# Chapter 2: LITERATURE REVIEW



## Introduction

**Social engineering** is a psychological manipulation technique where cybercriminals exploit human trust, fear, or urgency to deceive individuals into performing actions that compromise security, such as disclosing confidential information or enabling system access. Instead of relying on traditional hacking tools to exploit software or hardware vulnerabilities, social engineering exploits human behavior, making it one of the most dangerous and successful forms of attack in the modern threat landscape. Techniques include phishing, baiting, pretexting, vishing, and quid pro quo.

Among these techniques, **phishing remains the most widespread and damaging form of social engineering attack**, and its prevalence has been supported by numerous global studies and industry analyses. Research shows that phishing is consistently ranked as the most common social engineering threat due to its effectiveness, ease of execution, and adaptability across communication channels (Alzahrani et al., 2021; Basit et al., 2022). Attackers exploit digital platforms, particularly email, to impersonate legitimate institutions and deceive users into providing sensitive information.

What sets phishing apart from other social engineering methods is its ability to scale rapidly and target both individuals and organizations with minimal resources. Unlike baiting or pretexting, phishing campaigns can be automated and distributed widely using email bots and phishing kits available on underground forums. These features make phishing a recurring top threat in annual cybersecurity threat reports by multiple academic and professional organizations.

The continued evolution of phishing tactics, from generic spam to personalized spear phishing and smishing (SMS phishing), has contributed to its resilience and growth. This explains why, compared to other forms of social engineering attacks, phishing is the focus of this research. It provides not only a well-documented attack vector but also a diverse set of datasets and practical applications for testing AI-based detection models.

As the complexity and frequency of phishing attacks grow, the need for intelligent, adaptive defense systems becomes critical. Traditional security measures often fall short, especially against zero-day phishing attacks or personalized spear phishing. This has led to a strong shift toward **Artificial Intelligence (AI)**, specifically **Machine Learning (ML)**, **Deep Learning (DL)**, and **Reinforcement Learning (RL)**, as a robust solution for phishing detection. AI-based models can process large volumes of data, identify hidden patterns, and adapt over time, enabling more accurate and timely threat detection.

Given phishing's prominence and evolving nature, this study focuses exclusively on AI-based phishing detection. Compared to other forms of social engineering, phishing offers both a greater volume of research data and clearer use cases for AI implementation, making it an ideal focal point for academic and practical analysis. The subsequent sections provide a deep dive into AI techniques, available software tools, datasets, and recent research that contribute to building effective, intelligent phishing detection systems.



Figure 1: Literature Review Structure Diagram

## Related Work

Recent advancements between 2020 and 2024 have demonstrated the growing impact of AI in combating phishing threats. Researchers have developed increasingly sophisticated models leveraging ML, DL, and RL to classify, detect, and respond to phishing activities in real time.

Rahman et al. (2022) introduced a hybrid model combining Decision Trees and anomaly detection, achieving improved phishing email detection rates in enterprise environments. Their work highlighted the effectiveness of ensemble methods in capturing varied attack patterns. Aljohani and Hossain (2021) employed the BERT transformer model to detect phishing content by analyzing contextual semantics within emails. Their results showed significant accuracy improvements over traditional ML models.

Zhao et al. (2020) used Long Short-Term Memory (LSTM) networks to detect phishing attempts in mobile text messages. LSTM’s ability to process sequential information proved critical in identifying suspicious messages that mimic legitimate communications. Nasir et al. (2023) proposed a CNN-based phishing detection system that integrated user behavior logs with content analysis. The combination of behavioral and textual features enhanced detection performance, particularly for spear-phishing cases.

Asker and Essa (2024) emphasized the potential of unsupervised models, such as Autoencoders, to identify phishing emails without labelled data. Their findings suggest these models are well-suited for environments where phishing strategies evolve rapidly and labelled examples are scarce. Zhang et al. (2023) explored the use of Reinforcement Learning to create adaptive phishing detection systems. Their RL-based framework optimized decision-making policies in real-time, improving resilience against evolving attack vectors while minimizing false positives.

These studies demonstrate a transition from static rule-based systems to adaptive, intelligent models that integrate language understanding, behavior analysis, and real-time learning. They underscore the importance of hybrid approaches and model interpretability in building effective phishing detection solutions.

### Phishing

Phishing is a cyber attack technique in which malicious actors impersonate trustworthy entities to deceive victims into revealing sensitive information, such as usernames, passwords, credit card details, or login credentials. These attacks often take the form of emails, websites, or messages that closely resemble legitimate communication from known organizations. The purpose of phishing is typically to commit fraud, gain unauthorized access to systems, or install malware.

Phishing continues to be the most prevalent type of social engineering attack globally. Its popularity among attackers stems from its high success rate, scalability, and ability to exploit human behavior rather than technical vulnerabilities. According to the 2023 Verizon Data Breach Investigations Report, phishing accounted for more than one-third of all social engineering-related incidents. The Anti-Phishing Working Group (APWG) also reported over 1.2 million phishing attacks in a single quarter, indicating a steady and alarming upward trend. These attacks have evolved from generic bulk emails to more targeted approaches such as spear phishing, business email compromise (BEC), and smishing (SMS phishing), which are increasingly difficult to detect. Figure below shows that Pretexting is now more prevalent than Phishing in Social Engineering incidents. However, when we look at confirmed breaches, Phishing is still on top. (Verizon, 2023)



**Figure 2: Action varieties in Social Engineering incidents (Verizon, 2023)**

Due to the sophistication and frequency of phishing, AI-based techniques have become central to its detection. Machine Learning approaches such as Support Vector Machine (SVM) and Random Forest are commonly used due to their strong classification capabilities on structured datasets. These models analyze features such as email headers, URL characteristics, and content patterns., models like Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT) have proven effective in understanding the semantic structure of emails, enabling the identification of sophisticated phishing attempts that evade traditional filters.

Recently, Reinforcement Learning (RL) has emerged as a powerful approach for dynamic and adaptive phishing detection. Unlike ML and DL, which learn from static datasets, RL enables an agent to learn optimal policies through interactions with an environment, receiving feedback in the form of rewards or penalties. Techniques such as Q-learning, Deep Q-Networks (DQN), and Policy Gradient Methods are used to develop adaptive systems that can fine-tune detection thresholds based on evolving user behavior and attack strategies. These RL-based methods are especially useful in real-time systems, where phishing tactics change frequently and require continuous learning to maintain detection accuracy.

#### Machine Learning Approaches

##### Support Vector Machines (SVM)

Support Vector Machine (SVM) model as a hyper plane in which several classes are represented. So that we may reduce the amount of mistakes we make, SVM will create the hyper plane in an iterative fashion. SVM's purpose is to classify datasets such that a maximum marginal hyper plane may be discovered (Jain & Gupta, 2023).

The high accuracy of SVMs in phishing detection has been demonstrated in various studies. For instance, research by Gupta et al. (2023) highlights that SVMs, when properly tuned, can achieve remarkable detection rates and low false positive rates, making them a reliable choice for cybersecurity applications. However, the effectiveness of SVMs comes with certain challenges. The considerable parameter adjustment necessary to maximize the model's performance is one of its primary drawbacks. It can take some time to carefully choose parameters like the kernel parameters and the penalty parameter (C) to balance the choices between variance and bias.

The computational expense associated with Support Vector Machines (SVMs), particularly when using non-linear kernels, can be significant. Training an SVM model involves solving a complex optimization problem, which can be computationally intensive, especially when dealing with large datasets typically encountered in phishing detection systems. This computational burden can limit the practicality of SVMs in real-time phishing detection, where fast processing is crucial to ensure timely responses and effective protection. Despite these challenges, the high precision and robustness of SVMs in identifying complex patterns in data make them a valuable asset in phishing detection frameworks.

To mitigate these limitations, hybrid approaches that combine SVMs with other machine learning techniques have been explored. These methods aim to capitalize on the strengths of SVMs while addressing their computational drawbacks. For example, integrating SVMs with feature selection techniques can reduce data dimensionality, lessening the computational load and improving real-time applicability. As cyber threats continue to evolve, the role of SVMs in phishing detection remains critical, providing a balance of high accuracy and adaptability necessary for effective cybersecurity defences.



Figure 2: SVM algorithm (Jain & Gupta, 2023)

###### Attributes

To ensure that examples from different classes are as far apart as feasible, the SVM model represents examples as points in space. After that, fresh instances are shown within this identical area and categorized according to their placement along the margin (Patil et al., 2017). The following is a list of the attributes utilized for SVM classification:

|  |  |  |
| --- | --- | --- |
| Sr No | Features | Significance |
| 1 | Having IP Addr | If a domain name contains an IP address, the website is likely to be phishing. |
| 2 | URL\_Length | Legitimate URLs are typically around 75 characters long. URLs longer than 75 characters are likely to be phishing sites. |
| 3 | Shortening Service | Link softeners are used to deceive people. |
| 4 | Having\_At\_Symbol | Websites that contain an @ symbol are often considered suspicious. |
| 5 | Double slash redirecting | If a URL includes '/1', it may be classified as a phishing website. |
| 6 | Having Sub Domain | Usually, legitimate websites have a domain structure that has two levels. Since they could have multiple domains within a single domain, websites with more than three dots are frequently phishing sites. |
| 7 | URL of Anchor | The link tag on legitimate websites points to the same domain as the source code. Phishing websites, on the other hand, frequently have links to other domains. |
| 8 | Links in tags | Tag links could take you to fraudulent websites. |
| 9 | Abnormal URL | The primary identifier of legitimate websites is found in the URL; this element was taken from the Who is Database. |
| 10 | Age of domain | Websites that are older than six months are considered Phishy; those that are older than this are legitimate. Websites that are older than six months are considered Phishy; those that are older than this are legitimate. |
| 11 | Page Rank | Phishing websites usually don't have many links leading to them, so they have a poor page rank. |
| 12 | Links Pointing to page | Phishing websites often include links to zip files that automatically download malware. |

**Table 1: SVM Algorithm Attributes (Patil et al., 2017)**

##### Random Forest

Random Forest is a well-known machine learning method that performs excellently when dealing with regression and classification problems. It creates a lot of decision trees during training, outputs the mode of classes for classification tasks or the mean prediction for regression tasks and increases accuracy and generalization ability by voting or averaging over all trees. Because of the ensemble technique, the model resists overfitting. Because Random Forest can handle big datasets with high dimensionality and is flexible and effective, it is widely employed. It is also resistant to noise and outliers and provides insights into feature importance. Visualization techniques can aid in model evaluation and decision-making (Gunjan & Prasad, 2024)

###### Random Forest Technique

Random Forest (RF) is a powerful ensemble learning algorithm that has shown strong performance in phishing detection due to its ability to handle complex patterns and noisy data. It operates by building multiple decision trees using different random subsets of the training dataset—a process known as bootstrapping. Each tree is trained independently, which introduces diversity and reduces the risk of overfitting. This is especially beneficial in phishing detection, where malicious patterns can be subtle and varied across different instances. Once the forest is constructed, each decision tree contributes to the prediction process. In classification tasks such as distinguishing between phishing and legitimate activities, each tree casts a "vote" for a class label. The final decision is based on majority voting, where the class with the most votes becomes the output. This collective decision-making mechanism improves both accuracy and stability, making Random Forest well-suited for phishing detection systems that require consistent performance under diverse threat scenarios. Refer to the figure below, which illustrates how multiple decision trees independently classify an instance and then combine their outputs through majority voting to determine the final class.



**Figure 3: Random Forest Algorithm**

#### Deep Learning Approaches

##### Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to effectively capture long-term dependencies in sequential data. In the context of phishing website detection, LSTM plays a crucial role in modeling temporal patterns and contextual features that may not be apparent in static data. As described by Elberri et al. (2024), LSTM was integrated with a Convolutional Neural Network (CNN) in a hybrid deep learning architecture aimed at improving phishing detection accuracy. While CNN excels at extracting spatial features from grayscale images generated from URL and web content data, LSTM contributes by analyzing the sequential behavior and dependencies within the data. This combination allows the model to better understand complex relationships, such as those found in URL structures and embedded script behavior typical of phishing attacks. The study demonstrated that the CNN-LSTM hybrid model outperformed standalone CNN and LSTM models in terms of accuracy, sensitivity, and precision, highlighting LSTM’s effectiveness in enhancing the classification of phishing and legitimate websites.

###### Structure of an LSTM neural network cell

LSTM is a variant of the RNN deep learning architecture designed specifically for tasks like time series analysis and classification. LSTM effectively uses a gating mechanism to deal with vanishing gradient problems in the training process.The LSTM memory cell has four gates named forgetting f, input gate i, control gate c, and output gate o. The fundamental configuration of the LSTM cell is presented in Figure 4 and it consists of the output of the previous memory cell Ct-1. (Elberri et al., 2024)

This neural network uses components such as the input signal at each time step Xt, the current memory cell Ct output, the previously hidden unit Ht − 1, and the currently hidden unit Ht. The forget gate determines the way in which the contribution from the previous time step is incorporated, resulting in a value ranging from zero to one for each datapoint in Ct-1. The input gate regulates the amount of input that is stored in the memory cell from the current time step. Meanwhile, the control gate updates the memory cell contents from Ct-1 to Ct. The output gate dictates the extent to which the internal state influences the external state at the current time step. The symbol ⊗ represents the element-wise multiplication of vector elements, while ⊕ signifies the summation of vector along with the application of the σ (sigma) function. To formulate the LSTM artificial neural network. (Elberri et al., 2024)

Diagram of a diagram of a cell state

AI-generated content may be incorrect.

**Figure 4: Structure of an LSTM neural network cell (Elberri et al., 2024)**

A math equations with black text

AI-generated content may be incorrect.

**Figure 5: Equations of LSTM (Elberri et al., 2024)**

##### Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a powerful deep learning model that has shown great effectiveness in detecting phishing attacks, particularly those involving social engineering through text-based communication. Phishing often relies on carefully crafted language to manipulate users into revealing sensitive information or clicking malicious links. BERT’s unique bidirectional architecture allows it to analyze both the left and right context of each word in a sentence, enabling it to detect subtle linguistic patterns and deceptive cues used in phishing attempts. In recent studies, BERT has been applied to phishing detection by transforming message content into 768-dimensional contextual embeddings using the [CLS] token, which captures the overall semantics of the text. These embeddings, when combined with other extracted features such as URLs, email addresses, or phone numbers, are used as inputs for neural network classifiers to distinguish phishing messages from legitimate ones. Compared to traditional text classification techniques, BERT has demonstrated superior performance in identifying context-based phishing strategies, making it a valuable AI tool for building intelligent systems that can proactively detect and prevent social engineering-based cyber threats.



**Figure 6: Transformer Model Architecture (Jain et al., 2025)**

#### Reinforcement Learning Approaches

## Analysis of Datasets

On the availability and quality of datasets. Each type of attack presents unique challenges in dataset collection and labeling, which in turn influences model performance and generalizability.

For phishing detection, widely-used datasets include the UCI Phishing Websites Dataset and Kaggle’s Phishing Website Dataset. These datasets often contain features such as URL structure, presence of IP addresses, HTML content analysis, and domain-based indicators (Abdelhamid et al., 2014). Additionally, the Email Spam Dataset and Enron Dataset are frequently employed for identifying malicious email characteristics, aiding both phishing and spam detection models (Chandrasekaran et al., 2006).

Spam detection leverages datasets like the Ling-Spam Corpus and the SpamAssassin Public Corpus. These datasets provide labeled examples of spam and ham emails, enabling supervised learning approaches. Features typically include word frequency, n-gram analysis, and syntactic patterns derived from email content and headers (Delany et al., 2012).

Baiting datasets are comparatively scarce, often requiring simulation or aggregation from broader security datasets. Researchers generate synthetic datasets using logs of user interactions, such as clickstreams, download histories, or device connection patterns (Kumar & Sharma, 2021). In some cases, honeypots are deployed to collect real-world baiting behavior data. Anomaly Detection models use these behavioral logs to identify deviations from normal patterns, indicating potential baiting scenarios.

Despite progress, many datasets lack standardization and are often imbalanced, which affects model training. Ongoing efforts are needed to develop comprehensive, diverse, and realistic datasets for all social engineering attack types to improve AI model robustness and transferability across different domains.

## Critical Review ( kene ada features, math equation, justification)

A growing body of research has been conducted to evaluate and benchmark the performance of AI models in detecting social engineering attacks. Numerous studies have compared traditional ML models with advanced DL techniques, highlighting the trade-offs between interpretability, accuracy, and computational efficiency. Research paper focusing on phishing detection have shown that while traditional model like SVM and Naïve Bayes provide fast and interpretable results, deep models like BERT offer superior accuracy and contextual understanding. In the domain of baiting detection, unsupervised approaches such as Autoencoders have demonstrated promising results, especially in scenarios where labeled data is scarce. However, challenges remain in generalizing models across different attack types and adapting to evolving tactics used by attackers. Future research should focus on hybrid models that combine the strengths of both ML and DL, and the development of explainable AI techniques to ensure transparency and trustworthiness in automated detection systems.

## Project Solution (kene tahu algorithm, study algorithm)

## Summary