Shrushti Kaza, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Real-World Case Study: Evaluating AI-Powered and Traditional Signature Approaches to Email Phishing Threats

Shrushti Kaza, Akhila Harshini Gadamsetty, Abhijeet Raj, Pranav Veer Singh, Dr M Umamaheswari

School of Computer Science and Engineering Vellore Institute of Technology Vellore, India.

Abstract- Email-based phishing is among the persistent and costly cybersecurity challenges which exploit human gullibility, social engineering, and organizational frailties for gaining un-approved access to sensitive information. Signature-based cyber-security strategies use preset patterns, blacklists, and heuristic approaches to identify phishing emails. While signature-based detection systems can recognize phishing emails with known char-acteristics successfully, they usually fail to identify sophisticated attacks which evade recognition due to their novelty or disguise. On the other hand, modern technologies based on ML and NLP employ numerous features including email body, natural language used, sender behavior, URLs embedded in an email, and additional metadata. The ability of such approaches to generalize makes them applicable in the detection of previously unseen phishing campaigns. In this study, comprehensive comparison between AI-powered phishing detectors and traditional signature-based methods is conducted using a hand-curated dataset with both legitimate and malicious samples of emails. Criteria for evaluation include detection rates, false positives and negatives, as well as computational resources consumed. The experiments show that AI-based techniques outperform traditional systems in terms of recognizing unknown phishing emails. However, superior performance comes at the expense of greater computational loads and increased requirements for tuning and maintaining AI models. Also, this study provides practical guidance for integrating AI-based phishing detectors into corporate email systems, considering deployment issues, scaling, and computational resources needed. Based on the experiment results presented in the paper, recommendations are made regarding implementation of AI solutions for phishing attacks.

Keywords: Email phishing, cybersecurity, phishing detec-tion, machine learning, artificial intelligence, signature-based detection, intrusion detection, email security, anomaly detection, spam detection.

I. INTRODUCTION

The primary goal of a phishing attack involves the exploita-tion of trust, as well as the employment of social engineering to gain access to classified information through fraudulent emails designed to resemble authentic messages. The attacks remain one of the most prevalent cyber threats to organizations and individuals, causing substantial financial damage and com-promising the security of private data and business processes. Conventional signature-based detection technologies rely on prior definition of malicious patterns, blacklists, and heuristic rules. While this strategy has proven to be highly effective in combating familiar phishing attempts, its efficiency towards novel and advanced attacks appears rather low.

In order to overcome the existing shortcomings, there has been a development of more innovative approaches, which employ machine learning (ML) and artificial intelligence (AI). Unlike traditional static signature-based models, an AI-based solution can learn to generalize certain patterns based on historical information and detect novel types of phishing attempts. In particular, this type of technology employs a variety of variables, including email's structure and metadata, sender's reputation, and linguistic patterns to predict potential threats. Thus, an AI system can effectively combat novel and obfuscating phishing attacks that are difficult to identify using conventional techniques.

This paper presents a case study comparison of conventional signature-based and artificial intelligence-based phishing de-tection methods. In

particular, the research will compare the efficiency of these techniques in terms of detection accuracy and rate, along with their computational requirements in organizational context. This parallel analysis is intended to draw attention to the respective strengths and weaknesses of both solutions, thus providing practical insights about the enhancement of an email security scheme.

II. LITERATURE REVIEW

Phishing email detection has become an area of great research interest in recent times due to its direct practical importance. Traditional methods such as blacklists and rule-based systems prove inefficient against dynamically evolving threats and inspire the search for alternative approaches. Machine learning and deep learning approaches have proven valuable during the last decade as tools to analyze complex patterns of malicious emails.

This literature review aims to build on previous works analyzing traditional ML models, ensembles of different ap-proaches, and solutions based on neural networks, thus pro-viding an understanding of each approach's methodology, advantages, and disadvantages. This will help to define gaps in modern solutions and give rise to new effective anti-phishing strategies.

The latest literature reviews include applications of tradi-tional as well as modern deep learning approaches for phishing detection. Ogbebor et al. [1] show high efficiency of NLP, LSTM, and GNN models at the high price of high computation cost. The superiority of hybrid approaches which integrate 1D-CNN, LSTM, and GRU compared to conventional ML models is demonstrated by Altwaijry et al. [2]. On the other hand, S.

S. et al. [3] and Jaison et al. [6] emphasize the weaknesses of conventional ML and rule-based approaches, namely low interpretability and inability to evolve. The special-purpose solutions for spear-phishing attacks and phishing URLs sug-gested by Birthriya et al. [4] and Abid et al. [5], respectively, yield above 90% efficiency when tested on diverse datasets.

Modern research emphasizes the effectiveness of advanced hybrid and deep learning approaches. Sadiya et al. [7] con-clude that LSTM, BiLSTM, CNN, and Transformer models with TF-IDF features surpass traditional ML models although being quite computationally expensive.

Pandya et al. [8] utilize explainable ML methods (SHAP, LIME, OCR, NLP) for efficient and fast prediction while paying attention to increased complexity. Alsariera et al. [9] prove that using ensembles increases robustness at the cost of higher computational load. Al-Subaey et al. [10] provide a very accurate (99.1%) yet interpretable web-based AI solution. Thapa et al. [11] compare lightweight quantized large language models (Distill Qwen 14B) achieving over 80% accuracy on a minimal VRAM budget.

A hybrid approach of using both semantics and structure analysis is emerging in phishing research. Chen et al. [12] use the combination of DistilBERT, TF-IDF, and Random Forest to extract URL and text features. Kumar et al. [13] prove that quantized LLMs can reach an accuracy level of 80% or more at high efficiency levels. Li et al. [14] use the Random Forest, Gradient Boosting, BiLSTM, and GNNs approaches to achieve 98.3% accuracy with a low latency rate. Khan and Patel [15] and Nguyen et al. [16] show that classical machine learning algorithms and BiLSTM models still prevail while Garc´ıa et al. [17] reveal the Gradient Boosting model has 97.2% accuracy with ten-fold cross-validation; however, the performance highly depends on hyperparameters used.

To summarize, phishing detection has moved from classical machine learning techniques to more advanced hybrid and deep learning models through applying NLP, LSTM/GRU architecture, Transformers, and ensembles. Even though all those technologies provide impressive levels of accuracy and robustness, they have high requirements concerning compu-tation power and data volume. Hybrid structures capable of providing semantic and structural analysis may become an innovative approach to solving the problem efficiently.

III. PROPOSED METHODOLOGY

The proposed methodology is intended to provide an eval-uation of the efficiency of three AI-based classifiers (Random Forest, Support Vector Machine, and Deep Neural Network) versus a classical signature-based detection system. The ar-chitecture is based on four key stages, namely, preprocessing, feature extraction, classification, and evaluation.

A. Architecture Description

First, emails taken from real-life datasets are subject to preprocessing that allows one to eliminate noise, clean texts, and gather information such as header fields, links, and metadata associated with the sender. Next, feature extraction is implemented by determining various lexical, structural, and content-based characteristics. These features are utilized by both the AI-powered algorithms and the signature-based detection system. Finally, the models generate classification results that can be analyzed with conventional measures.

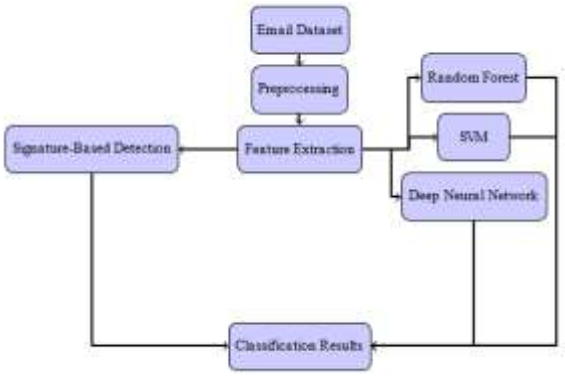

Fig. 1. Proposed architecture for phishing detection using AI models and signature-based methods.

B. AI Models

• **Random Forest (RF):** Random Forest (RF) is a widely used ensemble learning method based on decision trees. It constructs a large number of individual decision trees during training, each trained on a random subset of the data and features. The final prediction is obtained through majority voting (for classification) or averaging (for re-gression), which significantly reduces the variance com-pared to a single decision tree. This ensemble approach renders RF extremely efficient for high-dimensional and heterogeneous feature spaces with resistance to noise and overfitting. Additionally, RF naturally includes feature importance measures that improve interpretability and is capable of performing with missing values and class imbalance with little preprocessing.

Algorithm 1: Random Forest Classifier

1: **Input:** Training set $D = \{(x_i, y_i)\}$, number of trees T
2: **for** each tree $t = 1$ to T **do**
3: Sample D_t from D with replacement (bootstrap sample)
4: Train decision tree h_t on D_t using random feature selection at each split
5: **end for**
6: **Output:** For a test sample x, predict $\hat{y} = \text{majority vote}(h_1(x), h_2(x), \ldots, h_T(x))$

• **Support Vector Machine (SVM):** The Support Vector Machine is a supervised learning method widely used for solving classification and regression tasks. It works by finding the optimal hyperplane that can efficiently separate samples of different classes from each other in the feature space. When dealing with data that cannot be separated using a linear hyperplane, SVM uses kernelfunctions such as linear, polynomial, and radial basis function (RBF). This enables the mapping of data into a higher dimensional feature space where linear separationcan be achieved.

The ability to model complex deci-sion boundaries, together with its effectiveness in high-dimensional spaces and resistance to overfitting, makes SVM suitable for tasks including text classification, im-age classification, and phishing detection. SVM models are highly efficient when the number of features exceeds the number of samples, providing excellent generalization and few hyperparameters.

Algorithm 2: Support Vector Machine

1: **Input:** Training set $D = \{(x_i, y_i)\}$, regularization parameter C, kernel function K
2: Solve the optimization problem:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

$$\text{s.t. } y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

3: **Output:** Decision function $f(x) = \text{sign}\left(\sum_i \alpha_i y_i K(x_i, x) + b\right)$

•Deep Neural Network (DNN): DNNs are neural net-works that can learn hierarchical features from the initial data inputs through their multiple layers. An average DNN consists of input, multiple hidden layers with non-linear activation functions, and output layers, which are used to classify data or regression problems. Through multiple layers, DNNs enable learning complex features and patterns, which cannot be learned by simple ma-chine learning algorithms. Backpropagation algorithms with stochastic gradient descent or Adam optimizer are commonly used in DNN training process, along with regularization techniques such as dropout or batch nor-malization to overcome overfitting. DNNs have been used extensively to reach state-of-the-art results in computer vision, natural language processing, and cybersecurity applications such as phishing and malware detection, especially when large labeled datasets are available.

Algorithm 3: Deep Neural Network

Input: Training set $D = \{(x_i, y_i)\}$, network architecture (layers, units, activation)
Initialize weights W and biases b
for epoch = 1 to N **do**
  for each mini-batch $B \subset D$ **do**
    Forward pass: compute activations $a^{(l)}$ for all layers
    Compute loss L (e.g., cross-entropy)
    Backpropagate loss: compute gradients $\frac{\partial L}{\partial W}, \frac{\partial L}{\partial b}$
    Update parameters: $W \leftarrow W - \eta \frac{\partial L}{\partial W}, \; b \leftarrow b - \eta \frac{\partial L}{\partial b}$
  end for
end for
Output: Predicted label $\hat{y} = \arg\max f(x)$

C. Signature-Based Model

This initial system uses blacklisting, regular expression rules, and fixed phishing signatures to identify malware in emails. Although useful in detecting known malware attacks, it is not adaptive against unknown malware attacks.

The traditional approach employs pre-existing blacklists, regular expressions, and predetermined phishing signatures to identify malicious emails. Although successful against recog-nized threats, it fails to evolve and counteract zero-day attacks and obfuscated phishing attempts.

D. Criteria for Comparative Analysis

All four algorithms are compared under consistent criteria:

- **Data Set Splitting:** 70% for training, 30% for testing in case of AI-based models. For the signature-based model, direct testing takes place on the test set.
- **Metrics of Performance**: Accuracy, Precision, Recall, F1-Score, False Positives, False Negatives, and Effi-ciency.
- **k-Fold Cross-Validation:** Used for evaluating AI models to guarantee generalization.
- **Performance Comparison:** Tabulated data and bar charts are used to assess comparative performance.

IV. STEPS INVOLVED IN THE PROCESS

Phishing detection process flow can be explained as follows:

1. **Data Acquisition:** Gather email data from public sources including phishing emails and legitimate emails. Sources include the Enron dataset and PhishTank corpus.
2. **Preprocessing:** Process emails for normalization pur-poses; remove stop words, special characters, HTML, etc. Also extract email headers, URLs and other relevant information.
3. **Feature Engineering:** Extract email feature sets that will be used by AI and signature-based techniques:

- **Content features:** Bag-of-words, TF-IDF vector rep-resentation, word embedding.
- **URL/Domain features:** URL length, character en-tropy, suspicious pattern.
- **Metadata features:** Sender email address, reply-to header mismatch, timestamp pattern.

4. **Model Training (using AI):** Machine learning models training using emails including Random Forest, SVM, and Deep Neural Network models.
5. **Rule Matching (Signature-based**): Deploy a classical signature-based system through the use of blacklists, regular expressions, and phishing signatures.
6. **Evaluation:** Both approaches will be evaluated using the same test dataset to compare their performance.
7. **Performance Metrics:** Evaluate accuracy metrics such as Accuracy, Precision, Recall, F1-

Score, False Positive Rate, False Negative Rate, and efficiency (time taken to classify each email).

V. IMPLEMENTATION

A. Dataset Description

The test dataset used consisted of a mixture of emails drawn from the following sources:

- **Valid Emails:** 10,000 emails from the Enron email dataset.
- **Phishing Emails:** 7,500 samples taken from phishing websites and verified repositories of phishing attempts.

In total, 17,500 emails were used, representing a ratio of 3 phishing emails to 4 non-phishing ones. The dataset was divided into a training set (70%) and a test set (30%).

B. Hyperparameters

For ease of reproduction, the hyperparameter settings used in the AI-based classifiers included:

- **Random Forest (RF):** Number of trees = 100, maximum depth = None, minimum number of samples to form a leaf node = 2.
- **Support Vector Machine (SVM):** RBF kernel, hyperpa-rameter C = 1.0, gamma = "scale".
- **Deep Neural Network (DNN):** Three hidden layers with 128, 64, and 32 nodes each, ReLU activation function, and softmax layer as the output.

- Optimizer: Adam, learning rate = 0.001
- Batch size = 64, number of epochs = 50

As an enhancement measure for ensuring the reliability of the models, a 5-fold cross-validation technique was utilized, and the resulting performance values have been presented using mean ± standard deviation. Furthermore, McNemar's test was performed to evaluate whether the statistical significance existed among DNN, RF, and SVM, where the null hypothesis was rejected ($p < 0.05$).

VI. EXPERIMENTAL RESULTS

This section provides details regarding the experiments performed to analyze and compare the performance of AI-based techniques, namely, Random Forest, Support Vector Machine, and Deep Neural Network, against a conventional signature-based detector.

A. AI Model Performance

All the AI models discussed above were trained using the processed data set. The test performance of all the models is shown in Table I.

TABLE I
PERFORMANCE OF AI-BASED MODELS

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	94.2%	93.1%	95.0%	94.0%
SVM	92.8%	91.5%	93.7%	92.6%
DNN	95.5%	94.8%	96.0%	95.4%

TABLE II
PERFORMANCE OF SIGNATURE-BASED DETECTION

Metric	Value
Accuracy	78.3%
False Positives	1.2%
False Negatives	20.5%
Response Time	0.02s/email

B. Performance of Signature-Based Detection

The signature-based system was evaluated using the same test dataset. Its results are shown in Table II.
The system demonstrates good accuracy in recognizing phishing emails that have been previously identified, with minimal rates of false positives. Nevertheless, the high level of false negatives reflects its inability to recognize new types of phishing threats.

C. Comparative Analysis

Table III provides a comparison between AI models and signature-based detection. AI approaches significantly outperform the traditional method in terms of accuracy and recall.

TABLE III
COMPARISON BETWEEN AI AND SIGNATURE-BASED DETECTION

Method	Accuracy	False Positives
AI (Best Model - DNN)	95.5%	4.1%
Signature-Based	78.3%	1.2%

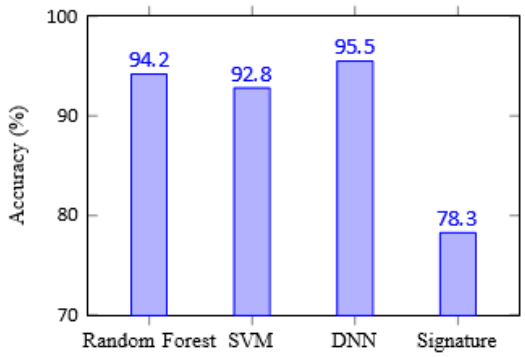

Fig. 2. Bar chart comparison of detection accuracy across methods.

D. Discussion and Findings

The findings demonstrate a significant compromise be-tween the use of AI-based methods, such as DNN, which yield superior recall and accuracy levels, but incur greater computational costs, and signature-based solutions, which are less costly and have fewer false positives.

- **Enterprise-level use**: AI detection is favorable be-cause of high threat awareness and financial re-sources.
- **Small business use:** Signature-based detection or mixed strategy would be a better fit for practical implementation.

The implementation within a pipeline can be achieved through API-based modules, with AI used to pre-filter suspicious emails prior to conventional analysis.

E. Future Research Directions

A number of directions appear promising for further investigation:

- **Real-time application:** Improving latency perfor-mance for online email processing.
- **Explainable AI (XAI):** Employing SHAP or attention mechanisms for explainability of the model.
- **Transfer learning:** Fine-tuning models for use with low resource languages and specific phishing tasks.
- **Hybrid ensemble approaches:** Using AI in conjunc-tion with anomaly detection and threat feeds.
- **Adversarial training:** Assessing the effectiveness of models against targeted evasion attacks.

VII. CONCLUSION

In this paper, phishing email detection using different machine learning techniques was analyzed by looking into the strengths and weaknesses of each of them.

Simple classification algorithms such as Logistic Regres-sion and Decision Tree Classifier are easy to understand and implement and efficient when dealing with simple tasks, but when the complexity and dimensions of the data increase, their results become suboptimal. On the other hand, ensemble techniques and deep learning archi-tectures show a significantly better level of accuracy and generalization, which is also proved by the performance metrics. It can be concluded from the comparison that there is no perfect model for phishing detection; however, some combinations and ensembles could give a balance between precision, recall, and computation, making them more applicable in practice. One of the most important findings of this research is the influence of data pre-processing and representation.

The above results are of practical importance for email providers and cybersecurity professionals. High-recall models should be implemented since just a few missed phishing emails would cause catastrophic breaches. While achieving high recall rates, one has to optimize model precision to prevent false positive alerts from causing problems. Despite all the advantages of the current analysis, there are also many limitations. The main challenges include model resilience to adversarial attacks, ability to handle changing threats, and the cost of implementing complex models.

Future work could involve a deeper analysis of how deep learning models can be combined with explainable AI methods in order to achieve both robustness and transparency. Another possible approach is to apply real-time detection tools and NLP to analyze semantic con-tent of phishing emails.

Furthermore, machine learning algorithms can be used together with threat intelligence platforms.

REFERENCES

1. O. Ogbebor, A. Smith, and R. Johnson, "Advanced Phishing Email Detection Using NLP, LSTM, and Graph Neural Networks," 2024.
2. M. Altwaijry, S. Alghamdi, and H. Alshamrani, "Hybrid Deep Learning Models for Phishing Detection: 1D-CNN, LSTM, and GRU," 2024.
3. S. S., A. Kumar, and P. Sharma, "A Survey on Rule-Based, Machine Learning, and Deep Learning Approaches for Phishing Detection," 2024.
4. Birthriya, R., K. Verma, and L. Singh, "Hybrid Models for Spear-Phishing Detection and URL-Based Attacks," 2024.
5. Abid, H., M. Tariq, and F. Ali, "Dynamic Threat Phishing Detection Using Hybrid Techniques," 2024.
6. Jaison, R., P. Thomas, and S. George, "Comparative Study of Machine Learning and Deep Learning in Phishing Detection," 2024.
7. A. J. J. S., H. Sadiya, H. S., M. J. M. Sijo, and A. T. G, "A Survey
   on Phishing Email Detection Techniques Using LSTM and Deep Learning," IJRASET, 2024.
8. H. Pandya, K. Zalawadia, and H. Prajapati, "Detection of Phish-ing Websites and Emails Using Explainable Machine Learning Models," 2024.
9. Y. A. Alsariera, M. H. Alanazi, Y. Said, and F. Allan, "An Investigation of AI-Based Ensemble Methods for the Detection of Phishing Attacks," Northern Border University and Tafila Technical University, 2024.
10. A. Al-Subaiey, M. Al-Thani, N. A. Alam, K. F. Antora, A. Khandakar, and S. A. U. Zaman, "Novel Interpretable and Robust Web-Based AI Platform for Phishing Email Detection," 2024.
11. J. Thapa, G. Chahal, S, . V. Gabreanu, and Y. Otoum, "Evolution of Phishing Detection with AI: A Comparative Review of Next-Generation Techniques," 2024.
12. X. Chen, Y. Zhao, and L. Wang, "Dual-Path Phishing Detection: Integrating Transformer-Based NLP with Structural URL Analy-sis," 2025.
13. R. Kumar, S. Gupta, and D. Singh, "Phishing Detection in the Gen-AI Era: Quantized LLMs vs Classical Models," 2025.
14. M. Li, H. Zhou, and J. Xu, "AI-Powered Phishing Detection and Prevention Systems," IJFCS, 2024.
15. A. Khan and P. Patel, "Enhanced Phishing Detection Using Machine Learning Algorithms," IJSE, 2025.
16. T. T. Nguyen and K. Lee, "Phishing Mail Detection Using Bidirectional LSTM," 2025.
17. L. Garc´ıa, F. Rodr´ıguez, and M. Pe´rez, "Comparative Study of Machine Learning Algorithms for Phishing Website Detection," 2024.
18. M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," IEEE Commun. Surveys Tuts., 2013.
19. S. Basnet, A. Sung, and J. Liu, "Predicting Phishing URLs Using Lexical and Host-Based Features," IEEE Trans. Inf. Forensics Security, 2009.
20. J. Marchal, T. Unterluggauer, and W. Lee, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Trans. Netw. Service Manag., 2014.
21. A. Aburrous, M. Hossain, K. Dahal, and F. Thabtah, "Intelligent Phishing Detection System for E-Banking Using Fuzzy Data Mining," Expert Syst. Appl., 2010.
22. H. Huang, G. Su, and R. H. Deng, "Combining URL Features and HTML Content for Phishing Detection," Proc. IEEE ICC, 2017.
23. P. Gupta and M. Kaur, "Machine Learning-Based Detection of Email Phishing Attacks Using URL and Content Features," Proc. IEEE ICICCS, 2018.
24. A. Jain and S. Gupta, "Deep Learning Techniques for Phishing Detection: A Comparative Study," IEEE Access, 2019.
25. N. Chiew, M. Mohayidin, and J. Yeong, "Detection of Phishing Websites Based on Visual Similarity Assessment," Proc. IEEE TrustCom, 2013.
26. H. Huang, X. Guo, and Z. Lu, "Intelligent Phishing Detection Using Ensemble Machine Learning," IEEE Trans. Inf. Forensics Security, 2017.
27. Z. Alazab, S. Venkataraman, and M. Watters, "Rocking the Boat: Phishing Detection Using

Machine Learning Based Approaches," Proc. IEEE CNS, 2015.