# Chapter 2: LITERATURE REVIEW



## Introduction

Social engineering attacks represent a formidable and persistent threat in today’s cybersecurity landscape. Unlike traditional cyberattacks that focus on exploiting system vulnerabilities or breaking through firewalls and encryption, social engineering manipulates human behavior to gain unauthorized access to sensitive information or systems. These attacks rely heavily on psychological manipulation rather than technical sophistication, making them particularly insidious and difficult to detect using conventional security measures. Common social engineering tactics include phishing, baiting, pretexting, quid pro quo, and tailgating. Each method exploits human trust, curiosity, or fear of deceiving individuals into revealing confidential data or performing actions that compromise organizational security.

As these attacks become increasingly sophisticated and adaptive, there is an urgent need for equally adaptive and intelligent detection techniques. Artificial Intelligence (AI), particularly through Machine Learning (ML) and Deep Learning (DL), has emerged as a powerful solution to this problem. AI models excel at learning from vast and diverse datasets, identifying hidden patterns, and making real-time decisions that would be impossible for static rule-based systems. ML and DL approaches can be tailored to recognize the unique characteristics of different types of social engineering attacks, from analyzing linguistic patterns in phishing emails to detecting abnormal behavior associated with baiting attacks. This chapter presents a comprehensive overview of AI-based approaches for detecting social engineering attacks, focusing on the classification of attacks, the learning paradigms employed, and the specific AI models and algorithms utilized. Furthermore, it includes a critical analysis of the current research landscape, identifying key gaps and potential areas for future development.



Figure 1: Literature Review Structure Diagram

## Artificial Intelligence Social Engineering Detection Techniques

Artificial Intelligence (AI) offers a dynamic and scalable approach to combatting social engineering attacks by leveraging advanced learning algorithms that can detect subtle behavioral and linguistic cues indicative of malicious intent. The detection techniques for various types of social engineering attacks are typically categorized into two major branches: Machine Learning Approaches and Deep Learning Approaches. These methods are applied based on the nature of the attack, data availability, and required detection complexity. The following subsections break down these techniques according to specific types of social engineering attacks.

### Phishing

Phishing attacks, which aim to deceive users into divulging sensitive information via deceptive emails or websites, are among the most researched in the domain of AI-based 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. On the Deep Learning side, 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.

#### 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**

#### 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 the 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) in phishing

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)**

### Spam

Spam detection, while traditionally considered a simpler problem, continues to benefit from advancements in AI. Machine Learning methods such as Naïve Bayes and Logistic Regression are efficient in filtering spam based on word frequencies and statistical patterns. Deep Learning models, including Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), offer improved accuracy by learning contextual relationships and recognizing patterns in unstructured data. These models are particularly effective in environments where spam messages constantly evolve in form and content.

#### Naïve Bayes

Naïve Bayes spam detection is a common model to detect spam, but it is mostly used with optimization. Naïve Bayes must be trained with controlled data that is already defined as spam or ham so the model can be applied to real world situations. Naïve Bayes also assumes that the features that it is classifying such as individual words in an email are independent from each other. A filter is created using training data that has been labeled ham or spam. Then the model signs the probability that each feature is in spam. The probabilities are written as values between 0 and 1. The filter will then evaluate whether an email is spam based on the individual probabilities of each word. The advantages of Naïve Bayes are that it is a very simple algorithm that performs equally well against more complex classifiers. It also does not classify the email from one or two words but considers every single relevant word. (Spain, 2019)

#### Logistic Regression

Logistic Regression is a commonly applied machine learning algorithm for handling binary classification tasks, such as spam detection. In this area, the algorithm determines whether a message should be classified as “spam” or “not spam” by identifying patterns within the text. It operates by estimating the likelihood that a particular input belongs to one of the two classes based on selected features, which may include specific keywords, the frequency of punctuation usage, or the inclusion of links in the content. While Logistic Regression is considered a relatively simple model when compared to more advanced approaches, it remains highly effective, particularly when paired with appropriate preprocessing techniques like text tokenization and TF-IDF for feature extraction. Among its key strengths are fast training speed, low computational resource requirements, and ease of interpretation, which make it easier for researchers to understand how each feature affects the classification outcome. As a result, Logistic Regression is often used as a foundational model in spam detection systems and continues to deliver dependable results across various filtering applications.

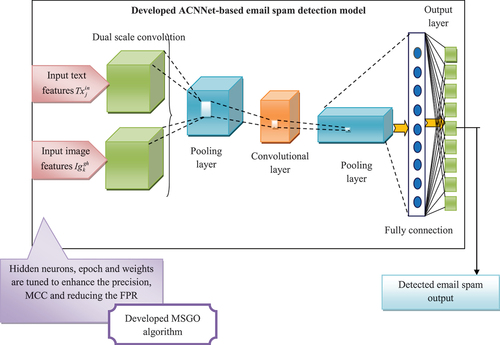
One of the major strengths of Logistic Regression is its high level of interpretability. The model assigns weight coefficients to input features, enabling analysts to assess the influence of each feature on the final prediction. This characteristic is especially important in spam mail detection, as it provides clarity on which words, expressions, or metadata elements are closely linked to spam content. Such interpretability proves valuable for email security professionals who need to continuously adjust spam filters in response to evolving spam strategies. This adaptability is particularly beneficial in scenarios where the distinction between spam and legitimate messages is complex and cannot be captured by a straightforward linear decision boundary. Logistic Regression also supports enhancements like regularization, which mitigates overfitting, and feature selection, which helps pinpoint the most relevant inputs for classification. These enhancements contribute to its effectiveness in refining spam detection models and improving prediction accuracy. (Shreenithi et al., 2023)

#### Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a type of artificial neural network that is specifically designed to work with sequential or time-series data. Unlike traditional feedforward neural networks, RNNs have internal memory that allows them to retain information from previous inputs in the sequence. This feature makes RNNs highly effective for tasks where the order of data matters, such as text classification, language modeling, and spam detection. In the context of phishing or spam email detection, RNNs can analyze the flow of words or phrases and capture patterns that are commonly used in deceptive messages. The ability to learn from past inputs helps RNNs make more accurate predictions when dealing with complex and context-dependent text data.

#### Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning model that has shown promising results in various text classification tasks, including spam detection. Although CNNs are widely known for image processing, they can also be applied to textual data by treating the input text as a sequence of word embeddings. In the context of spam detection, CNNs work by applying filters to detect local patterns within messages, such as commonly used spam phrases, suspicious keywords, or repetitive structures. These patterns are often indicators of spam, and the CNN model can learn to recognize them during training. One of the advantages of using CNNs for spam classification is that they can automatically extract important features from the text without manual intervention, making them efficient for large-scale email filtering. By identifying key textual cues, CNNs can accurately classify whether a message is spam or not, contributing to improved security in digital communication systems. The figure below is one of the examples of CNN email detection model.



**Figure 7: CNN architecture model (Kadam et al., 2024)**

### Baiting

Baiting involves luring individuals with the promise of a reward or desirable content, often to trick them into downloading malware or revealing credentials. Machine Learning techniques like K-Means clustering and Anomaly Detection are used to analyze user behavior, identifying unusual patterns such as unexpected file downloads or device interactions. Deep Learning models, including BERT and Autoencoders, enhance detection by learning high-dimensional representations of user activity, allowing for the recognition of complex anomalies that might indicate a baiting attempt.

#### K-Means

K-Means clustering is an unsupervised machine learning algorithm that helps group similar data points into clusters based on shared characteristics. When applied to bait detection, this method can be used to uncover hidden patterns or unusual behaviors in phishing or suspicious messages—without the need for labeled data. The algorithm works by assigning each message to one of several clusters, where messages within the same cluster are closely related, while those in different clusters are more distinct. This makes K-Means a practical approach for identifying bait messages, which often share common traits in their wording, structure, or the links they include. By analyzing features such as keyword usage, URL presence, or text length, K-Means can effectively separate potential bait messages from genuine ones. In addition, it is relatively fast and straightforward to implement, which makes it especially useful when working with large volumes of unlabeled message data where manual review is not feasible.

##### K-Means algorithm

K-Means algorithm based on dividing is a kind of cluster algorithm, and it is proposed by J.B.MacQueen. This algorithm which is unsupervised is usually used in data mining and pattern recognition. Aiming at minimizing cluster performance index, square-error and error criterion are foundations of this algorithm. To seek the optimalizing outcome, this algorithm tries to find K divisions to satisfy a certain criterion. Firstly, choose some dots to represent the initial cluster focal points(usually, we choose the first K sample dots of income to represent the initial cluster focal point); secondly, gather the remaining sample dots to their focal points in accordance with the criterion of minimum distance, then we will get the initial classification, and if the classification if unreasonable, we will modify it(calculate each cluster focal points again), iterate repetitively till we get a reasonable classification. K-Means algorithm based on dividing is a kind of cluster algorithm, and has advantages of briefness, efficiency and celerity. However, this algorithm depends quite much on initial dots and the difference in choosing initial samples which always leads to different outcomes. What’s more, this algorithm based on target function always uses gradient method to get extremum. The direction of search in gradient method is always along the direction in which energy decreases, which will lead to the fact that when the initial cluster focal point is not proper, and then the whole algorithm will easily sink into local minimum point (Li & Wu, 2012).

#### Anomaly Detection

Anomaly detection is a machine learning technique used to identify data points or patterns that lie considerably outside the norm. In baiting attacks, anomaly detection models are useful because the bait messages normally contain subtle differences that set them apart from regular communication, such as unusual language use, unexpected URLs, or questionable sender behavior. Unlike standard classification models, which require examples of both labeled normal and bait messages, anomaly detection focuses on learning a representation of what "normal" looks like and then alarms on anything that does not appear normal. This is especially useful for real-world use cases where new bait tactics or emerging techniques have yet to be labeled or identified. By modeling features such as message length, keyword distribution, and whether URLs are present, the model detects outliers that may be a baiting attempt. The approach is particularly suitable for imbalanced data sets, where bait messages are few compared to legitimate messages.

Also referred to as outlier detection, [anomaly detection](https://www.kdnuggets.com/2019/10/anomaly-detection-explained.html) is simply the mode of detecting and identifying anomalous data in any data-based event or observation that differs majorly from the rest of the data.

A graph showing anomaly and time

AI-generated content may be incorrect.

**Figure 8: Anomaly detection example diagram**

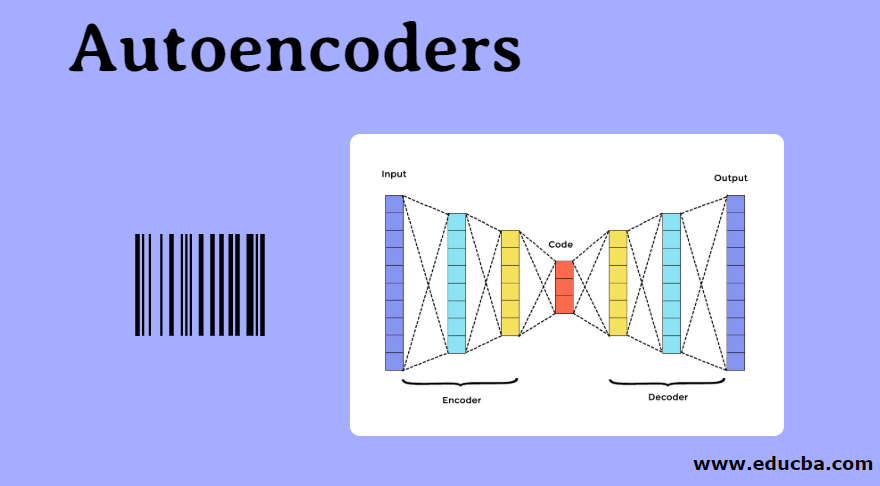
A diagram of a graph

AI-generated content may be incorrect.

**Figure 9: Example of anomaly detection (outlier)**

#### Autoencoders

Autoencoders are neural networks for unsupervised learning that are designed to learn good representations of data by compressing the input data and reconstructing it. They consist of two main elements: an encoder that transforms the input data to a lower-dimension latent space, and a decoder that attempts to reproduce the original data from this compressed representation. In the case of bait detection, autoencoders are particularly helpful in spotting suspicious or abnormal messages. When the model is trained on normal (non-bait) messages, it can reconstruct normal message patterns effectively. However, when an example bait message is given—typically with unusual structures, suspicious links, or misleading wording—the autoencoder cannot reconstruct it well. This generates a large reconstruction error, which can be used as a cue to label the message suspicious. Since autoencoders do not require labeled bait data, they are perfectly suited for use in real-world scenarios where new or novel bait techniques may not have been encountered before. Their ability to learn from normal patterns and detect anomalies makes them a useful tool for enhancing bait detection systems in cyber security.



**Figure 10: Autoencoders architecture diagram (Autoencoders | Main Components and Architecture of Autoencoder, 2019)**

### Critical Review