete or malformed data may reduce detection accuracy. Proper handling at this stage ensures that all relevant information is captured for analysis without introducing noise.

## 6.2 Preprocessing

Preprocessing transforms raw input into a structured format suitable for analysis. For textual content, this involves cleaning steps such as removing HTML tags, stopwords, punctuation, and special characters, followed by tokenization to split text into meaningful units. Preprocessing also standardizes input for deep learning models, converting text into vectors or embeddings. For URLs, preprocessing might include parsing the domain, path, and query parameters. This step ensures consistency and improves the quality of features fed into both NLP and deep learning modules.

## 6.3 NLP Analysis

Natural Language Processing (NLP) extracts semantic and syntactic features from text-based input. Keyword extraction identifies suspicious terms often used in phishing emails, sentiment analysis detects manipulative or urgent language, and entity recognition identifies targeted organizations, personal names, or financial institutions. NLP features provide interpretable signals that can indicate potential phishing attempts, enabling the system to flag content that exhibits manipulative or abnormal language patterns even before deep learning models are applied.

## 6.4 Deep learning

The deep learning block leverages advanced neural network architectures to automatically learn complex patterns in text and features. LSTM or Transformer-based models can capture sequential dependencies and contextual relationships in text, detecting subtle cues of phishing attempts that rule-based systems might miss. This block may also perform anomaly detection by identifying messages that deviate from normal communication patterns, providing a probabilistic score indicating phishing risk.

## 6.5 URL-based Analysis

URL-based analysis inspects web addresses within messages to detect potentially malicious destinations. It includes syntactic pattern matching to identify suspicious URL structures,

checking the domain against known blacklists, and evaluating reputation metrics from external sources. Features such as the use of uncommon TLDs, excessive subdomains, or obfuscated links provide strong indicators of phishing attempts, complementing textual analysis performed by NLP and deep learning.

## 6.6 Decision Engine

The decision engine integrates the outputs from NLP analysis, deep learning models, and URL-based analysis to generate a unified classification. Using weighted scoring, ensemble methods, or probabilistic models, this block determines the likelihood that a given input is malicious. By combining multiple sources of evidence, it improves overall accuracy and robustness, reducing false positives and false negatives that may occur when relying on a single detection approach.

## 6.7 Alert/Action

Once a message is classified, the alert/action block takes appropriate measures to protect the user. This may include flagging the email or message as phishing, quarantining it, or sending a warning to the end user. In automated systems, it can trigger further actions such as blocking the URL or reporting the domain to cybersecurity authorities. This block ensures timely intervention, mitigating the impact of phishing and social engineering attacks.

# 7. RESULTS

To compare accuracies of different models on different datasets, Figure 6 provides an annotated heatmap, which is a compact and highly interpretable visualization. This figure reproduces results from [27].

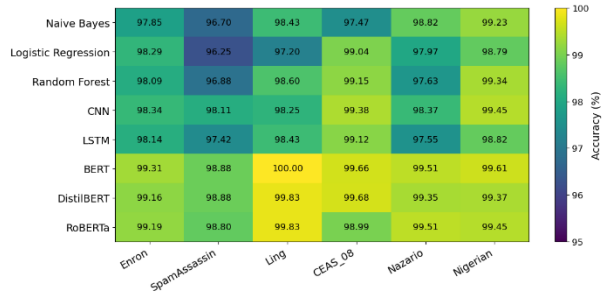


**Fig 6: Annotated heatmap of accuracy**

Rows correspond to the models (Naive Bayes, Logistic Regression, Random Forest, CNN, LSTM, BERT, DistilBERT, and RoBERTa) and columns correspond to the datasets (Enron, Spam Assassin, Ling, CEAS_08, Nazario, and Nigerian). Each cell contains the accuracy value of a specific model on a specific dataset, displayed numerically and simultaneously encoded by color intensity. The color gradient, fixed between 95% and 100% in this study, allows for rapid visual identification of performance differences: darker shades indicate higher accuracy, while lighter shades indicate comparatively lower performance.

The inclusion of numerical annotations within the cells ensures that precise values are available at a glance, eliminating ambiguity that could arise from relying solely on color. This combination of color coding and annotated values makes the heatmap particularly effective for highlighting subtle differences in accuracy between high-performing models, which might be difficult to discern in traditional bar plots. For instance, the heatmap immediately reveals that transformer-based models (BERT, DistilBERT, and RoBERTa) consistently achieve near-perfect accuracy across most datasets, while classical models such as Naive Bayes and Logistic Regression exhibit slightly lower performance on certain datasets.

Table 3 lists the classification report for RoBERTa on a merged dataset, while Figure 7 depicts the corresponding confusion matrix.

**Table 3. Classification Report for RoBERTa**

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Legitimate | 0.99 | 1.00 | 0.99 | 5,939 |
| Phishing | 1.00 | 0.99 | 1.00 | 6,434 |
| **Macro Avg** | 0.99 | 1.00 | 0.99 | 12,373 |
| **Weighted Avg** | 0.99 | 0.99 | 0.99 | 12,373 |

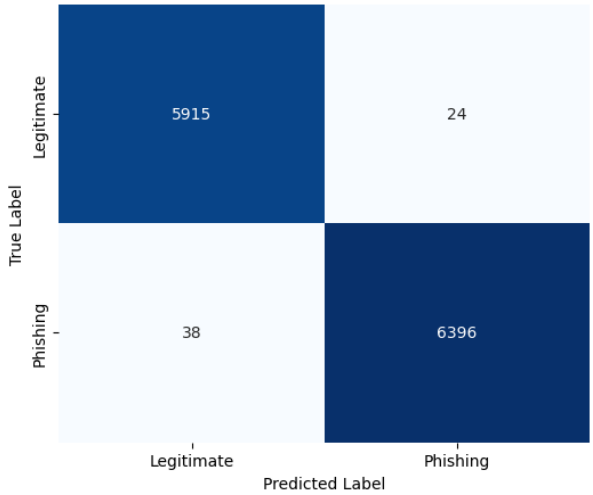


**Fig 7: Confusion Matrix for RoBERTa**

The implemented detection system demonstrated high performance across multiple evaluation metrics and attack scenarios. Transformer-based models, such as RoBERTa, accurately detected contextual manipulations and emotional cues in phishing messages

The integration of multiple detection layers enhanced overall robustness, significantly reducing false negatives compared to single-method approaches. Adaptive mechanisms addressing generative AI threats successfully flagged previously unseen phishing attacks, including AI-generated messages and deepfake content, demonstrating the system's capability to adapt to evolving threats. Cross-validation confirmed model generalizability across diverse datasets, while comparative analyses highlighted the superiority of layered hybrid models over traditional machine learning systems in both semantic comprehension and operational efficiency.

Results indicate that combining semantic, contextual, and technical analyses leads to a comprehensive detection framework. The findings emphasize that hybrid systems incorporating deep learning and transformer-based NLP models, complemented by fast URL-based classifiers, can provide highly effective and reliable protection against phishing and social engineering attacks. This demonstrates the practical feasibility of deploying such systems in real-world cybersecurity environments

The findings from the implemented system indicate that hybrid approaches combining NLP, deep learning, and URL-based

analysis are highly effective for phishing and social engineering detection. Transformer-based models, including BERT, provide advanced contextual understanding. This allows the system to detect subtle manipulative cues that conventional methods often overlook. URL-based modules complement the semantic analysis by providing fast, real-time classification, crucial for monitoring live traffic and minimizing exposure to threats.

Adaptive mechanisms addressing generative AI-enhanced attacks, including GPT-generated phishing messages and deepfakes, demonstrated significant potential in identifying previously unseen threat patterns. However, limitations remain. Transformer models are computationally intensive, and their reliance on English-language datasets restricts global applicability. Occasional false positives arise due to context ambiguity, which may require additional fine-tuning or multilingual datasets. Despite these challenges, the modular, object-oriented design ensures scalability, allowing future integration of multilingual data, multimodal inputs, and adversarial robustness strategies. The discussion underscores that layered hybrid systems, which combine semantic, contextual, and technical analyses, provide the most promising framework for modern phishing detection and social engineering defense.

# 8. CONCLUSION AND FUTURE WORK

This research confirms that hybrid frameworks combining NLP, deep learning, and URL-based analysis provide the most reliable approach for detecting phishing and social engineering threats. Models such as transformer-based architectures (BERT) demonstrated superior performance in semantic and contextual comprehension, while URL-based modules ensured real-time detection efficiency. Adaptive mechanisms addressed the growing threat of generative AI attacks, highlighting the system's capacity to handle both conventional and AI-enhanced phishing attempts.

The modular, object-oriented design allows extensibility, facilitating integration of multilingual datasets, advanced anomaly detection, and real-time operational deployment. Future research should explore multimodal approaches, incorporating voice, video, and image data to address emerging AI-driven phishing strategies. Efforts to optimize transformer models for computational efficiency and reduce false positives will enhance practical applicability. Additionally, testing the system in live operational environments and developing robust adversarial defenses will further strengthen resilience. This framework provides a comprehensive, scalable solution, offering a foundation for ongoing advancement in phishing and social engineering protection.

# 9. REFERENCES

[1] Akeiber, H.J., 2025. The evolution of social engineering attacks: a cybersecurity engineering perspective. Al-Rafidain Journal of Engineering Sciences, pp.294–316.

[2] Zdrojewski, K., 2025. AI-powered cyberattacks: a comprehensive review and analysis of emerging threats. Advances in IT and Electrical Engineering, 31, pp.55–70.

[3] Boulieris, P., Pavlopoulos, J., Xenos, A. and Vassalos, V., 2024. Fraud detection with natural language processing. Machine Learning, 113(8), pp.5087–5108.

[4] Wang, Z., Zhu, H. and Sun, L., 2021. Social engineering in cybersecurity: Effect mechanisms, human vulnerabilities and attack methods. Ieee Access, 9, pp.11895-11910.

[5] Fakhouri, H.N., Alhadidi, B., Omar, K., Makhadmeh, S.N., Hamad, F. and Halalsheh, N.Z., 2024, February. Ai-driven solutions for social engineering attacks: Detection, prevention, and response. In 2024 2nd international conference on cyber resilience (ICCR) (pp. 1-8). IEEE.

[6] Tang, L. and Mahmoud, Q.H., 2021. A survey of machine learning-based solutions for phishing website detection. Machine Learning and Knowledge Extraction, 3(3), pp.672-694.

[7] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K. and Faheem, M.H., 2023. A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics, 12(1), p.232.

[8] Jamal, S., Wimmer, H. and Sarker, I.H., 2024. An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. Security and Privacy, 7(5), p.e402.

[9] Acheampong, F.A., Nunoo-Mensah, H. and Chen, W., 2021. Transformer models for text-based emotion detection: a review of BERT-based approaches. Artificial Intelligence Review, 54(8), pp.5789-5829.

[10] Veach, A.M. and Abualkibash, M., 2022. Phishing website detection using several machine learning algorithms: a review paper. International Journal of Informatics, Information System and Computer Engineering (INJIISCOM), 3(2), pp.219-230.

[11] Mirsky, Y. and Lee, W., 2021. The creation and detection of deepfakes: A survey. ACM computing surveys (CSUR), 54(1), pp.1-41.

[12] Parasaram, V.K.B., 2023. Phishing Detection 2.0: A Natural Language Processing Approach to Identifying Generative AI-Crafted Social Engineering. In International Journal for Research Publication and Seminar (Vol. 14, No. 2, pp. 276-284).

[13] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp.4171–4186.

[14] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D. and Stamatopoulos, P., 2003. A memory-based approach to anti-spam filtering for mailing lists. Information Retrieval, 6, pp.49–73.

[15] Klimt, B. and Yang, Y., 2004. The Enron corpus: a new dataset for email classification research. In European Conference on Machine Learning. Berlin/Heidelberg: Springer, pp.217–226.

[16] Champa, A.I., Rabbi, M.F. and Zibran, M.F., 2024. Curated datasets and feature analysis for phishing email detection with machine learning. In 3rd IEEE International Conference on Computing and Machine Intelligence (ICMI), pp.1–7.

[17] DeBarr, D. and Wechsler, H., 2012. Spam detection using random boost. Pattern Recognition Letters, 33, pp.1237–1244.

[18] Nazario, J., 2023. The Online Phishing Corpus. Available at: https://monkey.org/~jose/phishing/ Accessed 20

January 2026.

[19] Radev, D., 2008. CLAIR Collection of Fraud Email. Available at: https://aclweb.org/aclwiki/CLAIR_collection_of_fraud_email_(Repository) Accessed 20 January 2026.

[20] Bagui, S., Nandi, D., Bagui, S. and White, R.J., 2021. Machine learning and deep learning for phishing email classification using one-hot encoding. Journal of Computer Science, 17, pp.610–623.

[21] Shahrivari, V., Darabi, M.M. and Izadi, M., 2020. Phishing detection using machine learning techniques. arXiv, arXiv:2009.11116.

[22] Salloum, S., Gaber, T., Vadera, S. and Shaalan, K., 2021. Phishing email detection using natural language processing techniques: a literature survey. Procedia Computer Science, 189, pp.19–28.

[23] Moriya, S. and Shibata, C., 2018. Transfer learning method for very deep CNN for text classification and methods for its evaluation. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), IEEE, Tokyo, Japan, 23–27 July 2018, Vol. 2, pp.153–158.

[24] Khataei Maragheh, H., Gharehchopogh, F.S., Majidzadeh, K. and Sangar, A.B., 2022. A new hybrid based on long short-term memory network with spotted hyena optimization algorithm for multi-label text classification. Mathematics, 10, p.488.

[25] Eang, C. and Lee, S., 2024. Improving the accuracy and effectiveness of text classification based on the integration of the BERT model and a recurrent neural network (RNN_BERT_Based). Applied Sciences, 14, p.8388.

[26] Sanh, V., Debut, L., Chaumond, J. and Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv, arXiv:1910.01108.

[27] Alhuzali, A., Alloqmani, A., Aljabri, M. and Alharbi, F., 2025. In-depth analysis of phishing email detection: evaluating the performance of machine learning and deep learning models across multiple datasets. Applied Sciences, 15(6), p.3396.