Development of the social engineering attack models*

Oleksandr Bokhonko[1,†], Sergii Lysenko[1,*,†] and Piotr Gaj[2,†]

[1] *Khmelnitsky National University, Khmelnitsky, Instytutska street 11, 29016, Ukraine*
[2] *Silesian University of Technology, ul. Akademicka 2A, 44-100 Gliwice, Poland*

Abstract

This study developed specialized models for detecting social engineering attacks, with a focus on spam emails, spear phishing, and trojan emails. Each model captures distinct features of these attacks using machine learning-based detection processes. Utilizing the BotGRABBER framework, which incorporates algorithms such as random forest, decision tree, K-nearest neighbor, and XGBoost, the models analyze characteristics like email metadata, user interaction patterns, attachment behaviors, and network anomalies to differentiate between malicious and legitimate communications. The targeted approach of each model enables tailored detection strategies that address specific social engineering tactics, whether they involve spam, personalized deceptive emails, or malware-infected attachments. For example, the trojan email model concentrates on identifying embedded malware within email attachments, utilizing sandbox environments for controlled testing and analysis. In contrast, the spear phishing model focuses on detecting personalized attack methods by analyzing sender details and links for suspicious patterns. The spam email model, on the other hand, prioritizes content filtering and tracking calls-to-action to distinguish between legitimate emails and mass-distributed spam. Empirical results demonstrate the models' effectiveness, achieving approximately 99% detection accuracy with a 6% false positive rate. This strong performance highlights the potential of these models to contribute to proactive defense strategies against evolving social engineering threats. By leveraging targeted feature sets and adaptive machine learning algorithms, these models can be effectively deployed in real-world environments to safeguard networks and systems from a wide array of social engineering attacks.

Keywords
social engineering attack, cybersecurity, models; cyberattacks; detection; network host

1. Introduction

A social engineering attack is a type of cyberattack that relies on human interaction and psychological manipulation to trick individuals into revealing confidential information or performing actions that compromise security. Unlike technical hacking methods, social engineering exploits human behavior and trust to gain unauthorized access, bypass security measures, or install malicious software [1]. Attackers often pose as trusted figures, such as employees, IT support, or even friends, to lower a person's guard and gain access to sensitive information. Such attacks exploit emotions such as curiosity, fear, urgency, or helpfulness. Attackers might, for example, create a sense of urgency to prompt immediate action without verification [2]. Social engineering does not rely on code manipulation or exploiting software vulnerabilities. Instead, it leverages human psychology as the "weakest link" in security [3].

Social engineering attacks are particularly effective because they exploit human psychology rather than technology [4]. Since people are typically more inclined to trust or respond to authority and act under pressure, these attacks often bypass traditional security defenses. This is why training

AdvAIT-2024: 1st International Workshop on Advanced Applied Information Technologies, December 5, 2024, Khmelnytskyi, Ukraine - Zilina, Slovakia
* Corresponding author.
† These authors contributed equally.
booweb24@gmail.com (O. Bokhonko); sirogyk@ukr.net (S. Lysenko); piotr.gaj@polsl.pl (P. Gaj)
0000-0002-7228-9195 (O. Bokhonko); 0000-0001-7243-8747 (S. Lysenko); 0000-0002-2291-7341 (P. Gaj)

and awareness programs are critical in defending against social engineering. Developing models to detect and respond to these attacks is a crucial area of cybersecurity [5].

Social engineering attacks have unique strategies to exploit human psychology and trust for unauthorized access or data extraction [6]. As these attacks continuously evolve in complexity and sophistication, there is an urgent need to model and analyze these threats to devise effective detection techniques [7]. Social engineering attacks exploit human factors, often bypassing traditional security controls; thus, enhancing detection strategies tailored to identify these behaviors is crucial. Each social engineering attack represents a unique set of tactics that adversaries can adapt over time [8, 9]. The frequency and evolving nature of phishing and spear phishing, for instance, indicate attackers' ability to customize messages to specific individuals or roles. Detection techniques must account for these evolutions to stay ahead of attackers [10].

Since social engineering primarily manipulates psychological factors, traditional security mechanisms often fail to detect it [11].

So, building comprehensive models for each attack type can provide a foundation for more effective detection. With effective models, it is possible to move towards predictive security, identifying potential attack patterns before they fully develop.

By structuring detection techniques around a clear understanding of each social engineering attack vector, organizations can build resilient defenses against both traditional and emerging social engineering strategies. This would provide an essential layer of security, strengthening overall cybersecurity frameworks.

2. State-of the-art

There is a huge number of researches devoted to the problem of the social engineering attack detection.

Thus, in article [12] by the author suggested phishing detection method called Freeze-Phish, which uses Python to create a web crawler to collect information such as hyperlinks from a website. In addition, the author created a database of brand words and suspicious words by editing distance algorithms such as Levenshtein distance and Hamming distance to compare the difference between the words in the web page URL and the suspicious word. The developer used a neural network to train this model and exported the code as an executable file (.exe) so that users can more easily use the code to detect suspicious web pages. Compared to other methods, the accuracy of the Freeze-Phish model is about 97% true positives, and the average execution time is 21.3 seconds.

The work [13] presents a new phishing detection model that uses feature selection to select highly correlated features with a class label. The feature selection step uses a library of independent significant features from MATLAB and a heatmap from Python to find highly correlated features. The model uses an adaptive boosting approach that consists of multiple classifiers to improve the accuracy of the model. The model proposed by the authors provides extremely high predictive accuracy of approximately 99%.

A malicious URL (or) a malicious website is a common and serious cyber security threat. Therefore, the search engine becomes the basis of information management. Most existing systems for detecting malicious websites focus on specific attacks. Meanwhile, blacklist-based browser extensions are powerless against numerous websites. Therefore, it is important that any data coming from the client side is effectively obfuscated so that the server cannot interpret any valuable information from the obfuscated data. In paper [4], the first PPSB service is proposed. It provides strong security guarantees that are lacking in existing SB services. In particular, it inherits the ability to detect dangerous URLs while protecting both the user's privacy (browsing history) and the proprietary assets of the blacklist provider (the list of dangerous URLs). The authors propose a model that encrypts sensitive user data to prevent interference by external analysts and service providers. It also fully supports selective aggregate functions for analyzing user behavior online and guarantees differential privacy. The RSA homomorphic algorithm is used to encrypt user behavior data online.

The implementation is complete and its performance is evaluated against a real-time behavioral data set.

In this study [5], the authors proposed an adaptive framework that combines deep learning and Randon Forest for image reading, speech synthesis from deeply faked videos, and natural language processing at different prediction levels to significantly improve the performance of machine learning models for detecting phishing attacks. To validate both the effectiveness and adaptability of our proposed framework to overcome the limitations of current approaches and its ability to detect sophisticated phishing sites, the researchers created 4 categories of phishing sites and uploaded them to a secure server with compromised DNS at a friendly URL; the first was a text-only phishing site, an image-only phishing site, a video-only phishing site, and a combination phishing site. The authors used SEO-friendly URLs and hacked the legitimate DNS on the text-only phishing site so that they could avoid detection at the 1st level to the 4th level of the framework where they were detected. Also, the developers created phishing sites where the text contains only image format, text-only format and video-only format using fake videos to test the adaptability of the proposed structure to different scenarios of a complex or complex phishing site, the proposed structure successfully overcomes the limitations of existing approaches, greatly improves the detection of phishing attacks and successfully detect sophisticated phishing web pages with multi-dimensional fake videos, images and texts.

This study [6] addresses limitations in existing research, such as reliance on proprietary datasets and lack of real-world application, by proposing a highly efficient machine learning model for email classification. Using the most complete and largest publicly available data set, the model achieves an f1 of 0.99 and is designed to be deployed in appropriate applications. Additionally, Explainable AI (XAI) is integrated to increase user trust. This research offers a practical and highly accurate solution that helps fight phishing by providing users with a real-time web application to detect phishing emails.

The work presented in source [7] aims to protect users' e-mail structure and settings to prevent attackers from using the account when it is hacked or hijacked and to prevent them from setting up forwarding in the victim's e-mail account to another account, which automatically stops the user from receiving emails. Secure code is applied to the submit button of the composition to reduce insider impersonation attack. In addition, to protect open applications on public and private devices.

Article [8] provides an overview of the revolutionary technology often referred to as the "Guardian of Artificial Intelligence". It is a technologically advanced strategy to combat social engineering attacks using artificial intelligence (AI). The method uses machine intelligence technologies such as behavioral pattern analysis, anomaly detection, and social engineering deception to perform real-time monitoring actions. Using artificial intelligence functions in cyber defense, this method emphasizes a proactive and adaptive methodology to increase the level of security and immunity from social engineering attacks. While conventional social engineering defenses have shown some success, they rely heavily on static rules and signatures, making it difficult for them to keep up with the rapidly evolving tricks of cybercriminals. Social engineering attacks have become more sophisticated and targeted, requiring organizations to go beyond layered defenses and equip themselves with more advanced and adaptive security tools such as machine learning-based detection and behavioral analytics tools to effectively address such challenges. However, the use of machine learning mechanisms in cyber security brings challenges such as data reliability, model readability, and aggressive attacks. Ensuring the integrity and reliability of the training data is critical to avoid data bias and enable the development of reliable ML models. Furthermore, making sense of the findings made by highly nested neural networks is a challenging task, leading to debates in the area of transparency and accountability.

In this study [9], the authors consider the potential of hybrid approaches that combine several models to increase both the reliability and effectiveness of phishing detection. The researchers highlight the limitations of existing hybrid models that focus primarily on efficiency while ignoring broader applicability. To address these gaps, the authors present a new framework explicitly designed for real-world applications that lays the foundation for practical and robust phishing

detection architectures. The authors performed a proof-of-concept to evaluate its effectiveness, reliability, and detection speed. The authors also present an innovative methodology for simulating bypass attacks on basic models with one analysis. These experiments demonstrate that the proposed hybrid framework outperforms individual models, exhibiting higher efficiency, resistance to circumvention attempts, and real-time detection capabilities. The proof-of-concept method achieves an accuracy of 97.44%, thus outperforming the current state-of-the-art approach while requiring less computational time. The results provide insight into the multifaceted factors behind hybrid models beyond simple performance and highlight the importance of holistic applicability of hybrid approaches to address the critical need for robust phishing protection.

In [20] system aims to enhance user security by detecting phishing websites, ensuring safe browsing and transactions while protecting sensitive information was proposed. It provides users with a browser extension that helps identify whether a website is legitimate or not. The system combines heuristic features, visual features, and various approaches to feed machine learning algorithms, ensuring effective detection. A key challenge is adapting to new phishing tactics, which requires algorithms that continually learn and evolve. To achieve high accuracy, the system uses online learning algorithms and multiple approaches to improve precision. However, the system may occasionally produce minor false positives and false negatives, which can be minimized by incorporating more advanced features for the machine learning model, leading to better accuracy.

Identification and labeling of fake news is a difficult problem due to the huge amount of heterogeneous content. Essentially, the functions of machine learning (ML) and natural language processing (NLP) are to improve, accelerate and automate the analytical process. In this paper [21], a combination of ML and NLP is implemented to classify fake news based on an open, large, and labeled corpus on Twitter. In this case, the authors compare several state-of-the-art machine learning and neural network models based on content-only features. In order to improve the classification performance, inverse document frequency functions (TF-IDF) were applied before the training process in ML training, while word embedding was used in neural network training. Due to the application of ML and NLP methods, all traditional models have an accuracy of more than 85%. All neural network models have over 90% accuracy. In their experiments, the authors found that neural network models outperformed traditional ML models by an average of about 6% accuracy, with all neural network models achieving up to 90% accuracy.

This research [22] presents a new method for detecting phishing attacks on websites, avoiding the problems associated with the shortcomings of knowledge-based representation and binary solution. The proposed detection method was performed using Fuzzy Rule Interpolation (FRI). FRI reasoning methods have added the advantage of increasing the robustness of fuzzy systems and effectively reducing system complexity. These benefits help the intrusion detection system (IDS) generate more realistic and comprehensive alerts in the event of phishing attacks. The proposed method was applied to a dataset of an open-source phishing website. The results show that the proposed detection method achieved a detection rate of 97.58% and effectively reduced the number of false alarms. Additionally, it effectively blurs the line between normal and phishing traffic due to its fuzzy nature. It has the ability to generate the necessary security alert in case of deficiencies in the knowledge-based representation. In addition, the results obtained using the proposed detection method were compared with other literature results. The results showed that the accuracy rate of this work is competitive with other methods. In addition, the proposed detection method can generate the necessary anti-phishing alerts even if one of the sparse anti-phishing rules does not cover some input parameters (observations).

Article [23] presents an in-depth exploration of the current landscape of social engineering attacks, detailing their classifications and outlining a range of mitigation strategies organizations can implement to protect their most valuable assets against these persistent and rapidly evolving threats.

In study [24], the authors proposed a new scheme called Routing Protocol for Energy-Efficient Networks (RPEEN) for clone attack detection in an IoT-based intelligent healthcare application. The main advantage of this scheme is the improvement of energy efficiency, since energy efficiency is the most important constraint in WSN systems. The performance of the proposed scheme is

highlighted using parameters such as time delay, residual energy, throughput, energy efficiency, and error rate. In addition, to show the effectiveness of the proposed algorithm, this algorithm is compared with the existing hybrid multilevel clustering (HMLC) algorithm. It is found that the proposed RPEEN scheme achieves a time delay of 0.63 and 0.6ms with 0 dead nodes and by avoiding the clone attack, respectively. In addition, the proposed scheme achieves the highest residual energy of 49.5 J for 2500 shots. In addition, the proposed algorithm achieves the highest throughput of 99.2% for 50 nodes. The emergence of large language models (LLMs), including ChatGPT, has had a significant impact on a wide range of fields. Although LLMs have been widely investigated for tasks such as code generation and text synthesis, their application to detect malicious web content, particularly phishing sites, has been little studied. To counter the growing wave of cyberattacks due to misuse of LLM, it is important to automate detection using advanced LLM capabilities.

In paper [25], the authors propose a new system called ChatPhishDetector that uses LLM to detect phishing sites. This system involves using a web crawler to gather information from websites, generate hints for LLM based on the crawled data, and then derive detection results from the responses generated by LLM. The system enables the detection of multilingual phishing sites with high accuracy, identifying fake brands and social engineering techniques in the context of the entire website without the need to train machine learning models. To evaluate the performance of the system, the authors performed experiments on their own dataset and compared it with the baseline systems and several LLMs. Experimental results using GPT-4V have demonstrated outstanding performance with a precision of 98.7% and a recall of 99.6%, outperforming the detection results of other LLMs and existing systems. These findings highlight the potential of LLM to protect users from online fraud and have important implications for strengthening cybersecurity measures.

Nevertheless, there are disadvantages for each of the provided detection approaches: complexity in deployment, when the model relies on a neural network and custom algorithms, making it potentially harder for users to maintain; high execution time; limited scalability; overfitting risk; huge feature dependence; high computational cost; resource intensive; high complexity; over-reliance on datasets; limited applicability; transparency challenges; false alarms. Such situation requires the development of new approaches and new model, that take into account all aspects of social engineering attack functioning.

3. Development of the social engineering attacks models

Let us define the set S as the social engineering attacks set, $\Xi=\{\alpha,\beta,\gamma,\delta,\varepsilon,\epsilon,\zeta,\eta,\theta,\vartheta,\iota,\kappa,\lambda,\mu,\nu\}$, where α – the vishing attack; β – phishing attack; γ – profile cloning; δ – grooming; ε – dumpster diving attacks; ϵ – tailgating; ζ – file masquerade; η – baiting; θ – scareware or pop-up windows; ϑ – water-holing; ι – trojan mail; κ – spear phishing; λ – spam mail; μ – interesting software; ν – hoaxing.

3.1. Trojan mail attack model

In order to develop Trojan mail attack model, let us focus on the key components of the attack:

1. Email crafting. Hackers design emails that appear to be from trusted sources, such as a known colleague, a reputable company, or even government entities. The content of the email typically contains a sense of urgency or relevance to prompt the recipient to interact with the links or attachments. For example, the email might reference an overdue invoice, a shipment confirmation, or a required update.
2. Spoofing and deceptive tactics. Hackers may use email spoofing to mask their true identity, making the email appear as if it's coming from a legitimate sender. They often replicate the visual style and tone of official communication to reduce suspicion, using logos, familiar phrases, or signatures.
3. Malicious link or attachment. The email includes a malicious link or attachment, often in the form of a document (e.g., PDF, Word, Excel) or a ZIP file. Clicking on the link directs the user

to a compromised site or triggers a malware download. Opening the attachment similarly executes the malware, installing it on the user's system.

4. Trojan execution. Once activated, the malware (often a trojan) installs itself silently on the user's device. The trojan may open a backdoor for remote access, allowing hackers to control the system, capture keystrokes or take screenshots to steal login credentials and sensitive information, or spread laterally across the network, infecting other systems.
5. Unauthorized access and data theft. After the trojan is successfully installed, hackers can gain unauthorized access to the infected system. The hackers may use this access to steal confidential information such as passwords, financial details, or personal data, monitor network activity and gather intelligence for further attacks, or encrypt the system or files for ransomware attacks.
6. Continued Exploitation. The trojan remains hidden and continues to operate without the user's knowledge, enabling ongoing surveillance or exploitation. Hackers can use the compromised system to launch additional attacks, either within the organization or against external targets.

Trojan mail attack model has to include the set of countermeasures. Thus, to protect against Trojan mail attacks, individuals and organizations should implement several defensive strategies:

1. Email Filtering and Security. We are to use advanced email filtering solutions to block suspicious emails or detect common signs of phishing and malware delivery, and implement email authentication protocols, such as SPF, DKIM, and DMARC, to prevent email spoofing.
2. User Awareness and Training. We are to educate users to recognize phishing emails, especially those containing suspicious attachments or unexpected requests for action, and train users to avoid clicking on unfamiliar links or downloading attachments from unverified sources.
3. Antivirus and malware protection. We are to ensure that antivirus and anti-malware solutions are updated regularly to detect and block trojans and other types of malware, and enable real-time scanning of email attachments and downloads.
4. Network Segmentation and Access Controls. We are to limit the lateral movement of malware by segmenting networks and implementing access controls. This helps to contain an infection if it does occur, and employ least privilege policies, ensuring users have access only to the resources they need.
5. Backup and Recovery. We are to regularly back up critical data and maintain a recovery plan in case of infection. This can help mitigate the damage caused by ransomware or data theft following a trojan mail attack.

Let us present the model of the trojan mail attack as the tuple:

$$M_\delta = \langle A_\delta, E_\delta, U_\delta, M_\delta, S_\delta, D_\delta \rangle, \quad (1)$$

where $A_\delta = \{a_{\delta 1}, a_{\delta 2}, \dots, a_{\delta N_{A_\delta}}\}$ is the set that represents the hackers responsible for crafting and distributing trojan mail, N_{A_δ} - the number of individuals conducting the trojan mail attacks;

$E_\delta = \{e_{\delta 1}, e_{\delta 2}, \dots, e_{\delta N_{E_\delta}}\}$ is the set that represents the emails that are sent to potential victims, which contain malicious links or attachments, N_{E_δ} - the number of emails;

$U_\delta = \left\{u_{\delta 1}, u_{\delta 2}, \dots, u_{\delta N_{U_\delta}}\right\}$ is the set that represents the users who receive and interact with the malicious emails, N_{U_δ} - number of users;

$M_\delta = \left\{m_{\delta 1}, m_{\delta 2}, \dots, u_{\delta N_{M_\delta}}\right\}$ is the set that represents the trojan malware that is embedded in the links or attachments, N_{M_δ} – number of trojan malware;

$S_\delta = \left\{s_{\delta 1}, s_{\delta 2}, \dots, s_{\delta N_{S_\delta}}\right\}$ is the set that represents the systems that are compromised once the malware is activated, N_{S_δ} – number of compromised systems;

$D_\delta = \left\{d_{\delta 1}, d_{\delta 2}, \dots, d_{\delta N_{D_\delta}}\right\}$ is the set that represents the confidential data that hackers target for theft or unauthorized access, N_{D_δ}– number of confidential data.

Let us define the email crafting function $f_{\delta E^C}$ that describes how the attacker crafts emails and sends them to users, as:

$$f_{\delta E^C}: A_\delta \times U_\delta \to E_\delta,$$
$$f_{\delta E^C}\,(a_{\delta j}, u_{\delta r}) = e_{\delta p}.$$

where the attacker $a_{\delta j}$ sends an email $e_{\delta p}$ to the user $u_{\delta r}$.

Let us define the Malware Delivery Function $f_{\delta M^D}$ that describes how users interact with the email and trigger the malware, as:

$$f_{\delta M^D}: E_\delta \times U_\delta \to M_\delta,$$
$$f_{\delta M^D}(e_{\delta p}, u_{\delta r}) = m_{\delta q},$$

where the user $u_{\delta r}$ opens the email $e_{\delta p}$ and activates the malware $m_{\delta q}$.

Let us define the System Compromise Function $f_{\delta S^C}$ that describes how the malware infects the user's system, as:

$$f_{\delta S^C}: M_\delta \times U_\delta \to S_\delta,$$
$$f_{\delta S^C}(m_{\delta q}, u_{\delta r}) = s_{\delta x},$$

where the malware $m_{\delta q}$ compromises the user's system $s_{\delta x}$.

Let us define the Unauthorized Access Function $f_{\delta U^A}$ that describes how hackers gain unauthorized access to systems through the trojan, as:

$$f_{\delta U^A}: A_\delta \times S_\delta \to S_\delta,$$
$$f_{\delta U^A}(a_{\delta j}, s_{\delta x}) = S_{\delta\, x_\delta}^{\,a_\delta},$$

where the hacker $a_{\delta j}$ gains control of the system $s_{\delta x}$, creating $S_{\delta\, x_\delta}^{\,a_\delta}$ (the compromised system).

Let us define the data theft function $f_{\delta D^T}$ that describes how hackers steal data from the compromised systems, as:

$$f_{\delta D^T}: A_\delta \times S_\delta \to D_\delta,$$
$$f_{\delta D^T}(a_{\delta j}, s_{\delta x}) = d_{\delta t}.$$

where the hacker $a_{\delta j}$ steals data $d_{\delta t}$ from the compromised system $s_{\delta x}$.

The overall impact of the trojan mail attack can be measured by the number of infected systems, the amount of stolen data, and the extent of unauthorized access. Thus, impact function g_δ can be presented as:

$$g_\delta: A_\delta \times E_\delta \times U_\delta \times M_\delta \times S_\delta \times D_\delta \to \mathbb{R},$$
$$g_\delta\left(a_{\delta i}, e_{\delta p}, u_{\delta r}, m_{\delta q}, s_{\delta x}, d_{\delta t}\right) = I_{\varepsilon_T},$$

where I_{ε_T} represents the impact of the trojan mail attack, considering factors such as the number of compromised systems, the severity of the data theft or unauthorized access, the spread of the malware across users and systems.

Thus, for a specific victim $u_{\delta i}$ targeted by attacker $a_{\delta j}$:
$f_{\delta E^C}(a_{\delta j}, u_{\delta r}) = e_{\delta p}$ the hacker sends a malicious email to the user;
$f_{\delta M^D}(e_{\delta p}, u_{\delta r}) = m_{\delta q}$ the user opens the email, activating the malware;
$f_{\delta S^C}(m_{\delta q}, u_{\delta r}) = s_{\delta x}$ the malware compromises the user's system;
$f_{\delta U^A}(a_{\delta j}, s_{\delta x}) = S_{\delta x_\delta}^{a_\delta}$ the hacker gains unauthorized access to the system;
$f_{\delta D^T}(a_{\delta j}, s_{\delta x}) = d_{\delta t}$ the hacker steals data from the compromised system.

3.2. Spear phishing attack model

In order to develop spear phishing attack model, let us focus on the key components of the attack:

1. Information Gathering. Attackers begin by researching their target extensively. This may involve collecting data from social media profiles, professional networking sites (like LinkedIn), or publicly available information. The goal is to create a detailed profile of the victim, including their job role, interests, contacts, and recent activities.
2. Message Crafting. With the gathered information, attackers craft a convincing email or message that is highly personalized and relevant to the victim. The message often includes familiar references, such as the names of colleagues, recent projects, or organizations the victim is associated with. This familiarity is intended to lower the victim's defenses.
3. Deceptive Links or Attachments. The crafted message typically contains links to malicious websites or attachments with embedded malware. These links might mimic legitimate URLs or point to fake websites designed to harvest credentials.
4. Execution of the Attack. When the victim clicks the link or opens the attachment, they may be directed to a fake login page where they are prompted to enter their credentials, unknowingly providing them to the attacker. If the attack involves malware, it may be downloaded onto the victim's system, allowing the attacker to gain access to sensitive data or further infiltrate the network.
5. Account Compromise. Once the attacker obtains the victim's login credentials or malware is installed, they can access the victim's accounts, whether personal or organizational. This access may lead to unauthorized transactions, data theft, or further attacks against the victim's contacts.
6. Exploitation of Access. Attackers may use the compromised account to send additional spear phishing emails to the victim's contacts, thereby expanding the attack. They may also exploit the access to steal sensitive data, conduct fraud, or manipulate transactions.

Attack model has to include the set of countermeasures. Thus, to protect against attacks, individuals and organizations should implement several defensive strategies:

1. User Education and Awareness. Train users to recognize the signs of spear phishing, including suspicious emails, unexpected requests for sensitive information, and links to unfamiliar sites. Encourage users to verify the authenticity of messages before clicking links or providing information.
2. Email Security Measures. Implement email filtering solutions to detect and block suspicious messages. Use email authentication methods (e.g., SPF, DKIM, DMARC) to reduce the likelihood of spoofed emails.
3. Multi-Factor Authentication (MFA). Enable MFA on sensitive accounts to add an additional layer of security. This makes it more difficult for attackers to gain access even if they have stolen login credentials.
4. Regular Monitoring and Incident Response. Monitor accounts and systems for unusual activity that may indicate a successful attack. Establish an incident response plan to quickly address any security breaches.

5. Limit Information Sharing. Be cautious about the amount of personal and professional information shared on social media and other online platforms. Review privacy settings to control who can see information.

Let us present the model of the spear phishing attack as the tuple:

$$M_\varepsilon = \langle A_\varepsilon, T_\varepsilon, I_\varepsilon, M_\varepsilon, R_\varepsilon, S_\varepsilon, D_\varepsilon \rangle, \quad (2)$$

where $A_\varepsilon = \{a_{\varepsilon 1}, a_{\varepsilon 2}, \dots, a_{\varepsilon N_{A_\varepsilon}}\}$ is the set that represents the attackers involved in spear phishing, N_{A_ε} – number of attackers;

$T_\varepsilon = \{t_{\varepsilon 1}, t_{\varepsilon 2}, \dots, t_{\varepsilon N_{T_\varepsilon}}\}$ is the set that represents the specific individuals or organizations targeted by the attack, N_{T_ε} – number of individuals;

$I_\varepsilon = \{i_{\varepsilon 1}, i_{\varepsilon 2}, \dots, i_{\varepsilon N_{I_\varepsilon}}\}$ is the set that represents the collected information about the targets, such as personal details and professional affiliations, N_{I_ε} – number of collected information;

$M_\varepsilon = \{m_{\varepsilon 1}, m_{\varepsilon 2}, \dots, m_{\varepsilon N_{M_\varepsilon}}\}$ is the set that represents the crafted messages sent to targets, which may contain malicious links or attachments, N_{M_ε} – number of crafted messages;

$R_\varepsilon = \left\{r_{\varepsilon 1}, r_{\varepsilon 2}, \dots, r_{\varepsilon N_{R_\varepsilon}}\right\}$ is the set that represents the malicious software that may be delivered through the attack, N_{R_ε} – number of malicious software;

$S_\varepsilon = \left\{s_{\varepsilon 1}, s_{\varepsilon 2}, \dots, s_{\varepsilon N_{S_\varepsilon}}\right\}$ is the set that represents the systems compromised as a result of the attack, N_{S_ε} – number of compromised systems;

$D_\varepsilon = \left\{d_{\varepsilon 1}, d_{\varepsilon 2}, \dots, d_{\varepsilon N_{D_\varepsilon}}\right\}$ is the set that represents the confidential information targeted for theft, N_{D_ε} – number of confidential information;

Let us define the information collection function $f_{\varepsilon I^C}$ that describes how attackers gather information about their targets, as:

$$f_{\varepsilon I^C}: A_\varepsilon \times T_\varepsilon \to I_\varepsilon,$$
$$f_{\varepsilon I^C}\ (a_{\varepsilon j}, t_{\varepsilon r}) = i_{\varepsilon p}.$$

The attacker $a_{\varepsilon j}$ collects information $i_{\varepsilon p}$ about the target $t_{\varepsilon r}$.

Let us define the message crafting function $f_{\varepsilon M^C}$ that describes how attackers create personalized messages based on the collected information, as:

$$f_{\varepsilon M^C}: I_\varepsilon \times T_\varepsilon \to M_\varepsilon,$$
$$f_{\varepsilon M^C}\ (i_{\varepsilon p}, t_{\varepsilon r}) = \ m_{\varepsilon q},$$

where the attacker $a_{\varepsilon j}$ crafts a message $m_{\varepsilon q}$ for the target $t_{\varepsilon r}$.

Let us define the Message Sending Function $f_{\varepsilon M^S}$ that describes how the crafted message is sent to the target, as:

$$f_{\varepsilon M^S}: M_\varepsilon \times T_\varepsilon \to T_\varepsilon,$$
$$f_{\varepsilon M^S}\ (m_{\varepsilon q}, t_{\varepsilon r}) = t_{r_\varepsilon}^{m_\varepsilon},$$

where the message $m_{\varepsilon q}$ is sent to the target $t_{\varepsilon r}$.

Let us define the Malware Delivery Function $f_{\varepsilon M^D}$ that describes how the target interacts with the message, potentially activating malware, as:

$$f_{\varepsilon M^D}: M_\varepsilon \times T_\varepsilon \to R_\varepsilon$$
$$f_{\varepsilon M^D}\ (m_{\varepsilon q}, t_{\varepsilon r}) = \ r_{\varepsilon s}$$

where the target $t_{\varepsilon r}$ activates the malware $r_{\varepsilon s}$ by interacting with the message.

Let us define the System Compromise Function $f_{\varepsilon S^C}$ that describes how the malware compromises the target's system, as:

$$f_{\varepsilon S^C}: R_\varepsilon \times T_\varepsilon \to S_\varepsilon,$$
$$f_{\varepsilon S^C}\ (r_{\varepsilon s}, t_{\varepsilon r}) = \ s_{\varepsilon x},$$

where the malware $r_{\varepsilon s}$ compromises the system $s_{\varepsilon x}$ of the target $t_{\varepsilon r}$.

Let us define the Data Theft Function $f_{\varepsilon D^T}$ that describes how attackers gain access to confidential data once the system is compromised, as:

$$f_{\varepsilon D^T}: A_\varepsilon \times S_\varepsilon \to D_\varepsilon,$$
$$f_{\varepsilon D^T}\ (a_{\varepsilon j}, s_{\varepsilon x}) = \ d_{\varepsilon t},$$

where the attacker $a_{\varepsilon j}$steals data $d_{\varepsilon t}$ from the compromised system $s_{\varepsilon x}$.

The overall impact of a spear phishing attack can be quantified based on the number of systems compromised, the volume of data stolen, and the extent of unauthorized access achieved.

Thus, let us present the impact function $g_\varepsilon$as:

$$g_\varepsilon: A_\varepsilon \times T_\varepsilon \times M_\varepsilon \times R_\varepsilon \times S_\varepsilon \times D_\varepsilon \to \mathbb{R},$$
$$g_\varepsilon\left(a_{\varepsilon i}, t_{\varepsilon r}, m_{\varepsilon q}, r_{\varepsilon s}, s_{\varepsilon x}, d_{\varepsilon t}\right) = I_{\varepsilon_S},$$

where I_{ε_S} represents the impact of the spear phishing attack, considering factors such as -the number of compromised systems, the value of stolen data or unauthorized access obtained, the potential damage to the victim's reputation and finances.

Thus, for a specific victim $t_{\varepsilon i}$ targeted by attacker $a_{\varepsilon j}$:

$f_{\varepsilon I^C}(a_{\varepsilon j}, t_{\varepsilon r}) = i_{\varepsilon p}$ the attacker gathers information about the target;

$f_{\varepsilon M^C}(i_{\varepsilon p}, t_{\varepsilon r}) = m_{\varepsilon q}$ the attacker crafts a personalized message for the target;

$f_{\varepsilon M^S}(m_{\varepsilon q}, t_{\varepsilon r}) = t_{\varepsilon_{r_\varepsilon}}^{m_\varepsilon}$ the message is sent to the target;

$f_{\varepsilon M^D}(m_{\varepsilon q}, t_{\varepsilon r}) = r_{\varepsilon s}$ the target activates the malware from the message;

$f_{\varepsilon S^C}(r_{\varepsilon s}, t_{\varepsilon r}) = s_{\varepsilon x}$ the malware compromises the target's system.

$f_{\varepsilon D^T}(a_{\varepsilon j}, s_{\varepsilon x}) = d_{\varepsilon t}$ the attacker steals data from the compromised system.

3.3. Spam mail attack model

In order to develop spam mail attack model, let us focus on the key components of the attack:

1. Spam mail attack model email list acquisition. Attackers often obtain lists of email addresses through various means, including data breaches, purchasing lists from underground markets, or using web scrapers to collect publicly available addresses. This list serves as the target pool for the spam campaign.
2. Message crafting. Spam emails can vary widely in content, from promotional offers and phishing attempts to scams and malicious links. Attackers may create enticing subject lines to increase open rates, often using urgency or enticing offers (e.g., "Limited Time Offer!" or "You've Won a Prize!") to lure victims.
3. Distribution methods. Emails can be sent using various methods, including botnets, bulk email services, or compromised accounts. Botnets, which are networks of infected computers, are often employed to distribute spam more efficiently and evade detection.
4. Call to action. The emails typically include a call to action, encouraging recipients to click on a link, enter personal information, or download attachments. Links may lead to phishing sites designed to capture sensitive information or malicious downloads that infect the user's system with malware.
5. Malware delivery. Some spam emails contain attachments that, when opened, install malware on the recipient's device. This can include ransomware, spyware, or adware, leading to further exploitation of the victim's data. Infected systems may be used for further spam distribution, creating a cycle of infection.

6. Tracking and analytics. Attackers often implement tracking mechanisms to measure the effectiveness of their campaigns, such as monitoring open rates, click-through rates, and conversions. This information helps refine future spam campaigns and target more effectively.

Consequences of spam mail attacks are to be added to the modes as well:

1. Information Theft. Users who fall for phishing scams may inadvertently provide personal data, leading to identity theft or unauthorized financial transactions.
2. Malware Infection. Clicking on links or downloading attachments can lead to malware infections, compromising the victim's system and possibly leading to network breaches in organizational settings.
3. Resource Drain. The sheer volume of spam can overwhelm email systems, causing legitimate emails to be lost or delayed. This can lead to reduced productivity for individuals and organizations alike.
4. Reputation Damage. If a user's account is compromised due to spam, it may be used to send further spam, damaging the sender's reputation and leading to blacklisting.

Spam mail attacks model has to include the set of countermeasures. To mitigate the risks associated with spam mail, individuals and organizations can adopt the following strategies:

1. Email filtering. Implement spam filters and email security solutions to block unwanted emails before they reach users' inboxes.
2. User education. Educate users about recognizing spam and phishing attempts, including common signs like poor grammar, generic greetings, and suspicious links.
3. Avoiding unsubscribe links. Encourage users not to click unsubscribe links in spam emails, as they may confirm to the sender that the email address is active, leading to more spam.
4. Use of strong security practices. Utilize strong passwords and enable two-factor authentication to protect email accounts from being compromised.
5. Regular software updates. Keep operating systems, antivirus software, and applications updated to protect against vulnerabilities that could be exploited by malware delivered via spam.

Let us present the model of the spear spam mail attack as the tuple:

$$M_\epsilon = \langle A_\epsilon, T_\epsilon, I_\epsilon, M_\epsilon, R_\epsilon, S_\epsilon, D_\epsilon \rangle, \quad (3)$$

where $A_\epsilon = \{a_{\epsilon 1}, a_{\epsilon 2}, \dots, a_{\epsilon N_{A_\epsilon}}\}$ is the set that represents the individuals or groups sending spam emails, N_{A_ϵ} – number of individuals;

$R_\epsilon = \{r_{\epsilon 1}, r_{\epsilon 2}, \dots, r_{\epsilon N_{R_\epsilon}}\}$ is the set that represents the potential victims who receive spam emails, – number of potential victims;

$E_\epsilon = \{e_{\epsilon 1}, e_{\epsilon 2}, \dots, e_{\epsilon N_{E_\epsilon}}\}$ is the set that represents the spam emails sent out by attackers, N_{E_ϵ} – number of spam emails;

$M_\epsilon = \{m_{\epsilon 1}, m_{\epsilon 2}, \dots, m_{\epsilon N_{E_\epsilon}}\}$ is the set that represents the malicious software that may be included in the spam emails, N_{E_ϵ} – number of malicious software;

$T_\epsilon = \{t_{\epsilon 1}, t_{\epsilon 2}, \dots, t_{\epsilon N_{T_\epsilon}}\}$ is the set that represents the tracking data collected by attackers to measure the success of their spam campaigns, N_{T_ϵ} – number of data collected by attackers;

$C_\epsilon = \{c_{\epsilon 1}, c_{\epsilon 2}, \dots, c_{\epsilon N_{C_\epsilon}}\}$ is the set that represents the potential consequences for the recipients of spam emails, N_{C_ϵ} - number of potential consequences.

Let us define the email distribution function $f_{\epsilon E^D}$ that describes how attackers send spam emails to recipients, as:

$$f_{\epsilon E^D}: A_\epsilon \times E_\epsilon \to R_\epsilon,$$

$$f_{\epsilon E^D}\ (a_{\epsilon j}, e_{\epsilon k}) = r_{\epsilon p},$$

where the attacker $a_{\epsilon j}$sends email $e_{\epsilon k}$ to recipient $r_{\epsilon p}$.

Let us define the click function $f_\epsilon C$that describes how recipients may interact with the spam emails, as:

$$f_\epsilon C: R_\epsilon \times E_\epsilon \to M_\epsilon,$$
$$f_\epsilon C\ (r_{\epsilon p}, e_{\epsilon k}) = m_{\epsilon q},$$

where the recipient $r_{\epsilon p}$ clicks on a link or downloads malware $m_{\epsilon q}$ from email $e_{\epsilon k}$.

Let us define the infection function $f_\epsilon I$ that describes the process of a recipient's system being infected by malware, as:

$$f_\epsilon I: M_\epsilon \times R_\epsilon \to C_\epsilon,,$$
$$f_\epsilon I\ (m_{\epsilon q}, r_{\epsilon p}) = c_{\epsilon t},$$

where the malware $m_{\epsilon q}$ infects the recipient's system, resulting in consequence $c_{\epsilon t}$.

Let us define the Tracking Function $f_\epsilon T$ that describes how attackers track the success of their spam campaign, as:

$$f_\epsilon T: A_\epsilon \times R_\epsilon \to T_\epsilon,$$
$$f_\epsilon T\ (a_{\epsilon j}, r_{\epsilon p}) = t_{\epsilon s},$$

where the attacker $a_{\epsilon j}$ collects tracking data $t_{\epsilon s}$ based on recipient's interaction $r_{\epsilon p}$ with spam.

The overall impact of a spam mail attack can be quantified based on the number of recipients affected, the amount of malware delivered, and the potential damage caused. Thus, the impact function g_ϵ can be presented as:

$$g_\epsilon: A_\epsilon \times R_\epsilon \times E_\epsilon \times M_\epsilon \to \mathbb{R},$$
$$g_\epsilon\left(a_{\epsilon j}, r_{\epsilon p}, e_{\epsilon k}, m_{\epsilon q}\right) = I_{\varepsilon_S},$$

where I_{ε_S} represents the impact of the spam mail attack, considering factors such as the number of systems infected, the volume of personal data compromised, the cost associated with the attack, including system recovery and reputation damage.

Thus, for a specific victim $u_{\varepsilon i}$ targeted by attacker $a_{\varepsilon j}$:

$f_{\varepsilon E^D}(a_{\varepsilon j}, e_{\varepsilon k}) = r_{\varepsilon p}$ the attacker sends spam email $e_{\varepsilon k}$ to recipient $r_{\varepsilon p}$.

$f_C(r_{\varepsilon p}, e_{\varepsilon k}) = m_{\varepsilon q}$ the recipient clicks on a link or downloads malware $m_{\varepsilon q}$ from email $e_{\varepsilon k}$.

$f_I(m_{\varepsilon q}, r_{\varepsilon p}) = c_{\varepsilon t}$: the malware $m_{\varepsilon q}$ infects the recipient's system, leading to consequence $c_{\varepsilon t}$.

$f_T(a_{\varepsilon j}, r_{\varepsilon p}) = t_{\varepsilon s}$ the attacker collects tracking data $t_{\varepsilon s}$ based on recipient $r_{\varepsilon p}$ interaction with the spam.

4. Experiments

4.1. Title information

To assess the effectiveness of the developed models the BotGRABBER framework was employed. It is a security tool designed to enhance network resilience against cyberattacks. The system leverages machine learning. One of the key aspects of BotGRABBER is its ability to perform self-adaptive security actions. Additionally, framework is designed to integrate seamlessly with various machine learning algorithms to refine its detection capabilities, ensuring efficient performance in complex network environments [8].

To conduct an experiment for detecting social engineering attacks (such as spam mail, spear phishing, and trojan mail attacks), the setup involves specific hardware, network configurations, and security settings. A controlled email server with logging enabled, such as Microsoft Exchange, to

capture emails and analyze metadata, as well as the simulated mailboxes to receive test spam, phishing, and trojan emails were used.

For detection automation, libraries such as scikit-learn or TensorFlow to process email features like sender, subject line, body text, and links for patterns indicative of social engineering were employed.

An isolated network to avoid real-world impact if malware is executed during testing. Use virtual machines or a controlled subnet within a Virtual Private Cloud (VPC) was set up. Snort tool as IDS/IPS to monitor network traffic for anomalies that align with phishing and trojan attacks was used. For trojan attack detection, a sandbox environment (Cuckoo Sandbox) that safely opens and monitors email attachments to identify potentially malicious behavior without compromising real systems was incorporated.

A labeled dataset with examples of spam, spear-phishing, and trojan mails was created. To do this an open-source datasets [26, 27, 28] for training and testing were used. Detailed logging on email servers and network devices to capture metadata (e.g., headers, sender IPs, attachment details) were enable.

Traffic logs were stored in a centralized logging system for analysis. Features such as the frequency of certain keywords, unusual sender addresses, mismatched domain names, attachment types, and user interaction patterns were extracted. Machine learning models (random forest, decision tree, K-nearest neighbor, and XGBoost) on both legitimate and malicious email datasets to classify emails based on phishing indicators were trained [29].

Results for three types of attacks are presented in Figures 1-3. The experiment tested algorithms including random forest, decision tree, K-nearest neighbor, and XGBoost, analyzing host network data that can signal a potential social engineering attack. The empirical findings showed a high detection accuracy of about 99%, alongside a false positive rate near 6%. Thus, the implementation of the developed models for social engineering attacks detection has demonstrated high detection potential.

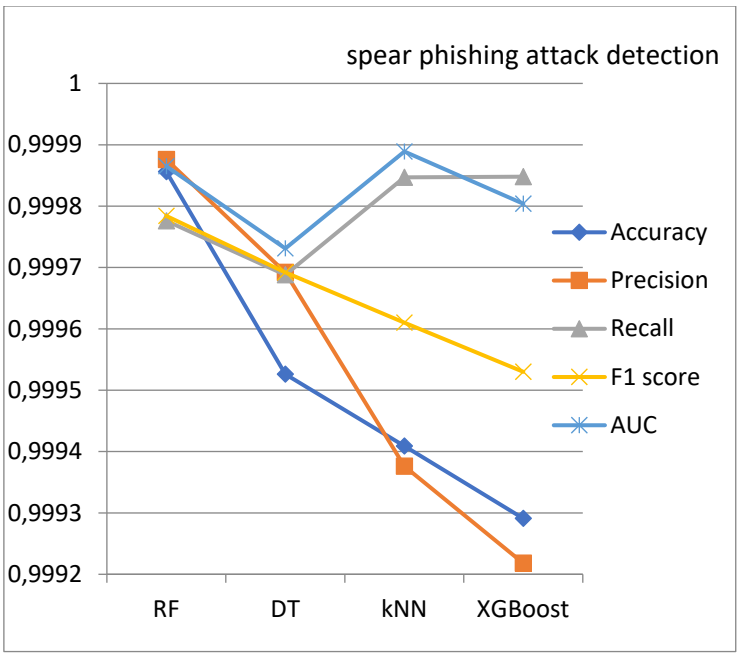

Figure 1: Spear phishing attack detection results.

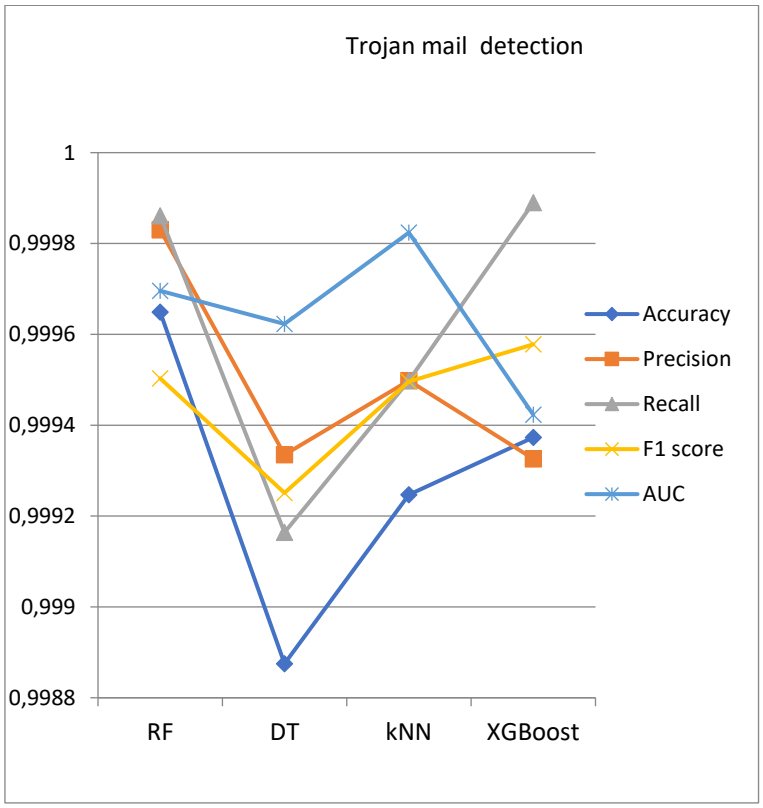

Figure 2: Trojan mail attack detection results.

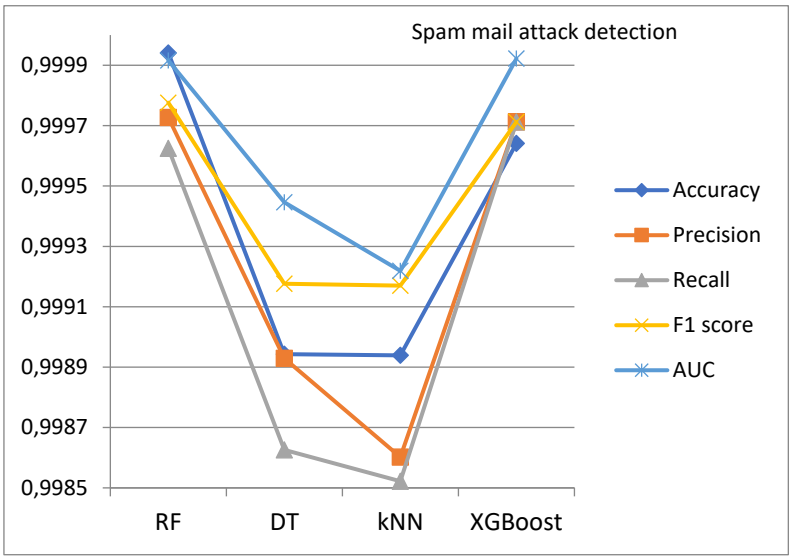

Figure 3: Spam mail attack detection results.

5. Conclusions

The research developed specialized models for detecting social engineering attacks, focusing on spam mail, spear phishing, and trojan mail. Each model captures unique characteristics of these attacks through a series of machine learning-based detection processes. Developed models analyze features such as email metadata, user interaction patterns, attachment behaviors, and network anomalies to distinguish malicious activity from legitimate communication.

The trojan mail model emphasizes the identification of embedded malware within email attachments, employing sandbox environments for controlled testing and analysis of attachment behaviors. The spear phishing model, in contrast, focuses on personalization tactics, using sender

recognition and link analysis to detect contextually suspicious patterns. The spam mail model prioritizes content filtering and call-to-action tracking to differentiate legitimate communication from mass-distributed spam.

The empirical findings validate the models' robustness, achieving approximately 99% accuracy in detection while maintaining a 6% false positive rate. This high detection performance illustrates the potential of our models to support a proactive defense framework against evolving social engineering threats. By leveraging specific feature sets and adaptive machine learning algorithms, these models can be effectively implemented in real-world scenarios to protect networks and systems from a wide range of social engineering attacks.

Future work may explore hybrid models, advanced behavioral analytics, and real-time detection capabilities to further enhance resilience against increasingly sophisticated attacks. The future development of these models may explore the combining these individual models to create a more unified system capable of detecting multiple attack types simultaneously, improving the adaptability of the detection framework; integrating behavioral profiling to understand normal user behavior and identify deviations that may signal an attack.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: grammar and spelling check; DeepL Translate in order to: some phrases translation into English. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Huseynov, B. Ozdenizci Kose. Using machine learning algorithms to predict individuals' tendency to be victim of social engineering attacks. Information Development, 2024, 40.2: 298-318.

[2] V. Kolluri. Revolutionary research on the ai sentry: an approach to overcome social engineering attacks using machine intelligence. International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.

[3] T. Rathod et al. A comprehensive survey on social engineering attacks, countermeasures, case study, and research challenges. Information Processing & Management, 2025, 62.1: 103928.

[4] A. Sharma. Natural Language Processing for Cybersecurity: Detecting and Mitigating Social Engineering Attacks. International Meridian Journal, 2024, 6.6.

[5] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, A. Nicheporuk. A Technique for detection of bots which are using polymorphic code. Communications in Computer and Information Science, 2014, vol. 431. PP.265-276.

[6] O.Savenko, S. Lysenko, A. Kryschuk. Multi-agent based approach of botnet detection in computer systems. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2012. CCIS, vol. 291, pp. 171–180. Springer, Heidelberg (2012). Doi:10.1007/978-3-642-31217-5_19.

[7] S. Lysenko, O. Pomorova, O. Savenko, A. Kryshchuk, K. Bobrovnikova. DNS-based Anti-evasion Technique for Botnets Detection. Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw (Poland), September 24–26, 2015. Warsaw, 2015. Pp. 453–458.

[8] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova Anti-evasion Technique for the Botnets Detection Based on the Passive DNS Monitoring and Active DNS Probing. Communications in Computer and Information Science. (2016) 608. 83-95.

[9] A. Naz, M.Sarwar, M. Kaleem, M. A. Mushtaq, S. Rashid, (2024). A comprehensive survey on social engineering-based attacks on social networks.

[10] S. Gupta et al. A Comprehensive Analysis of Social Engineering Attacks: From Phishing to Prevention-Tools, Techniques and Strategies. In: 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI). IEEE, 2024. p. 1-8.
[11] N. Akyeşilmen, A. Alhosban. Non-Technical Cyber-Attacks and International Cybersecurity: The Case of Social Engineering. Gaziantep University Journal of Social Sciences, (2024) 23.1 342-360.
[12] Cheng-Ying Yang, Chun-Yi Shih, Chou-Chen Yang, Min-Shiang Hwang, Freeze-Phish: An ANN Based Phishing Detection System, in International Journal of Network Security, 2023/09 893-898. doi:10.6633/IJNS.202309_25(5).19.
[13] A. Odeh, I. Keshta, E. Abdelfattah, PhiBoost- A novel phishing detection model Using Adaptive Boosting approach, Jordanian Journal of Computers and Information Technology (2021). doi: 10.5455/jjcit.71-1600061738.
[14] A.K.S. Sekar, S.S. Kumar, S. Sampath, U.T. Kumar, V. Vignesh, Phishing website clone detection using machine learning rules with cryptography technique, International Journal of Gender, Science and Technology, 13 (2024).
[15] I. Tosin, C. Kiekintveld, Aritran Piplai, Deep Learning-Based Speech and Vision Synthesis to Improve Phishing Attack Detection through a Multi-layer Adaptive Framework, Computer Science (2024). doi: 10.48550/arXiv.2402.17249.
[16] Abdulla Al-Subaiey, Mohammed Al-Thani, Naser Abdullah Alam, Kaniz Fatema Antora, Amith Khandakar, Novel interpretable and robust web-based AI platform for phishing email detection, in Computers and Electrical Engineering, 2024, Volume 120, Part A, doi:10.1016/j.compeleceng.2024.109625.
[17] N. W. Peace, A framework for securing email entrances and mitigating phishing impersonation attacks, Computer Science (2023). doi: 10.5121/ijnsa.2023.15602.
[18] V. Kolluri. Revolutionary research on the ai sentry: an approach to overcome social engineering attacks using machine intelligence. International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.
[19] R.J. van Geest , Cascavilla G., Hulstijn J., Zannone N., "The applicability of a hybrid framework for automated phishing detection", Computers & Security, 2024, doi:1016/j.cose.2024.103736.
[20] S. R. Janani, et al. Detection of Phishing Page Using Machine Learning and Response HTML. In: International Conference on Communications and Cyber Physical Engineering 2018. Singapore: Springer Nature Singapore, 2024. p. 499-508.
[21] C.M. Lai, M.H. Chen, E. Kristiani, V.K. Verma, C.T. Yang, Fake news classification based on content level features, Applied Sciences, (2022) 12(3) 1116.
[22] A. Maen, Al-S. Jamil, Cyber-Phishing Website Detection Using Fuzzy Rule Interpolation in Cryptography (2022) 6(2) 4. doi:10.3390/cryptography6020024.
[23] V. Bharath, , et al. Introduction to Social Engineering: The Human Element of Hacking. In: Social Engineering in Cybersecurity. CRC Press, 2024. p. 1-25.
[24] S.Vaishnavi, T. Sethukarasi. Detection and Avoidance of Clone Attack in IoT Based Smart Health Application, in Intelligent Automation & Soft Computing, 2022, 31(3):1919-1937
[25] K. Takashi, F. Naoki, N. Hiroki, C. Daiki, Detecting Phishing Sites Using ChatGPT in Computer Science, 2023. v1 , doi: 10.48550/arXiv.2306.05816
[26] M. Lansley, F. Mouton, S. Kapetanakis, N. Polatidis, SEADer++: social engineering attack detection in online environments using machine learning, J. Inf. Telecommun., 4(3) (2020) 346–362. doi: 10.1080/24751839.2020.1747001.
[27] A. A. Akinyelu, A. O. Adewumi, Classification of phishing email using rand om forest machine learning technique, J. Appl. Math., vol. 2014, 2014, doi: 10.1155/2014/425731.
[28] A.El Aassal, L. Moraes, S. Baki, A. Das, R. Verma, Anti -Phishing Pilot, in ACM IWSPA 2018 Evaluating Performance with New Metrics for Unbalanced Datasets, pp. 21–24, 2018.
[29] O. Revniuk, A. Postoliuk, Research on the application of adaptive risk assessment methods for web applications, Computer Systems and Information Technologies, 2024 (3), 34–43. https://doi.org/10.31891/csit-2024-3-5.