VESTA

Proactive Protection against Phishing-based Ransomware

D2.1 – Academic and Technology SoTA Report

Submission date of deliverable: November 30, 2024

Edited by: Hakan Kilinc (Orion, Türkiye), Eva Catarina Gomes Maia (ISEP, Portugal), Orhan Yildirim (Beam Teknoloji, Türkiye), Gabriela Sousa (VisionWare, Portugal), Özgü Özkan, Melike Çolak, Nesil Bor (Bites, Türkiye), Daniel Esteban Villamil Sierra (Panel, Spain)

Project start date	Jan 1, 2024
Project duration	36 months
Project coordinator	Murat Duran, duSoft Yazılım A.Ş.
Project number & call	21011 - ITEA 4
Project website	https://itea4.org/project/vesta.html
Contributing partners	See affiliations in “Edited by” above.
Version number	V1.7
Work package	WP2
Work package leader	Hakan Kilinc (Orion, Türkiye)
Dissemination level	Public
Description ***(max 5 lines)***	The goal of D2.1 (part of T2.1) is to conduct a comprehensive analysis of the current academic and technological aspects to identify scenario specific requirements and create innovative solutions.

Change Log

Version	Date	Authors	Description of changes
1.0	03.07.2024	Hakan Kilinc (Orion)	First template and content are created.
1.1	03.09.2024/ 17.09.2024	Hakan Kilinc (Orion)	Ransomware detection techniques and behaviour analysis chapter is updated
1.2	24.10.2024	Orhan Yıldırım (Beam)	Introduction and phishing detection techniques chapters are updated
1.3	19.11.2024/ 21.11.2024	Eva Catarina Gomes Maia (ISEP)	Phishing detection techniques chapter is updated
1.4	22.11.2024	Gabriela Sousa (VisionWare)	Advanced email security measures chapter is updated.
1.5	25.11.2024	Özgü Özkan, Melike Çolak, Nesil Bor (Bites)	Phishing detection techniques chapter is updated
1.6	26.11.2024	Daniel Esteban Villamil Sierra (Panel)	Network analysis chapter is updated.
1.7	30.11.2024	Hakan Kilinc (Orion)	Introduction and conclusion chapters are updated and finalized.

Executive Summary

Proactive protection against phishing-based ransomware requires a multi-layered approach that combines technology, training and best practices. Using these approaches, organizations can significantly reduce the risk of phishing-based ransomware attacks and improve their overall security posture. The VESTA project aims to develop solutions by focusing on these approaches.

Specifically, the VESTA project focuses on advanced email security measures for email phishing attacks, malware/ransomware detection and prevention, behaviour analysis, monitoring of suspicious activity, and network security analysis.

“D2.1 Academic and Technology SoTA Report” deliverable is the output of “T2.1 State of the Art and Technology”. This task defines a comprehensive analysis of the current academic and technological aspects to create innovative solutions.

This deliverable D2.1 reports on the technological and academic state of the art on the identified focus areas. The outputs of this deliverable will enable to initiate the other WPs.

Table of contents

Document Glossary

Acronym	Description
1D-CNN	One-Dimensional Convolutional Neural Networks
AI	Artificial Intelligence
APT	Advanced Persistent Threat
AV	Anti-Virus
Bi-LSTM	Bidirectional LSTM
Bi-GRU	Bidirectional GRU
CNN	Convolutional Neural Networks
DKIM	DomainKeys Identified Mail
DL	Deep Learning
DMARC	Domain-based Message Authentication, Reporting & Conformance
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DPI	Deep Packet Inspection
DT	Decision Tree
ELK	Elasticsearch, Logstash, and Kibana
ELM	Extreme Learning Machine
EOP	Exchange Online Protection
GRU	Gated Recurrent Units
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JS	JavaScript
IDS	Intrusion Detection Systems
IoT	Internet of Things
IPS	Intrusion Prevention Systems
LAN	Local Area Network
LLM	Large Language Model
LSTM	Long Short-Term Memory Networks

Acronym	Description
ML	Machine Learning
NLP	Natural Language Processing
OCR	Optical Character Recognition
OSINT	Open-Source Intelligence
RDP	Remote Desktop Protocol
RF	Random Forest
RNN	Recurrent Neural Networks
SIEM	Security Information and Event Management
SMB	Server Message Block
SOAR	Security Orchestration, Automation, and Response
SOC	Security Operations Center
SPF	Sender Policy Framework
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
UEBA	User and Entity Behavior Analytics
UI	User Interface
URL	Uniform Resource Locator
XGB	Extreme Gradient Boosting
YARA	Yet Another Ridiculous Acronym
ZTA	Zero Trust Architecture

1. Introduction

Phishing is a cyberattack method where attackers impersonate (pretend to be) legitimate organizations to deceive individuals into providing sensitive information, such as login credentials or financial data. When combined with ransomware, these attacks can cause severe financial and reputational damages.

Typical steps of a phishing-based ransomware attack:

1. **Initial Phishing Attack**: The attacker sends fraudulent (fake) emails containing malicious links or attachments. These emails may look legitimate and often include urgent language to prompt quick action.
2. **Malware Deployment**: If a recipient clicks on the link or opens the attachment, ransomware is downloaded onto their device. This malware typically encrypts files, making them inaccessible to the user.
3. **Ransom Demand**: Once the files are encrypted, the attacker demands a ransom (often in cryptocurrency) for the decryption key, leaving victims with the difficult choice of paying or losing their data.

Characteristics of phishing-based ransomware attacks:

- **Evolving Techniques**: Attackers constantly adapt their tactics, using social engineering (manipulation of individuals) techniques to increase the chances of success.
- **Targeted Campaigns**: Many phishing attacks are tailored to specific organizations or individuals (spear phishing), making them harder to detect.
- **Multiple Entry Points**: Ransomware can spread through various channels, including email attachments, malicious websites, or compromised software.

Importance of Proactive Protection Strategies:

Ransomware attacks increased by 74%, from 2,593 global attacks in 2022 to 4,506 attacks in 2023. There were 2,321 attacks in the first half of 2024 [1].

Given the increasing prevalence and frequency of phishing-based ransomware, it poses a significant threat to organizations and therefore proactive protection strategies such as email filtering, employee training and awareness, and incident response planning are required for organizations to mitigate risks, protect sensitive data and minimize potential impacts.

2. Phishing Detection and Prevention Techniques

Phishing attacks involve adversaries impersonating legitimate entities to send deceptive emails or texts containing malware or links to fraudulent websites. These attacks exploit human psychology through social engineering to trick victims into sharing confidential information, compromising security, or performing harmful actions. The goals include infecting devices with malware, stealing sensitive data (e.g., usernames and credit card details), seizing control of digital accounts, or prompting financial transactions [2]. Phishing poses a persistent threat, targeting individuals, businesses, and governments. These attacks have become more sophisticated, adapting to advancements in technology to create convincing and personalized campaigns. The consequences

can be severe, including financial losses, identity theft, unauthorized access to sensitive data, or systemic compromise [3].

Phishing leverages user interface flaws and human difficulty in verifying Uniform Resource Locator (URL)s and dynamic content, often serving as the initial access point to a victim's device or account, enabling further attacks. In this context, phishing detection techniques vary widely, from analysing the URL to analysing the message content.

In recent years, the application of Natural Language Processing (NLP) and Machine Learning (ML) has shown great promise in detecting phishing emails. By leveraging the power of NLP to analyse the linguistic patterns, syntax, and semantics of phishing emails, researchers have been able to identify potential threats that may be missed by traditional methods. Furthermore, ML algorithms can learn from large datasets of labelled examples, enabling them to develop robust models that can accurately distinguish between genuine and malicious emails.

Generative AI has also emerged as a powerful tool in phishing detection, leveraging advanced capabilities of large language models (LLMs) to analyze and identify potential threats. By scrutinizing email text, LLMs can detect signs of malicious intent, such as suspicious phrasing, abnormal sender behavior, or urgent language often associated with phishing scams. These models compare the analyzed content to vast databases of known phishing examples, enabling them to identify even subtle or novel attack patterns [4].

2.1. Email Filtering

Systems can implement various countermeasures to mitigate phishing attacks, with one prominent example being email filtering. This approach involves deploying multiple techniques to identify and block phishing or other malicious emails before they reach users' inboxes. By intercepting such emails, email filtering significantly reduces users' exposure to potential threats, minimizing the risk of interacting with harmful content and falling victim to phishing scams [5].

Several different approaches can be used for email filtering, but the one that has lately been getting a lot of attention and is showing promising results is ML. This approach seems appropriate to the problem of phishing detection since email filtering problem can be easily transformed into a typical classification task [6]. This happens due to the nature of phishing, which typically involves identifying specific patterns or characteristics in emails that distinguish them from legitimate messages. Phishing emails often use similar tactics, such as urgent language, requests for sensitive information, or spoofed sender addresses, making them identifiable by classification models trained to recognize these features [7].

M. Avukarasi and A. Antonidoss (2019) [8] used different ML algorithms to detect whether a mail is phishing or not. Then, results of these algorithms are compared. Even though the Neural Networks gave the best accuracy, each algorithm performed close and provided good results. They also state that classifiers (Decision Tree (DT), Support Vector Machine (SVM), ...) utilizing ML algorithms perform best for phishing detection. As another result of this study, it is found that different ML algorithms are good at predicting phishing content.

Other approaches used for email filtering include Natural Language Processing (NLP) [9]. NLP teaches computers the semantic meaning of natural-language text. Thus, an NLP system reads plain English (among other languages) and categorizes what it's seen in terms of conceptual themes and ontological concepts. This is very useful for phishing detection and can be used for several different methods [10].

NLP is particularly useful in phishing detection, as it enables models to assess the language and tone of email content. By analysing sentence structure, tone, and context, NLP-based algorithms can detect the specific language often used in phishing emails, improving the classifier's accuracy and helping it evolve with new phishing tactics [11].

The bag-of-words (BoW) model is a foundational approach in NLP for text representation, often used in text classification tasks such as phishing detection. In the BoW model, a text is represented as a collection of its words, disregarding grammar, word order, and sentence structure while focusing solely on the presence or frequency of each word in the document. This representation allows for straightforward analysis of word patterns, making it effective for applications where specific keywords or terms might indicate phishing attempts [12].

While simple, BoW can be powerful when paired with machine learning algorithms, as it helps them identify significant word patterns that differentiate between legitimate and malicious emails [13]. However, BoW has limitations in capturing context or semantics between words, so it's often enhanced with different techniques or used alongside more sophisticated models, such as word embeddings, to improve accuracy in complex language tasks [14,15, 16].

NLP offers numerous other advanced methods for email filtering beyond the traditional BoW approach. Techniques such as word embeddings (e.g., Word2Vec [17] or GloVe [18]) allow models to capture the semantic meaning of words by representing them as vectors in a continuous space, which helps identify similar words or phrases even if they don't appear in the same form [19].

In a study carried out by Bhowmick and Hazarika (2016) [20], ML and DL algorithms, combined with NLP techniques are explained. According to this study, even though NLP techniques developed on ML provide good results, it is still highly dependent on surface text instead of deep semantics. Thus, when the structure of the text is changed with synonyms, it is difficult for these methods to detect phishing. It is also stated that DL is more effective in processing emails more accurately and efficiently than ML methods.

2.2. URL Analysis

URL Analysis can also be a good strategy for phishing detection since it allows the detection of phishing attacks initiated through malicious URLs embedded in emails by analysing the structure, content, or behaviour of the URLs. There are several different approaches to URL Analysis, including domain reputation checks, blacklisting, whitelisting, behavioural analysis, and pattern recognition [21].

In the context of combating phishing attacks, two prominent methodologies—reputation-based assessments and behavioural analysis—serve critical functions in identifying and mitigating potential threats by utilizing historical data and analysing user interactions.

Reputation-based Methods:

Reputation-based methods involve assessing the credibility or trustworthiness of a sender, domain, or website based on historical data and known behaviours. These methods are particularly useful for phishing detection by evaluating the following:

- **Domain Reputation**: If a domain is newly registered or has been associated with previous phishing activities, it may be flagged as suspicious. Phishing websites often use newly created or rarely used domains to bypass detection.
- **IP Reputation**: Email senders from IP addresses with a poor reputation or those that have been involved in previous phishing attacks can be identified and blocked.
- **Sender Reputation**: Emails from senders that frequently send unsolicited or malicious emails are flagged based on their past behaviour and known phishing activities.

These reputation-based methods often rely on pre-established databases of known or potentially suspicious URLs to assess whether a given URL is trustworthy or harmful. These methods employ blacklisting to identify and block suspicious URLs and whitelisting to confirm and allow access to trusted ones. While these techniques are highly efficient and accurate in detecting known threats, they face challenges such as the ongoing effort required to maintain up-to-date reputation databases and their inability to identify new or previously unknown threats [22].

Prakash et al. [23] introduce PhishNet, a system designed to tackle the limitations of URL blacklisting. It exploits the observation that attackers often employ simple modifications to existing URLs. PhishNet has two key components: (1) a heuristic-based mechanism that generates variations of known phishing URLs to uncover new ones, and (2) an approximate matching algorithm that breaks URLs into components for individual comparison against the blacklist. The system suffers from low false positives and is remarkably effective at flagging new URLs.

2.3. Behavioral Analysis

Behavioural analysis for phishing detection focuses on studying the actions and interactions of users, websites, or emails to identify suspicious or abnormal behaviours that might indicate phishing attempts. Understanding user's behaviour when presented with phishing emails is highly relevant in combating phishing attacks. Beyond implementing technical defences, understanding how users interact with suspicious emails can reveal critical insights into areas of vulnerability and help shape effective preventive strategies. User behaviour plays a crucial role in the effectiveness of phishing defences, as even the most advanced security measures can be undermined by human error.

Different strategies can be used in this context, namely:

2.3.1. Email Behaviour

Behavioural analysis looks at how emails behave, such as whether they prompt users to click on suspicious links or request sensitive information under urgent circumstances (e.g., password resets,

financial transfers). These are common phishing behaviours. Salloum et al. [9,11], Li et al. [24], and Tamal et al. [25] emphasize the recurring use of urgency, fear, and authority in phishing tactics, with users often overlooking critical details like typos in URLs or sender addresses, allowing attackers to create convincing imitations of legitimate sources. To counter these threats, ML and DL approaches have been widely applied. Traditional ML algorithms, including SVM, Naive Bayes, and Random Forest, focus on extracting features from email headers, content semantics, and URLs to identify phishing attempts. Advanced DL models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs), analyse textual and structural anomalies, offering improved accuracy and adaptability against complex phishing strategies. Despite these advancements, studies like Thakur et al. [26] and Catal et al. [27] underscore challenges such as limited dataset diversity and the inability of models to adapt effectively to multilingual or evolving phishing tactics.

Experimental studies demonstrate that interventions and training can enhance user performance in identifying phishing emails. Li et al. and Tamal et al. find that targeted warnings and monetary incentives improve user attention and accuracy, although the effects are often short-lived. These findings highlight the importance of adaptive and continuous training mechanisms to foster long-term behavioral improvements. Furthermore, the integration of behavioral patterns into detection systems has been shown to significantly enhance their robustness, as noted in Salloum et al., Thakur et al., and Catal et al.

All these studies agree on the importance of expanding dataset diversity and adopting unsupervised and semi-supervised learning approaches to improve detection systems. Real-time systems that integrate technical and behavioral insights are critical for addressing the growing sophistication of phishing attacks. Future research should also prioritize developing language-agnostic models and adapting to evolving phishing strategies to enhance the generalizability and robustness of detection technologies. By combining behavioral analysis with advanced ML and DL models, phishing detection systems can better address the human factors that attackers exploit, ultimately reducing user susceptibility to cyber threats.

2.3.2. Website Behaviour

Behavioral analysis can examine how a website behaves once a user land on it. For example, phishing sites might try to quickly capture login credentials or personal information through suspicious forms or redirects. Detecting these behaviours helps block phishing attempts. When the literature was examined, the authors published many papers on the website behaviour.

Ravindra et al. [28] focus on behavioral analysis through URL features, examining suspicious patterns such as the misuse of HTTPS to mimic legitimacy, subdomains, and special characters. By employing the Random Forest algorithm, their system achieves 86% accuracy in classifying phishing websites, leveraging critical features like URL length and the presence of phishing-related terms. Their findings emphasize the significance of analyzing user-facing interactions, such as rapid redirects and deceptive page structures, to bolster detection methods.

Çolhak et al. [29] introduce an advanced model that integrates NLP with a MLP to analyze HTML content and webpage behavior. The study identifies key phishing traits, including hidden forms,

excessive external resources, and obfuscated JavaScript, all hallmark behaviors of malicious websites. Their hybrid MultiText-LP model achieves an impressive 97.18% accuracy, demonstrating how combining textual and structural analyses can significantly improve phishing detection capabilities.

Opara et al. [30] present the WebPhish model, which employs deep learning techniques to process raw URLs and HTML content in an end-to-end manner. This approach captures intricate behavioral patterns, such as suspicious embeddings and page layouts. Achieving a remarkable 98.1% accuracy, WebPhish removes the need for manual feature engineering, showcasing the scalability and effectiveness of integrating website behaviors into detection systems.

In addition to these groundbreaking studies, survey papers have explored phishing detection in detail, focusing on behavioral features like HTML redirects, hidden forms, and deceptive links [31,32,33,34,35]. These surveys delve into the utilization of ML and DL models, highlighting their ability to analyze and mitigate evolving phishing tactics effectively.

2.3.3. User Interaction

Phishing attacks often rely on manipulating user behavior, like clicking links or downloading attachments. Analyzing how users respond to certain email formats or actions (e.g., unusually urgent requests) can help detect phishing attempts.

When the literature has been reviewed, it is seen that there are several papers about user interaction. Through on-site and online experimental designs, Li et al. [36] examine user interaction during phishing email detection. ML approaches are used to assess user interaction and enhance phishing defences. Key behavioral variables, including sorting accuracy, mouse movement, and response times, were gathered for study while participants were asked to distinguish between phishing and authentic emails. An important factor in determining a person's vulnerability to phishing is user interaction. On the other hand, there are some surveys in this topic. Salloum et al. [9,11] emphasizes the crucial role that user interaction plays in phishing with an emphasis on how attackers take use of psychological characteristics, cognitive biases, and emotional triggers like urgency, personality factors, cognitive styles, and security awareness all affect user vulnerability; short-term resilience is enhanced by training interventions such as simulated exercises. However, long-term awareness is still difficult to achieve and calls for ongoing, flexible instruction. To create efficient, user-centric phishing defences, the study places a strong emphasis on combining behavioral insights with machine learning and encouraging interdisciplinary collaboration.

Aldakheel et al. [37] use more conventional techniques like Random Forest and RNNs, this paper presents a CNN-based phishing detection system that focuses on URL analysis and achieves 98.77% accuracy on the PhishTank dataset. The model analyzes URL features including length and domain properties using seven optimal layers. It considers insights from user interactions, which show that people frequently rely on lightweight indicators like logos or well-known words, leaving them open to smart phishing techniques. The system improves detection capabilities by identifying behavioral patterns in user responses to phishing content. By combining technical solutions with user behavior analysis, a strong, practical phishing detection solution is provided, addressing both technical and human vulnerabilities.

These studies display the importance and usage of human interaction with the help of ML and DL techniques.

Other important practice that companies can use to prevent phishing attacks is to educate and train employees. This training typically includes recognizing common phishing tactics, such as urgent requests, suspicious links, unusual sender addresses, and unexpected attachments. Companies often simulate phishing attacks by sending realistic mock emails to test employee responses in a controlled environment. These simulations help identify users who require additional training and reinforce good security habits among the broader team [38,39].

Several other factors can influence the user's decision-making when faced with phishing attacks [40]. For example, individuals under time pressure or facing high-stress situations may be more likely to overlook warning signs of phishing, such as minor inconsistencies in email addresses or unexpected requests for personal information [41]. Understanding these behavioural patterns can allow organizations to tailor their training and policies to account for real-world pressures that users face, enhancing overall security [42].

2.4. Content Analysis

Content Analysis involves examining the content within an email to identify phishing attempts. It mainly consists of inspecting textual and visual elements in the email body to detect patterns, keywords, or images commonly associated with phishing. This approach can help identify phishing emails that lack obvious malicious URLs or malicious attachments.

2.4.1. Textual analysis techniques

Textual Analysis Techniques involve examining the text content of emails for signs of phishing. It can include analysing the language style, tone, spelling, and grammar, as well as specific keywords often used in phishing attempts, such as "urgent," "password," or "click" [43,44]. More traditional textual analysis techniques use rule-based methods and heuristics to make these detections. Creating and fine-tuning these methods can often be very complex and time-consuming [45]. These rules and heuristics normally depend on known patterns, which generally makes these methods unable to detect zero-hour phishing attacks [46]. More recent textual analysis tools, including ML and NLP, were used to assess linguistic features, such as threatening or manipulative language or unusual language patterns, and detect discrepancies that suggest the email is not from a trusted source [47].

For example, in the study of Bhowmick and Hazarika (2016) [48], 'bag-of-words' method is used for selecting content features. The bag-of-words method is used to represent textual data by focusing on word frequency while ignoring grammar and word order. Each word is treated as a distinct feature, and the frequency of these words is counted.

In another study carried out by Sheneamer (2021) [49], (as mentioned earlier) deep learning and traditional ML algorithms, combined with NLP techniques, are explained and compared for phishing detection. According to this study, deep learning methods provided better results than traditional ML methods. Deep Learning methods use converting words into vectors via word embeddings, and this method captures semantic relationships between words but machine learning methods like

'bag-of-words' can't. This difference allows for deep learning methods to reach deeper contextual information and higher performance.

In this study, it is also stated that semantic processing is essential for phishing detection since it is required to understand the intention of the email sender. For example, personal information obtained from social media can be taken advantage in a phishing email. Such an activity may not include attachment or link. So, it becomes more difficult to detect phishing. However, semantic analysis is still not enough all the time.

Altwaijry et al. [50] explores the application of deep learning techniques to detect phishing emails. Specifically, it examines various architectures of one-dimensional convolutional neural networks (1D-CNNs) augmented with recurrent layers such as Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Gated Recurrent Units (GRU), and Bidirectional GRU (Bi-GRU). These models were trained and tested using Nazario Phishing Corpus and SpamAssassin. The 1D-CNNPD augmented with Bi-GRU achieved the best results, reaching a 99.66% F1 score. Even though the semantic-based approach is effective, it is also stated that non-semantic features like presence of attachments, number of URLs in the email, proportion of symbols in email body etc. should also be considered to reach better and more promising results.

Text-based phishing detection algorithms are designed only to analyse different components of an email. However, they are not effective for emails that contain images rather than texts. In order to take advantage from this, usage of images that contains phishing content is rapidly spreading. Nahmias et al. [51] proposes using an ensemble of LLMs, prompted with human-crafted questions, to detect spear-phishing attacks. The LLMs provide probability-based answers combined into document representation vectors for each email. This study uses three different datasets, Enron Corpus and SpamAssassin for training and Realistic Gen SPH, presented and made available in this paper for testing. Despite the limitations pointed out by the authors, the presented approach showed improvements over more traditional ML-based approaches, reaching a 91% F1 score.

2.4.2. Image Analysis

Phishing emails often use brand logos, layout styles, or other visual cues to deceive recipients. Image recognition techniques focus on identifying malicious images or logos that mimic legitimate brands, such as altered, low-quality, or replicated ones [52,53]. This technique can detect these images even if they are embedded or obfuscated within the email [54,55,56].

Some phishing attacks can use QR codes to deliver malicious payloads, initiate phishing attacks, or redirect users to fraudulent websites. This is used to bypass other phishing detection techniques or even some approaches to this one. Despite this, image recognition can also include malicious QR code detection to avoid these scams. Ford et al. [57] used Convolutional Neural Networks (CNNs) to tackle this issue and detect these malicious QR codes. The CNNs were initially trained using carefully crafted datasets containing several kinds of QR codes and non-QR code images. The authors also made efforts towards continuous improvements to the model.

Other phishing attacks create an image from the text message or email to bypass any textual analysis and avoid detection. Some approaches to image recognition tackle this problem by extracting dominant text from images into textual-based features [58,59].

According to Kumar et al. (2018) [60], Optical Character Recognition (OCR) methods extract the text from the image and then passes to a trained phishing classifier. However, phishers have applied various image processing techniques such as changing foreground, background, text font size and

colour, which made the OCR based methods obsolete. Then the focus on phishing detection from image content shifted to deep learning algorithms.

Zhang et al. [61] also proposes a phishing detection framework that uses OCR to extract textual content from images on phishing web pages. This extracted text is used alongside URL-based, web-based, and rule-based features in a two-stage Extreme Learning Machine (ELM) model. This method detects phishing attempts, even hiding information in images, as validated by experimental results.

In the study carried out by Abuhammed and Abuzaid in 2022 [62], ML and DL techniques are compared for fake image detection. As result of that study, it is found that CNN structure is best at classifying phishing content in images. In other words, it is possible to detect a phishing email that contains mostly images without using any NLP technique. CNN is a type of deep learning model that is particularly effective in analyzing visual data, such as images. In the context of phishing email detection, where images are predominantly used instead of text, CNNs are useful because they can automatically extract and learn features from the images, such as patterns, shapes, and textures, to determine whether an image contains phishing content.

2.5. Machine Learning and Artificial Intelligence Applications

As has already been said in the previous sections, the use of ML and AI for phishing detection is currently a hot topic. ML has several advantages used for phishing detection, but it also has its limitations. Supervised ML models need datasets to be trained, and their accuracy and reliability depend on the data they contain.

Datasets are sets of data relevant to a specific topic. The data can be artificially generated, derived from existing datasets, or collected from different sources. Datasets are labelled if they contain a classifying feature or are not labelled. In supervised learning, labels identify each sample for model training and evaluation.

The following table contains some of the existing open-source datasets used in ML phishing detection along with their content, label information, sample size and description.

Dataset	Content	Label	Sample size	Description
Enron Corpus [63]	Email	No	517401	A collection of emails generated by Enron Corporation's employees. It was obtained by the FDRC during its investigation of Enron's collapse.
Realistic Gen SPH [64]	Email spear phishing	-	7156	Generated by a company using a proprietary system, containing very realistic spear phishing emails

Dataset	Content	Label	Sample size	Description
Nazario Phishing Corpus [65]	Email phishing	Yes	7,315	A publicly available phishing email set originally compiled by Jose Nazario
SpamAssassin [66]	Email spam	Yes	5809	Phishing/Ham emails taken from the SpamAssasin dataset
Nigerian-5 [67]	Email phishing	Yes	6331	This dataset is a collection of more than 2,500 "Nigerian" Fraud Letters, dating from 1998 to 2007
TREC-05	Email phishing	Yes	56334	The TREC Public Corpus is a collection of email messages collected between 08/04/2005 and 06/07/2007. The TREC datasets are the largest publicly available datasets [68]
TREC-06	Email phishing	Yes	16451	
TREC-07	Email phishing	Yes	53757	
CEAS-08	Email phishing	Yes	39154	CEAS 2008 was a Live Spam Challenge Corpus in 2008 [69]
Ling [70]	Email Spam	Yes	2893	The dataset was made publicly available as a part of the paper [70]
URL4S [71]	URL	Yes	808042	Compilation of the body text of various email, generated by Enron Corporation employees
Email4S [71]	Email phishing	Yes	18650	A collection of data samples from various sources, including the JPCERT website, existing Kaggle datasets, GitHub repositories and other open-source databases
Phishing Detection Dataset [72]	URL	Yes	247951	The dataset contains 41 different features extracted from phishing and legitimate URLs using a technique called OFVA

Dataset	Content	Label	Sample size	Description
HELPHED Dataset	Email phishing	Yes	35511	The dataset was created purposefully to tackle issues found on the available phishing datasets during the development of the paper [73]
UCI Phishing Repo [74]	URL	Yes	235795	The dataset was generated by extracting different features, related to legitimate and phishing websites, from Phishtank
Phishing Websites Dataset	URL	Yes	80000	-

With the proper dataset, ML can be used for several different approaches to phishing detection. Some of these approaches are described below.

2.5.1. User and System Behavior Analytics

This refers to analyzing both user and system actions to identify patterns, detect anomalies, and predict future behavior. By looking at the interactions between users and the system, this type of analytics helps in identifying potential security threats, system performance issues, or unusual behavior that may signal fraud, breaches, or other risks.

Together, these techniques and methods enhance security by identifying both known and unknown threats through behavior monitoring and pattern recognition. The focus is on how users interact with applications, data, and networks over time.

Therefore, this is very related with the behavioural analysis mentioned previously mentioned. ML algorithms are trained to learn about the user's normal behaviour and to distinguish it from its behaviour when presented with a phishing email. The trained algorithm can then be used to enhance phishing awareness training or detect users who may require more training. This data can also be useful for assessing the impact of the training on the user's behaviours [75].

2.5.2. Anomaly Detection and Pattern Recognition

Anomaly detection is the process of identifying deviations from established patterns of normal behavior. Pattern recognition involves recognizing regularities or trends in data. Due to its nature, ML is highly effective for anomaly detection and pattern recognition. By training models on large datasets of both legitimate and phishing emails, ML algorithms can learn to recognize common

characteristics and behaviours associated with phishing, such as certain phrases, sender addresses, or email structures [76].

It can also be trained on normal email traffic and learn to detect anomalies, such as a sudden increase in emails from unfamiliar domains or irregular patterns in how users interact with emails. Additionally, as the system continues learning from new data, it becomes more adept at recognizing emerging phishing tactics, making it a dynamic and adaptive tool for combating phishing attacks [77,78].

Andriu et al. [79] developed an adaptive phishing detection system using machine learning and natural language processing to address the limitations of traditional methods against sophisticated, AI-driven phishing attacks. The system incorporates advanced techniques for real-time email security and proposes solutions like reinforcement and federated learning to tackle challenges such as scalability and privacy compliance. Experimental results showed superior performance, highlighting the system's potential to enhance cybersecurity in the face of evolving threats.

Bountakas et al. [80] propose a novel phishing email detection methodology called HELPHED, which combines Ensemble Learning methods with hybrid features to improve detection accuracy. The approach integrates both content-based and text-based features, offering a more comprehensive representation of emails. Two Ensemble Learning techniques were compared on HELPHED, Stacking and Soft Voting. Experimental results demonstrate that HELPHED using Soft Voting significantly outperforms existing methods, achieving an F1-score of 99.42%.

2.5.3. Heuristic and Behavioural Analysis

Heuristic analysis involves using rules or algorithms to evaluate and detect potentially malicious behavior based on known patterns or behaviors, even if the specific threat is not identified by traditional methods. Behavioral analysis looks at how systems or programs behave over time to detect suspicious actions, focusing on how software or users interact with a system to identify potential threats.

One of the main drawbacks of rule-based methods is that creating and fine-tuning the rules and heuristics is very complex and time-consuming. ML can solve this problem as it is great at pattern recognition and can generate a set of rules and heuristics from a dataset [81,82,83].

2.5.4. Endpoint Behavior Monitoring

This refers to tracking and analyzing the behavior of individual devices (endpoints) like computers, mobile devices, or servers. By monitoring how these devices interact with the network and other systems, organizations can detect signs of compromise or irregular activity, such as unauthorized access or malware infection.

3. Advanced Email Security Measures

3.1. Email Security Protocols, Techniques and Frameworks

DMARC (Domain-based Message Authentication, Reporting & Conformance) is an email authentication protocol designed to help prevent email spoofing and phishing attacks that are built

upon two existing protocols, SPF (Sender Policy Framework) and DKIM (DomainKeys Identified Mail), by providing an additional layer of security [84].

While SPF allows domain owners to specify which mail servers are permitted to send an email on behalf of their domain, DKIM allows the sender to sign their emails with a digital signature, which can then be verified by the recipient to confirm the authenticity of the email and its content. Both email authentication methods ensure that email messages are sent by authorized servers and have not been tampered with during transit. SPF does this by checking the domain's SPF record when an email is received to find out if the sending mail server's IP address is authorized [85]. If this IP address is listed in the SPF record, the email passes the SPF check; however, if the IP is not listed, the email is flagged as suspicious, potentially being marked as spam or rejected. DKIM also flags emails that have been potentially tampered with, but it does this based on whether the signature of the email matches the private key incorporated by the sender's email server [86].

Despite both SPF and DKIM protocols independently verify different aspects of an email's legitimacy, when used together they provide a stronger defense against email fraud. On one hand, SPF validates that the sending server is authorized to send email on behalf of the domain, on the other hand, DKIM ensures that the email's content has not been modified in transit and verifies the identity of the sender. Together, they help improve email security, making it difficult for attackers to impersonate a legitimate domain or alter an email's content.

Adding on to these protocols, by enabling domain owners to publish a policy in their Domain Name System (DNS) records, specifying how email servers should handle messages, DMARC can instruct the server to either accept, quarantine, or reject emails that do not pass the checks [87]. Moreover, DMARC helps monitor the effectiveness of email security measures by allowing domain owners to receive reports from receiving mail servers about email authentication results.

Furthermore, DNSSEC (Domain Name System Security Extensions), a set of extensions to the DNS, can help enhance security and prevent certain types of attacks, such as DNS spoofing (also known as cache poisoning) and man-in-the-middle attacks [88]. This set of extensions works by ensuring the authenticity and integrity of the data returned by DNS servers, which are responsible for translating human-readable domain names into IP addresses [89]. Using digital signatures and cryptographic techniques to verify that the DNS responses a user receives are authentic and have not been tampered with during transmission, DNSSEC addresses multiple security issues. The main process revolves around cryptographic keys. Thereby, when a domain owner adds these extensions to their domain, they sign the DNS records (such as A, MX, TXT records) with a cryptographic key that is then stored in the DNS records themselves. Each domain has a private and public key pair; one is used to sign the records, and the other is published in the DNS for verification purposes, respectively. When a resolver (the server that answers DNS queries for end users) receives a DNS response, it checks for a signature and if the response is signed and the signature matches the public key from the DNS, the data is considered valid. However, if not, the query is considered compromised, and the response is discarded or flagged as invalid. Following this process, DNSSEC helps prevent attackers from intercepting and altering DNS queries to redirect users to fraudulent websites and can help secure online banking, email, and other services that rely on DNS for routing, increasing overall internet security by ensuring that users are directed to legitimate websites. In any case, there are some worth-mentioning limitations. For instance, DNSSEC adoption requires changes to DNS infrastructure, and not all domains and resolvers support it; moreover, while

DNSSEC protects the integrity of DNS data, it does not encrypt DNS queries themselves, so it does not protect against eavesdropping.

As a complement to the abovementioned security protocols, DNS “sinkholing”, a common cybersecurity technique used to redirect malicious or unwanted DNS requests to a controlled server (a "sinkhole") - instead of their intended malicious destination – can help mitigate threats like malware, botnets, and phishing.

The other two security protocols frequently used, particularly in Windows-based environments, are SMB (Server Message Block) and RDP (Remote Desktop Protocol). Despite both having legitimate uses, they are also often targeted in cyberattacks.

SMB is mainly used for sharing access to files, printers, and other network resources in a local area network (LAN) or over the internet, facilitating communication between computers, by allowing users to access files on remote servers or devices as if they were local files [90]. Additionally, by allowing user authentication to control access to resources, it ensures that only authorized users can read/write files or use printers. Although being most generally used in Windows environments, SMB protocol is also supported by other operating systems such as Linux and macOS.

Likewise, RDP allows users to connect to another computer remotely and interact with its graphical user interface (GUI) providing full access to a remote system, enabling users to control it as if they were physically present [91]. Unlike text-based remote access protocols (e.g., SSH), RDP transmits the graphical interface of the remote computer, including the desktop, windows, and applications, and uses encryption to secure communication between the client and server, protecting data in transit. It is particularly adequate for IT administrators to troubleshoot and manage remote systems, and useful for administrators to manage remote servers and configure them.

Regarding open-source framework solutions, Rekall and Volatility are both memory forensics frameworks that are used to analyze the contents of computer memory (RAM) to investigate cyber incidents, perform digital forensics, and uncover malicious activity. Rekall is an open-source digital forensics and incident response (DFIR) framework primarily used for memory forensics [92]. Volatility is a popular open-source framework for memory forensics and digital investigations mostly used to analyze a system's volatile memory (RAM) to uncover artifacts related to malicious activity, system misconfigurations, or forensic evidence in cybersecurity investigations [93].

While both have different features and implementations, they also share several common aspects. For instance, both are designed to analyze raw memory dumps, proving crucial for forensic investigations to recover data about running processes, network connections, open files, loaded drivers, and more at the time the memory was captured, and both are commonly used in malware analysis, especially for identifying rootkits, process injection, or memory-resident malware. However, built with Python 2.x and 3.x compatibility in mind, Rekall is designed to be a more modern, flexible, and faster alternative to Volatility, supporting multiple memory acquisition formats.

Equally worth mentioning among email security measures is YARA (Yet Another Ridiculous Acronym), a tool used to identify and classify malware by defining rules based on patterns, strings, or characteristics in files [94]. Commonly used in malware analysis and threat hunting, YARA contributes to defending against phishing in multiple ways [95]. Firstly, as phishing campaigns often

include malicious attachments (e.g., malware-infected documents, scripts, or executables) designed to compromise a user's device, by creating rules to identify specific patterns in file headers, strings, or known malware indicators, YARA can analyze email attachments and flag potentially harmful content. Secondly, not only can YARA be used to scan email attachments and links for signatures of known malware strains by matching them against pre-defined or community-shared rules, but it may also be employed to scan email attachments and links; if such links and attachments are proven connected to malware, security systems may block those emails.

Phishing campaigns often use pre-built phishing kits to create fake login pages or email templates and often exhibit repeatable patterns or behaviors (e.g., specific phishing email formats, document metadata, or encoding techniques). YARA can analyze phishing kit files (HTML pages, images, or scripts) for patterns commonly found and can target these patterns, allowing organizations to detect phishing campaigns linked to specific threat actors. Finally, when used alongside email authentication protocols like DMARC, DKIM, and SPF, YARA can provide an additional layer of inspection by examining email content and attachments for malicious payloads, complementing an already validated sender's legitimacy.

3.2. State of the Art Technology for Email Security

ML and AI have been extensively applied in email security and network attacks prevention, detection, and remediation, particularly in combating phishing and improving Security Information and Event Management (SIEM) systems.

3.2.1. Email Security Technology State of the Art

Microsoft Defender for Office 365 [96] uses ML and AI to scan incoming emails for phishing attempts by evaluating email content, URLs, and sender behavior. This security solution is designed to protect email, and collaboration tools and provides multiple layers of protection. Its key functionalities include threat protection, which allows to detect and mitigate email-based threats, including phishing, impersonation, and malicious attachments or links; Automated Investigation and Response (AIR), reducing manual intervention; real-time detection employing AI and ML to detect emerging threats in real-time; post-breach investigation and remediation tools; Threat Intelligence offering insights into the latest attack methods and tactics to help organizations stay proactive; and overall collaboration security, extending protection to Microsoft Teams, SharePoint, and OneDrive for Business.

As part of the services provided by the Microsoft Defender solution, Microsoft Exchange Online Protection (EOP) [97] is a cloud-based email filtering service that is designed to protect email users from spam, phishing, malware, and other malicious content. EOP is an integral part of Microsoft Exchange Online and works to safeguard organization's email system by applying security filters and policies to inbound and outbound email traffic. EOP protection is on by default thanks to policies for anti-malware protection, anti-spam protection, and anti-phishing (spoof) protection that cannot be disabled. Nevertheless, these policies can be overridden by preset security policies or custom policies created.

Similarly, Google's AI-enhanced spam and phishing filters for Gmail [98] block phishing emails by analyzing content, sender behavior, and link safety in real-time. This solution provides robust protection for Gmail users against a wide range of online threats, including spam, phishing attempts,

and malware, by using filters developed with advanced ML and AI to detect subtle patterns in emails, including deceptive tactics. Alongside spam and phishing detection, this solution uses AI to identify and block emails containing malicious attachments, reducing the risk of malware infections, flags dangerous links in emails so that users are warned before visiting suspicious websites, and encrypts emails both at rest and in transit, ensuring a high level of privacy and security. Additionally, it provides proactive alerts if attachments or links might compromise security and, as the system learns and updates its capabilities based on new threats, it also provides continuous improvement in security.

3.2.2. Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS)

Regarding IDS and IPS, ML models learn normal network behavior patterns (e.g., typical traffic, protocols, and device interactions) and flag any deviations as potential intrusions, and AI models detect patterns of malicious activity such as port scanning, brute-force login attempts, or data exfiltration, allowing for the deployment of immediate preventive measures (e.g., firewall rule updates or network isolation).

Solutions such as those provided by Darktrace [99] use AI to create a real-time understanding of network activity, detecting unusual behavior that may indicate an ongoing attack, helping organizations transition from reactive security to proactive resilience. By leveraging self-learning AI to continuously monitor and analyze data within a business's unique digital environment, it can tackle phishing, account takeovers, and insider threats across email and collaboration tools like Microsoft Teams. Its AI-driven features are designed to improve early-stage phishing detection, identify signs of account compromise across a wide range of communication platforms, and enhance the efficiency of security operations centers (SOC). Similarly, Cisco's AI-driven solutions [100] employ ML models in both IDS/IPS and firewall products to identify and block sophisticated attacks like ransomware or botnet activity.

3.2.3. Security Information and Event Management (SIEM) Systems

Security Information and Event Management (SIEM) systems have been transformed by AI and ML from reactive, log-based systems to proactive, threat-hunting platforms. For instance, AI and ML algorithms help automate the detection of sophisticated threats that would otherwise require manual investigation by automatically analyzing threat data in real-time and generating alerts when an attack pattern or anomaly is identified.

Examples of such technology include the solution provided by Elastic SIEM [101], that uses AI to detect outliers, unusual system access, and malicious patterns in real-time, enabling organizations to detect, investigate, and respond to security threats efficiently. Like, Splunk [102] integrates AI and ML to collect, index, and visualize data in real time, to automate the detection of suspicious network behavior and perform event correlation, using predictive analytics to foresee potential security incidents based on current data trends.

Adding on to these solutions, User and Entity Behavior Analytics (UEBA) is of major importance. Exabeam [103], a UEBA-integrated SIEM solution, uses AI to track abnormal user behaviors, such as unusual login times or unauthorized access to sensitive data, providing early warnings of insider threats or compromised accounts. Including advanced capabilities for log management, threat detection, and incident response, leveraging AI to enhance security operations, this solution's

behavioral analytics capabilities allow it to create baseline behavior models for users and devices, enabling more accurate detection of anomalies, including advanced threats like lateral movement. Yet, as one of the biggest challenges in SIEM is the volume of false positives, which overwhelms security teams, solutions like Securonix [104, 105], leveraging advanced ML and behavioral analytics, help drastically reduce false positives by correlating events across endpoints, users, and network activity. Additionally, by using AI to evaluate the context of an alert (e.g., location, time, behavior history) it is possible to determine whether it is likely to be malicious or benign. This platform also provides automation through Security Orchestration, Automation, and Response (SOAR), enabling faster incident management and reducing dwell time for threats.

As mentioned, a key feature in SIEM systems is Predictive Analytics Models that can forecast potential security incidents by analyzing historical patterns and trends. LogRhythm's AI Engine [106], designed to enhance threat detection, analysis, and response capabilities using advanced ML and automation, can predict security events by continuously learning from the organization's data and threat landscape. The AI Engine provided by this solution identifies security threats by correlating data from various sources in real-time, enabling the identification of anomalies and potential attacks across an organization's environment. Additionally, as it monitors and analyzes user and system behaviors, it can detect deviations from typical patterns, helping uncover insider threats, compromised accounts, or Advanced Persistent Threat (APTs).

3.2.4. Preventive Solutions

In addition to the abovementioned solutions, real-time insights, predictive analytics, and contextual threat intelligence are of major importance to organizations, helping them proactively detect and defend against phishing campaigns. By monitoring malicious activity, analyzing attacker behavior, and delivering insights in real-time, solutions like Recorded Future and IBM X-Force empower security teams to prevent, detect, and respond to phishing campaigns effectively.

Alongside these solutions, it is also paramount to invest in employee training. For example, KnowBe4 [107] and Cofense PhishMe [108] both focus on enhancing email security by addressing the human element of cybersecurity - training employees to recognize and handle phishing threats effectively. KnowBe4 provides a large library of customizable phishing templates that mimic real-world attacks, enabling organizations to train employees on recognizing phishing attempts. Additionally, the platform offers engaging, updated content like videos and interactive modules to educate users on cybersecurity best practices. Cofense PhishMe delivers realistic phishing simulations using insights from real-world threats that bypass secure email gateways helping employees practice identifying and responding to such threats. By integrating a global threat intelligence into its simulations, this solution helps ensure users are trained against the latest phishing tactics.

4. Ransomware Detection and Behaviour Analysis

Ransomware is the most widespread and visible type of malware that can lock systems, devices, or files, encrypt files on the endpoint, delete files, and close system access until a ransom is paid. Ransomware is evolving daily, and new variants, such as double extortion and triple extortion, are emerging. While double extortion ransomware encrypts and steals data, threatening to leak the stolen data if the ransom is not paid, triple extortion adds another layer, such as demanding ransom from the victim's customers or partners or launching DDoS attack [109].

The turning point of ransomware attacks was the WannaCry ransomware crypto worm in May 2017, a worldwide cyber-attack. The ransomware targeted computers running the Microsoft Windows operating system, encrypting data and demanding ransom payments in Bitcoin cryptocurrency. It affected many large organizations in 150 countries and is estimated to have infected around 200K computers [110].

According to Statista's report [111], the proportion of organizations affected by ransomware attacks worldwide has been increasing every year, from 55.1% in 2018 to 72.7% in 2023. Since 2018, more than half of survey respondents have reported that their organization fell victim to ransomware each year. The most targeted sectors are the health sector, critical manufacturing sector, public facilities sector, and manufacturing sector.

According to IBM Security's "The Cost of a Data Breach Report 2024" [112] , the average cost per incident is $4.88 million. This cost has increased by 10% compared to the previous year.

The ransomware landscape has evolved significantly over the last 2 years and as ransomware groups adapt their tactics, businesses face new challenges in protecting their networks and data. In addition, various artificial intelligence-based efforts are coming to the fore for the solution. For this reason, we focus on the studies conducted in the last 2 years.

In this study, we discuss and present the latest trends in ransomware detection and behavioral analysis of 14 ransomware families. The main contributions of this research are summarized in three stages.

- We present a behavioral analysis of 14 ransomware families in three stages.
- We analyze 8 publicly available and 9 non-published datasets, comparing the datasets and presenting the results.
- We investigate ransomware detection studies from the last two years (2023-2024) and compare a total of 12 studies.

4.1. Survey Papers

In this section, we provide an overview of notable contributions to the field of ransomware security research. Specifically, we will review surveys that analyse the characteristics and behaviours of ransomware, as well as those that offer a comprehensive examination of the existing literature on the subject. These surveys deliver valuable insights into various facets of ransomware, including its evolution, taxonomy, analysis techniques, and defence strategies.

Razaulla et al. [113] present a comprehensive survey on ransomware, covering its evolution, taxonomy, and current research trends from 2016 to 2023. They investigated 150 research papers which contain datasets, objectives, features, ML-DL algorithms, accuracy, approach, platform, and environment. Furthermore, they reveal that 72.8% of studies focused on detection, with 70% utilizing ML techniques. The survey identifies significant gaps in prediction methods and real-time protection and notes a lack of focus on adversarial ML and concept drift, which are crucial for improving ransomware defence.

Ispahany et al. [114] investigate the increasing impact of ransomware, highlighting the inadequate examination of real-time and early detection techniques. They propose a taxonomy that aligns detection methods with the Cyber-Kill Chain and identify issues with dataset inconsistencies, advocating for standardized and synthetic datasets to improve comparability. Moreover, the study points out limitations such as inconsistent reporting, validation challenges with outdated datasets, reliance on delayed detection methods, high computational costs of DL models, and infrequent updates of ML models. The authors recommend that future research focus on diverse evaluation metrics, synthetic datasets, real-time detection systems, advanced learning approaches like agent-based and incremental learning, and adversarial learning techniques.

Cen et al. [115] review early ransomware detection, focusing on identifying attacks at their initial stages for more effective prevention. Their survey of key papers since 2018 addresses ransomware evolution, attack processes, and early detection datasets. Challenges include advanced encryption techniques that complicate detection and the limitations of traditional and current ML methods. Consequently, the study emphasizes the need for standardized datasets, as inconsistent public samples are commonly used. It advocates for new detection techniques, especially for zero-day ransomware, and the development of comprehensive, standardized datasets to improve detection models.

Begovic et al. [116] focus on ransomware encryption activities during the pre-encryption and encryption phases. They identify key features for ransomware detection methods, such as application programming interface (API) calls, system calls, I/O operations, and file system activities. Meanwhile, API and system calls are used to detect real-time encryption but may produce false positives and require substantial resources. I/O operations identify unusual spikes in activity but may miss low-activity ransomware, while file system monitoring detects ransomware through file manipulation but may fail against unconventional methods.

Rehman et al. [117] categorize ransomware detection techniques into signature-based, heuristic, and ML-based approaches. They propose a hybrid detection model that combines these methods. Additionally, they present common ransomware tactics, such as social engineering, phishing, and exploiting vulnerabilities, and suggest a multi-layered defence involving technological and behavioural solutions. They advocate for pre-encryption phase detection using ML algorithms to reduce false positives and negatives.

4.2. Behavioral Analysis of Ransomware

Ransomware is a specific type of malware that aims to suspend the availability of data. To achieve their main objective, ransomware employs the "Impact Tactic" (ID: TA0040) and the "Data Encrypted for Impact" (ID: T1486) technique of the MITRE ATT&CK for enterprise [118]. Within the scope of this research, we analysed academic papers, threat reports from AV (anti-virus) vendors, and OSINT (Open-Source Intelligence) sources, including social media platforms and forums. Consequently, we summarize the behavioural analysis of 14 ransomware families in three stages and present our findings in Table I. In our previous research, which focused on IoT malware, we discovered that due to the high level of source code inheritance between the malware families, most IoT malware follows a specific attack pattern [119]. Moreover, our findings for ransomware behavioural analysis are parallel with that research, and we found that most ransomware families follow a similar attack pattern.

4.2.1. Infection Stage

Ransomware families employ a diverse range of infection methods. One of the primary infection methods utilized by ransomware is phishing emails, which contain malicious attachments or links to malicious web servers. Locky, Ryuk, Shade/Troldesh, Jigsaw, CryptoLocker, Petya, GandCrab, and Lockbit are some of the ransomware families that employ this infection method. Moreover, malvertising (malicious advertisements) directs users to websites that automatically download ransomware onto their systems. Bad Rabbit, GandCrab, and Winlock utilize this method for infecting the target. Furthermore, some ransomware families exploit vulnerabilities in target devices to deploy the ransomware. This technique is often used against specific targets, such as companies or organizations that utilize a particular service. WannaCry, GoldenEye, and NotPetya exploit the EternalBlue Vulnerability (CVE-2017-0144). In addition, Lockbit and DearCry are examples of ransomware families that employ this method.

4.2.2. Encryption Stage

Most ransomware families have adopted similar encryption techniques and algorithms. The common approach employed by ransomware families is to encrypt files on the target computer using a symmetric cipher, and then encrypt the symmetric cipher's key using an asymmetric cipher. The AES algorithm is the most commonly used symmetric cipher among ransomware families. In our collection, except for Petya, NotPetya, GandCrab, and Winlock, all ransomware families utilize AES. Conversely, Petya, NotPetya, and GandCrab employ the Salsa20 algorithm. Winlock is a locker-type ransomware, and its primary goal is to demand ransom by locking the target machine. For this reason, early versions of Winlock do not have an encryption function; however, the latest versions of the ransomware also encrypt files using the RC4 algorithm.

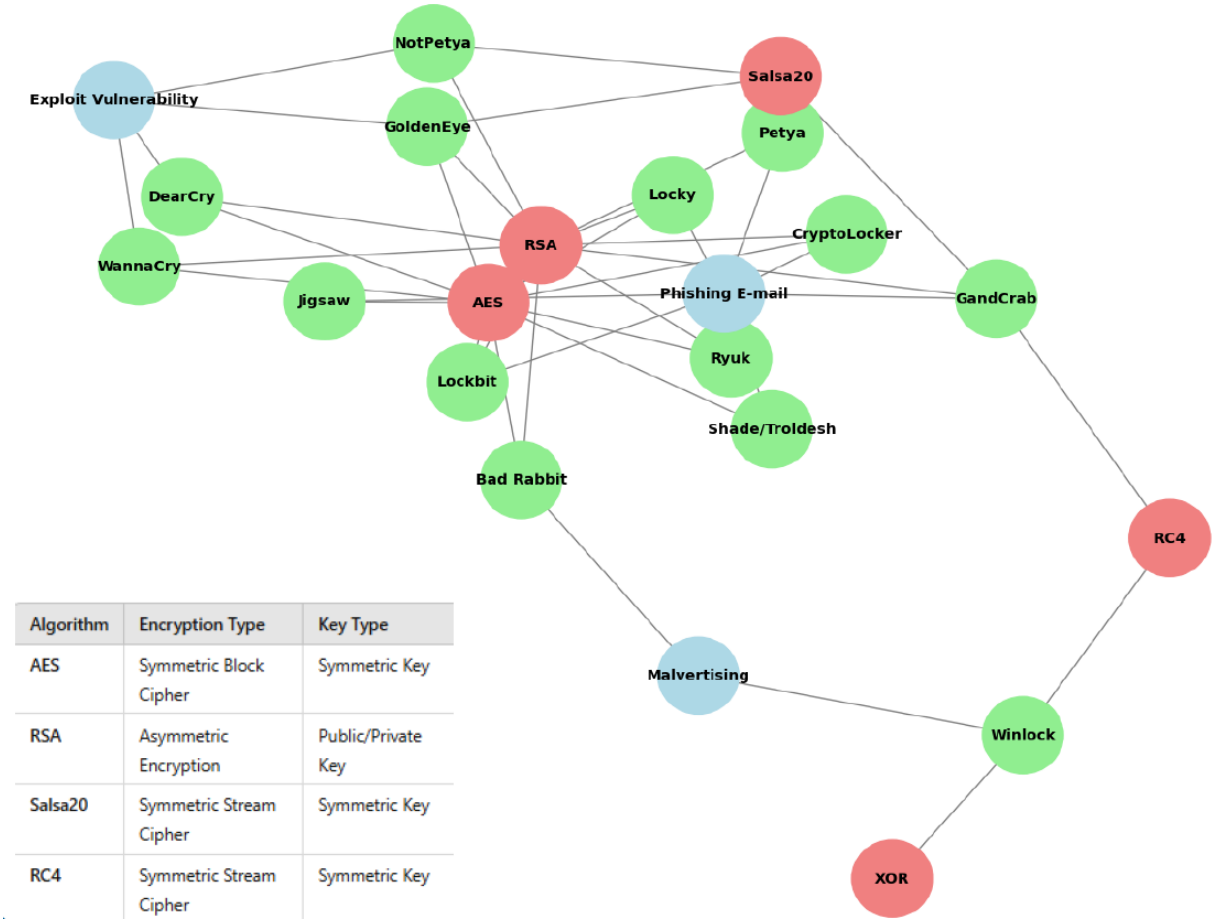
TABLE I
RANSOMWARE BEHAVIORAL ANALYSIS

Ransomware	Infection	Encryption	Other Activity
Locky	Phishing E-mail	AES and RSA	Directs victims to a TOR page for getting instructions for the ransom
Ryuk	Phishing E-mail	AES and RSA	Avoids encrypting certain file types (.exe and .dll) Deletes all shadow files and backups
Shade/Troldesh	Phishing E-mail	AES	-
Jigsaw	Phishing E-mail	AES	Disable Windows Firewall AES Key Revealed
CryptoLocker	Phishing E-mail	AES and RSA	-
Petya	Phishing E-mail	Salsa20 and RSA	Encrypts the MFT, Modifies MBR by XOR, Operation (Key: 0x37)
GandCrab	Phishing E-mail, Malvertising	Salsa20, RSA and RC4	Directs victims to a TOR page for getting instructions for the ransom. Deletes all shadow copies

Ransomware	Infection	Encryption	Other Activity
Lockbit	Phishing E-mail, Exploit Vulnerability (CVE-2018-13379)	AES and RSA	Lateral movement through Group Policiy Objects and SMB
Bad Rabbit	Malvertising	AES and RSA	Modifies MBR
Winlock	Malvertising	RC4 and XOR	Locks user screen with an image Demands ransom via SMS messages
WannaCry	Exploit Vulnerability (CVE-2017-0144)	AES and RSA	Self propagation, Does not encrypt executable files (.exe and .dll)
GoldenEye	Exploit Vulnerability (CVE-2017-0144)	AES, Salsa20 and RSA	Modifies MBR Encrypts the MFT
NotPetya	Exploit Vulnerability (CVE-2017-0144)	Salsa20 and RSA	Modifies MBR by XOR Operation (Key: 0x07)
DearCry	Exploit Vulnerability (CVE-2021-26855 CVE-2021-26857 CVE-2021-26858 CVE-2021-27065)	AES and RSA	Creates a new file and deletes the original file after the encryption.

Some early versions of ransomware families embedded the symmetric cipher keys within their code. This situation allowed the key to be revealed through static analysis, and as a result, decryption tools for these ransomware families were developed. For instance, the AES encryption key of Jigsaw ransomware was revealed through reverse engineering. The encryption password is "OoIsAwwF23cICQoLDA0ODe==" and when converted to binary format, a 192-bit encryption key is detected. Consequently, modern ransomware families have adopted more complex encryption methods. Most ransomware families now use different keys for each file and encrypt those keys using the RSA asymmetric cipher algorithm. The public key of the RSA is embedded within the ransomware code, but to decrypt the asymmetric cipher key, the private key of the RSA is required.

Attackers utilize various methods to prevent data recovery after encryption. Almost all ransomware families modify file extensions after the encryption process, whereas some simultaneously alter the entire file name. Furthermore, some ransomware families overwrite the original file with the encrypted version or save the encrypted version as a new file and subsequently overwrite the original file with random bits.



Algorithm	Encryption Type	Key Type
AES	Symmetric Block Cipher	Symmetric Key
RSA	Asymmetric Encryption	Public/Private Key
Salsa20	Symmetric Stream Cipher	Symmetric Key
RC4	Symmetric Stream Cipher	Symmetric Key

Figure 2: Encryption Methods used by Ransomware Families

4.2.3. Other Activities

Ransomware families apply many techniques to maintain their presence or avoid detection before the encryption process. Most of the ransomware families monitor the running processes on the system after gaining initial access. In this phase, if they detect any other ransomware process, they delete their own files before taking any further action. Additionally, some malware families stop some processes before starting the encryption process in order to avoid detection.

In some cases, ransomware families take more actions if the ransom is not paid. For example, Jigsaw ransomware deletes a certain number of files after every hour and deletes all files after 72 hours. Similarly, some Doxware (Leakware) type of ransomware threaten the victims by revealing their sensitive data.

4.3. Publicly Available Ransomware Datasets

ML applications are dominating most of the research areas in Computer Science, and the Ransomware Detection domain is no exception. The key to developing an efficient Ransomware Detection ML model highly depends on the dataset used for model training. However, the lack of a comprehensive and updated Ransomware dataset makes it almost impossible for researchers to develop an efficient Ransomware Detection tool. Most researchers create their own datasets for

their research. Publicly available ransomware datasets are quite limited. Some of these are represented in Table II.

The **UGRansome dataset** [120], developed in 2021, is a specialized resource for anomaly detection, particularly in identifying zero-day attacks. It includes 207,534 samples from 17 ransomware families, categorized into Normal Behavior, Abnormal Behavior, and Cyclostationary Patterns. Furthermore, the dataset is structured with 14 attributes essential for training and testing network intrusion detection systems, featuring data such as TCP protocol, IP addresses, and SSH attack clusters.

In addition, the **DREBIN dataset** [121] is a collection of 131,611 Android applications, including 123,453 benign apps and 5,560 malware samples, designed for Android malware detection. It features aspects such as hardware usage, requested permissions, suspicious components, triggered intents, system access functions, API calls, and network communications. Adnan et al. [122] note that these malware samples exhibit ransomware-like characteristics. The dataset's features are embedded in a joint vector space.

Moreover, the **EldeRan dataset** [123] comprises 582 ransomware samples from 11 families and 942 goodware samples, designed for dynamic ransomware analysis using the Cuckoo Sandbox tool. It includes features such as registry key modifications, API call statistics, embedded strings, targeted file extensions, file and directory operations, and dropped file types. EldeRan utilizes the Mutual Information criterion to identify the most relevant features for distinguishing between ransomware and benign software.

The **CCS-CIC-AndMal-2020 dataset** [124,125], created by the Canadian Institute and the Canadian Centre for Cybersecurity, includes 400K Android apps, split evenly between 200K benign and 200K malware samples. It features 6,202 ransomware samples across eight families. The dataset supports both static and dynamic analysis: static features include activities, permissions, and system access, while dynamic features cover memory usage, API interactions, network activity, battery usage, and process interactions. To ensure accuracy, each ransomware sample is meticulously labeled based on a consensus from 70% of antivirus engines on VirusTotal.

Concurrently, the **CICAndMal2017 dataset** [126], which includes 10,854 samples (4,354 malware and 6,500 benign), focuses on Android malware analysis. Among these, 1,000 samples are ransomware, classified into 10 families. The dataset features malware and benign applications collected from Google Play (2015-2017) and installed on real devices to avoid emulator detection. Over 80 network traffic features were extracted using CICFlowMeter-V3, providing a detailed analysis of ransomware behavior and network activity.

The **SoReL-20M dataset** [127], developed by the Sophos AI group, comprises nearly 20 million files, including 1,152,354 ransomware and 18,572,643 non-ransomware samples. It features comprehensive metadata and pre-extracted features, with metadata stored in databases, and features provided in compressed numpy arrays. The dataset also includes around 10 million "disarmed" malware samples with reset optional headers and file flags. Data collection occurred from 2017 to 2019, with suggested splits for training, validation, and testing.

Additionally, the **Ransomware Detection Dataset** [128,129] contains 62,485 ransomware samples and an equal number of benign samples. It includes features from Portable Executable (PE) files such as Debug Size, Major Image Version, Export Size, and Bitcoin Addresses, with detection using YARA rules. Metadata and features are stored in SQLite3 and LMDB databases. Furthermore, around 10 million disarmed malware samples are provided for further analysis. The data were collected from January 1, 2017, to April 10, 2019.

Lastly, the **RanSAP Dataset** [130] is designed to help develop ML models capable of detecting ransomware based on behavioural analysis of storage access patterns. It comprises storage access patterns from 7 prominent ransomware samples, 5 benignware samples, and 21 ransomware variants, all executed under different conditions, including varying operating systems and BitLocker-enabled SSDs. Moreover, the dataset captures low-level input and output operations on storage devices using a type-I lightweight hypervisor named BitVisor.

TABLE II
COMPARISON OF PUBLICLY AVAILABLE RANSOMWARE DATASETS

Dataset	Features	Ransom Families	Sample Count
UGRansome	Numerical Features, Categorical Features	17	56598 Ransom, 91360 Benign, 59576 Ambiguous
Drebin	APIs, Binary Features	20	5560 Ransom, 23453 Benign
EldeRan(RISS)	APIs, Registration key, File Info., Embedded string	11	582 Ransom, 942 Benign
CCS-CIC-AndMal-2020	APIs, Permissions	8	6202 Ransom, 200.000 Benign
CICAndMal2017	Network Traffic	10	1000 Ransom, 6500 Benign
SOREL-20M	APIs, PE Header, File Activities, DLL	NA	1.152.354 Ransom, 18.572.643 Benign
Ransomware Detection Dataset	PE header, Classification Features	NA	18 categories, 62.485 Ransom, 62.485 Benign
RanSap	I/O Requests, Entropy, Hash Values, Variance of LBA, Behavioral Features	7	21 Ransom, 5 Benign

As Table II shows, the size of datasets, the diversity of ransom families and the features used are variable and play an important role in ransomware and its detection. While large datasets generally provide more reliable results, small datasets may have limitations in generalization. For example, the RanSAP Dataset offers a variety of behavioural characteristics but has a very small sample size. On the other hand, the Ransomware Detection Dataset provides a balanced sample using PE

headers and classification features. The even distribution of ransomware and harmless software samples is advantageous for analysis and modelling.

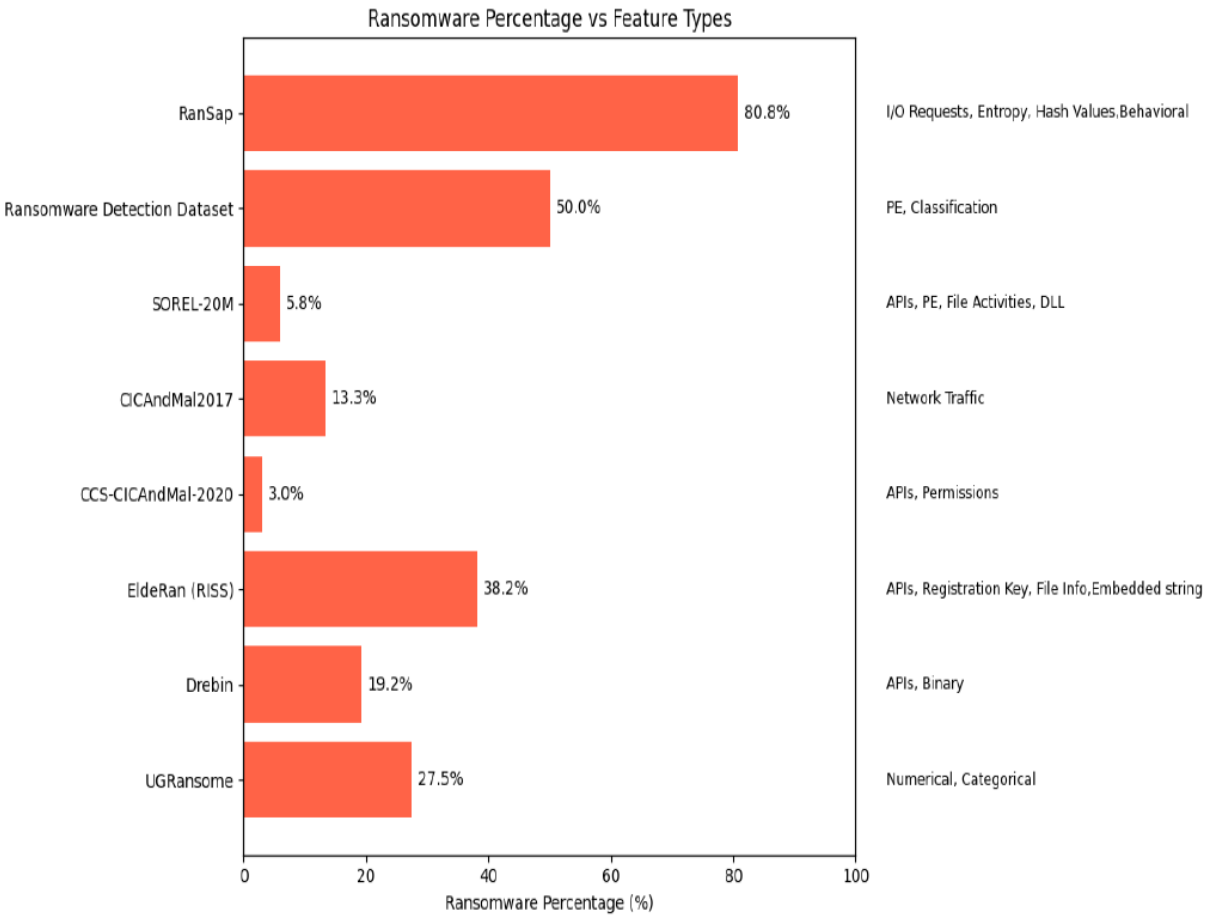


Figure 2: Publicly Available Ransomware Datasets & Ransom Percentage & Features

4.4. Non-Published Ransomware Datasets

The main downside of this approach is the impossibility of comparing the performance of the proposed method with other techniques in the literature. The studies that developed their own dataset and the specifications of those datasets are shown in Table III.

TABLE III
COMPARISON OF NON-PUBLISHED DATASET

Reference	Features	Ransom Families	Sample Count	Dataset Source
Zhang et al. [131]	IRPs, File IO Operations, System Calls	90	1206 Ransom, 4800 Benign	VirusTotal, VirusShare, Malware-Bazaar
Huertas et al. [132]	Resource Utilization, Disk I/O, Network Traffic Kernel	NA	NA	NA

Reference	Features	Ransom Families	Sample Count	Dataset Source
	Events, Syscalls			
Almoqbil et al. [133]	Network Traffic, User activity, System logs, Power Consumption, Environmental Sensor Data	NA	562 Ransom, 536 Benign	NA
Moreira et al. [134]	PE Header, DLL, Function calls, Section entropy	25	1793 Ransom, 1409 Benign	NA
Von Der Assenet et al. [135]	APIs, PE Header, Opcode, System Calls, Registry Activities, Host Logs, DLL Activities, File Activities, Network Traffic	7	NA	NA
Deng et al. [136]	PE Header	12	35367 Ransom, 27118 Benign	NA
Chaithanya et al. [137]	TF-IDF, Tokenization	21	NA	VirusTotal
Warren et al. [138]	API Calls	31	720 Ransom, 2000 Benign	VirusShare
Ciaramella et al. [139]	Opcodes	24	000 Ransom, 5000 Benign	VirusTotal

4.5. Machine Learning and Artificial Intelligence Applications

In this section, we conduct an analysis of existing studies in the literature that have applied machine learning (ML) and deep learning (DL) algorithms to ransomware datasets.

Zakaria et al. [140] present a study on the early detection of ransomware by analyzing pre-encryption API call features using ML. They propose the RENTAKA framework to detect ransomware, focusing on key behaviors during the pre-encryption stage. Notably, they achieved the best performance with Support Vector Machines (SVM).

Alqahtani et al. [141] investigate early ransomware detection in Windows environments using the eMIFS (Enhanced Mutual Information Feature Selection) technique. This method enhances detection accuracy by optimizing feature selection. In their study, they tested samples dynamically analyzed with real-time data collection and feature extraction via TF-IDF (Term Frequency-Inverse Document Frequency). SVM achieved the highest accuracy of 93%. Furthermore, the research utilized the Cuckoo Sandbox for analyzing samples, demonstrating eMIFS's effectiveness in improving ransomware detection.

Vaisakhkrishnan et al. [142] explore securing the IoMT (Internet of Medical Things) using DL techniques for attack detection. The study introduces an intrusion detection system that monitors network traffic within IoT environments. It utilizes models with the Long Short-Term Memory (LSTM) model, achieving 97% accuracy.

Rahman et al. [143] address the limitations of traditional signature-based antivirus solutions by proposing a detection model that combines behavior-based and signature-based features. This approach utilizes dynamic analysis within Windows environments. The model integrates various ML algorithms, with the Decision Tree (DT) showing the highest performance at 99.5% accuracy.

Radhakrishna et al. [144] introduce an approach for ransomware detection in IoT using ML. They develop a framework to detect ransomware at the network edge, utilizing 84 network-flow traffic features. Their method incorporates chi-square feature selection and data augmentation with SMOTE, applied to ML models such as XGB (Extreme Gradient Boosting) and RF. The results show that the XGB model achieves near 100% accuracy, outperforming other methods in ransomware detection.

Zhang et al. [145] present an approach to ransomware detection and defense using API sequences. The proposed system, REDDS, focuses on early detection by dynamically collecting API sequences during the pre-encryption stage. These sequences are converted into feature vectors using the n-gram model and TF-IDF algorithm. To enhance accuracy, they employ data augmentation, achieving up to 99.32% accuracy with RF. Additionally, the system uses an ontology-based method to map malicious APIs to security knowledge bases.

Ayub et al. [146] introduce RWArmor, a novel approach for the early detection of ransomware. This method combines static and dynamic analysis to enhance detection accuracy while reducing reliance on behavioral event logs. RWArmor employs a binary classification approach and uses PCA (Principal Component Analysis) for dimensionality reduction. The study utilized a dataset, analyzed via the Any Run sandbox platform. The RF model outperformed others, achieving 97.67% accuracy within 120 seconds of execution.

Von der Assen et al. [135] present a study on ransomware detection and mitigation within Linux-based systems through their proposed framework, GuardFS, at the file system level. GuardFS monitors system calls and analyzes features such as APIs, PE header files, and network traffic. The framework employs RF, achieving an accuracy rate close to 100%. The study tests the system against various ransomware families using both virtual machines and Raspberry Pi 4B.

Warren et al. [138] propose the FeSAD framework for ransomware detection by utilizing API calls. The framework includes the FeSAD Layer for feature selection, the Drift Calibration Layer, and the Drift Decision Layer. Their model achieves 95.60% accuracy with RF on a dataset of 720 ransomware and 2,000 benign samples.

Lee et al. [147] propose a methodology for detecting ransomware-infected files that utilizes ML techniques to address challenges posed by encoding algorithms. Their study focuses on file-level features, including file entropy and metadata such as file type, size, and creation dates. The approach, which involves a static analysis of these features, achieves high detection accuracy, approaching 100% with RF.

Ciaramella et al. [139] propose a ransomware detection approach by converting executable files into opcodes, which are then transformed into RGB images for analysis. They employ a static analysis method that leverages DL techniques to classify files as ransomware, generic malware, or legitimate software. Notably, the study assesses several models, with VGG-16 achieving the highest accuracy of 96.9%.

Rbah et al. [148] introduce a system for ransomware detection and prevention within Internet of Medical Things environments. The framework operates at the process level, utilizing lightweight DL algorithms. Specifically, the approach involves dynamic analysis of ransomware, focusing on file metadata. Furthermore, the system leverages DL models, including DL networks, achieving near 100% accuracy with LSTM.

TABLE IV
COMPARISON OF RANSOMWARE DETECTION RESEARCH

Research	Dataset	Analysis Technique	Features	Algorithms	Best Accuracy
Zakaria et al. (2024)	ElderRan(RISS)	Dynamic	APIs, Registration key, File Info., Embedded string	NB, KNN, SVM, RF, J48	93.8% (SVM)
Alqahtani et al. (2024)	Virusshare, Informer	Dynamic	PE headers	SVM, LR, RF, DBN, LSTM	93% (SVM)
Vaisakhkrishnan et al.(2024)	NA	Dynamic	Network Traffic	CNN, AE, LSTM, TN	97% (LSTM)
Rahman et al. (2024)	NA	Dynamic	Behavior Based: PFC, DLL, APIs, Signature-Based: NGS, HV	XGB, SVM, DT, RF	99% (DT)
Radhakrishna et al. (2024)	CICAndMal2017	Static	Network Traffic	RF, XGB, KNN, LR, EL	100% (XGB)
Zhang et al. (2023)	NA	Dynamic	APIs	SVM, NB, KNN, RF, MLP	99.32% (RF)

Research	Dataset	Analysis Technique	Features	Algorithms	Best Accuracy
Ayub et al. (2024)	SOREL-20M	Dynamic	APIs, PE Headers, File Activities,DLL	SVM, DT, RF, AdaBoost, GBC	97.67% (RF)
Von Der Assen et al. (2024)	NA	Dynamic	APIs, PE Headers, Opcode, I/O request, Syscalls, Assembly codes, Registry activities, Host logs, DLL, File activities, Network traffic	RF, LR, IF	100% (RF)
Warren et al. (2023)	Virusshare	Hybrid	APIs	RF, LR, J48, SVM, BN, GTB, MLP, DNN	95.6% (RF)
Lee et al. (2024)	GovDocs1	Static	File entropy, File metadata	KNN, LR, DT, RF, GB, SVM, MLP	100% (RF)
Ciaramella et al. (2023)	Virustotal	Static	Opcodes of executable files, converted into RGB images for analysis	CNN, LeNet, AlexNet, VGG16	96.9% (VGG-16)
Rbah et al. (2024)	Ransomware Detection Dataset	Dynamic	File, Linker, Debug, Stack, Resource and Security Metadatas	DNN, LSTM, BiLSTM	100% (LSTM)

As presented in Table IV, the majority of the analyzed studies prefer the dynamic analysis method. The datasets used in the studies are quite diverse, with APIs, PE Headers and file activity being the most prominent features. In addition, many machine learning and deep learning algorithms have been applied, with the performance of supervised learning algorithms being particularly remarkable.

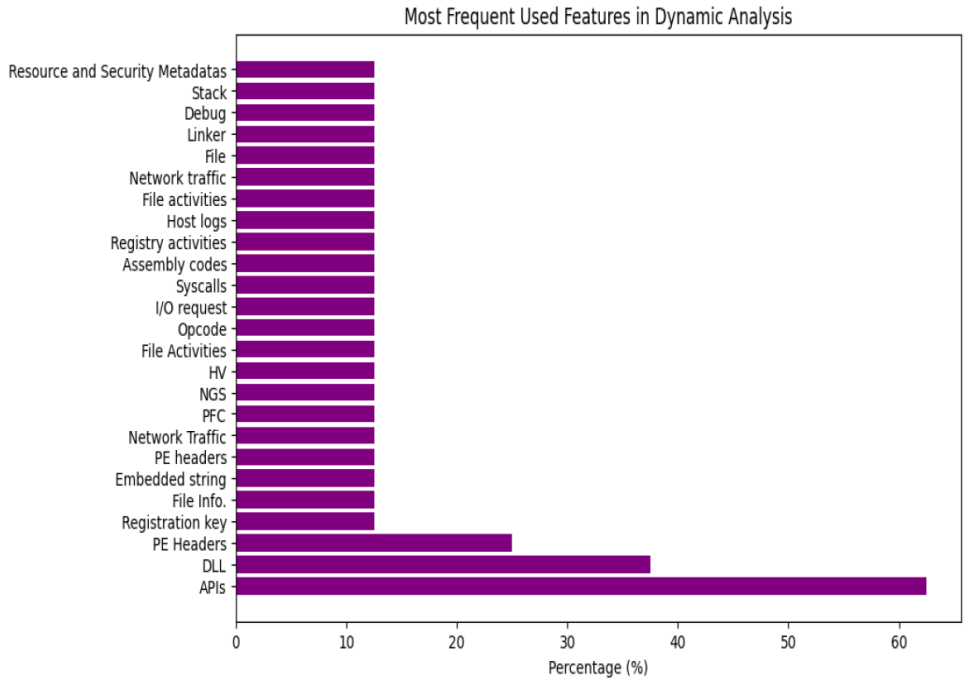


Figure 3: Most Frequent Used Features in Dynamic Analysis

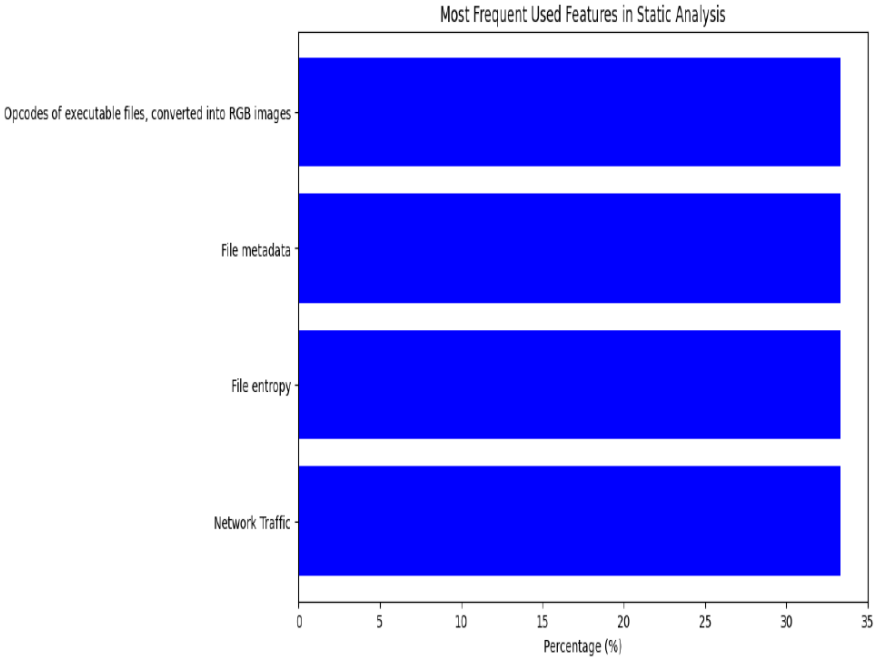


Figure 4: Most Frequent Used Features in Static Analysis

4.6. Summary

Ransomware poses a significant threat to both organizations and individuals, characterized by a higher probability of resulting in ransom payments compared to other cyber-attacks. Although significant progress has been made in ransomware detection, developing effective detection mechanisms remains a challenge. This paper argues that a deep understanding of malware

behavior is crucial for developing detection techniques. By highlighting current limitations and the need for improved methodologies, this research aims to guide future work to advance the effectiveness of ransomware detection.

Our analysis demonstrates that the RF method outperform other ML-DL algorithms in terms of ransomware detection efficacy, as illustrated in Figure 5. RF method achieving accuracy rates exceeding 99\% and the SVM method reaching approximately 93%, which is particularly effective in the early detection stages Moreover, LSTM networks have shown superior performance in DL approaches, especially for sequential or temporal data in complex environments such as IoT. In general, dynamic analysis techniques have proven to be more effective in early-stage ransomware detection, often outperforming static analysis. API calls are the most commonly used feature for detection across various datasets, followed by PE headers and network flow.

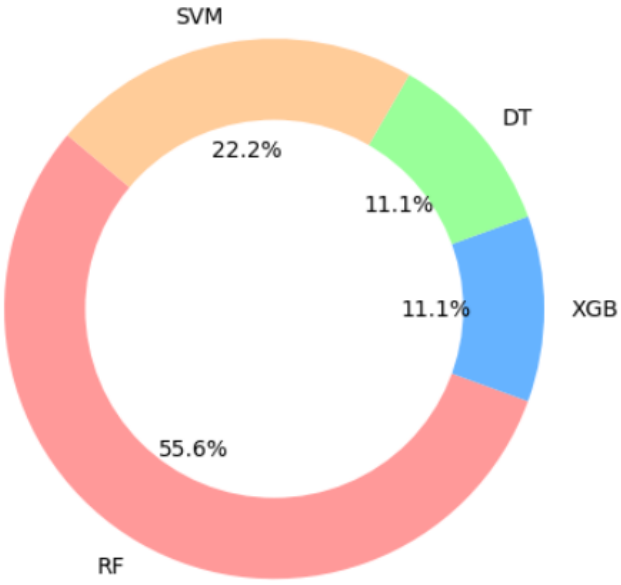


Fig. 5. Top-Performing Algorithms

An efficient ransomware detection application should be able to detect and stop the operations of ransomware before the initiation of the mass encryption process. To achieve this goal, the runtime of the ransomware detection services is crucial. Furthermore, for ML-based approaches, data pre-processing time should be included in the runtime. During our research, we were unable to obtain statistical data about the runtime of the mentioned studies. Consequently, localizing ransomware runtime detection services and the amount of data that could be encrypted in this time period is highlighted as a potential research area for future work.

This work will be presented in the 17th International Conference on Security of Information and Networks, indexed in Scopus and IEEE Xplore, and will be presented in December 2024.

5. Network Analysis

Network security is the backbone of modern organizational defence, designed to ensure the confidentiality, integrity, and availability of digital assets. In the face of increasingly sophisticated

threats such as ransomware and phishing attacks, robust network security measures have become essential. Cybercriminals continually refine their strategies to exploit vulnerabilities, making it imperative for organizations to adopt proactive, comprehensive approaches to mitigate risks [149].

Traditional network security methods primarily relied on perimeter defences such as firewalls and antivirus programs to guard against unauthorized access. While these tools were sufficient in isolated networks, the proliferation of cloud computing, mobile devices, and remote work environments has rendered these strategies inadequate [150]. Organizations must now incorporate more advanced techniques, such as endpoint protection, behavioural analytics, and network segmentation, to address modern attack vectors effectively [151].

One significant innovation in network security is the adoption of Artificial Intelligence (AI) and Machine Learning (ML). These technologies enable systems to analyse vast amounts of network data in real-time, identifying potential threats before they cause damage [152]. For instance, unsupervised learning techniques are particularly effective in detecting anomalies that deviate from established network behaviour, making them invaluable in identifying zero-day attacks and advanced persistent threats (APTs) [153].

Zero Trust Architecture (ZTA) represents a paradigm shift in network security, based on the principle of "never trust, always verify." Unlike traditional models that rely on perimeter-based trust, ZTA requires continuous verification of user identity, device security posture, and access permissions [154]. This model addresses vulnerabilities associated with lateral movement within networks and insider threats, ensuring that even authenticated users are monitored consistently [155].

Multi-factor authentication (MFA) and endpoint detection and response (EDR) technologies further complement ZTA by strengthening access controls and enabling real-time threat detection. For example, MFA provides an additional layer of verification, reducing the likelihood of unauthorized access even if credentials are compromised. EDR solutions continuously monitor endpoint activities, identifying suspicious behaviour indicative of a potential breach [151].

Despite these advancements, human error remains a leading cause of cybersecurity incidents. Phishing schemes, often the entry point for ransomware attacks, exploit gaps in user awareness [156]. To address this, organizations must invest in comprehensive cybersecurity training programs. Simulated phishing exercises, gamified training modules, and regular security updates can significantly reduce susceptibility to these attacks [157]. Behavioural analytics plays a crucial role in modern network security, enabling systems to identify patterns indicative of malicious intent. These tools analyse user activities, network traffic, and application behaviour to detect deviations from baseline norms [158]. When integrated with Security Information and Event Management (SIEM) systems, behavioural analytics provides a comprehensive view of potential threats across an organization's network [159].

The increasing use of encryption in network communications poses new challenges for threat detection. While encryption is vital for protecting data in transit, it also obscures malicious activities from traditional monitoring tools [160]. Advanced solutions such as Deep Packet Inspection (DPI) and ML-driven traffic analysis are being developed to address this issue, enabling security systems to identify threats within encrypted traffic without compromising privacy [161].

Emerging technologies, such as quantum cryptography and blockchain, are expected to further enhance network security in the coming years. Quantum cryptography, for example, promises to provide unbreakable encryption, while blockchain technologies can ensure data integrity and secure decentralized systems [162]. These advancements will be critical in addressing vulnerabilities associated with the Internet of Things (IoT) and edge computing environments. As IoT devices and edge computing become more widespread, securing decentralized networks will require innovative solutions. Endpoint monitoring and anomaly detection tools will need to adapt to the dynamic behaviours of IoT ecosystems, ensuring consistent protection across all connected devices [163]. Collaborative threat intelligence sharing among industry, academia, and governments will also play a vital role in addressing these challenges.

Looking forward, organizations must adopt an approach to network security that combines advanced technologies with strong policies and user education. By integrating tools such as ZTA, behavioural analytics, and AI-driven threat detection with regular training and awareness programs, businesses can build resilient networks capable of withstanding evolving cyber threats. This multifaceted approach will ensure that network security remains robust in the face of an increasingly complex threat landscape.

5.1. Network Intrusion Detection and Prevention Systems

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are foundational components of modern network security architectures. IDS monitors network traffic to detect and alert on suspicious activities, while IPS extends this functionality by actively blocking malicious traffic in real time [164]. Together, these systems provide comprehensive protection against a wide range of cyber threats, including phishing, ransomware, and advanced persistent threats (APTs) [165].

IDS and IPS primarily use two approaches for threat detection: signature-based detection and anomaly-based detection. Signature-based methods rely on predefined attack patterns to identify known threats, making them highly effective for common cyberattacks [166]. On the other hand, anomaly-based detection examines deviations from normal traffic patterns, enabling the identification of zero-day attacks and previously unseen threats [167]. Hybrid systems that combine both methods are increasingly adopted to ensure more accurate and reliable threat detection [168].

Machine learning (ML) has revolutionized IDS and IPS, providing advanced capabilities to detect sophisticated threats. Supervised learning algorithms such as Random Forests (RF) and Support Vector Machines (SVM) are used to classify traffic as malicious or benign based on labelled datasets [169]. These models excel in identifying known attack vectors but require extensive training data to achieve high accuracy. To address this limitation, unsupervised learning techniques such as clustering algorithms and autoencoders are employed to identify anomalies without prior knowledge of attack patterns [170]. Deep learning further enhances IDS and IPS performance by enabling systems to analyse large and complex datasets. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks are particularly effective for intrusion detection tasks. CNNs focus on extracting spatial features from network traffic, while LSTMs excel in modelling temporal dependencies, making them ideal for detecting APTs and other multi-stage attacks [171]. These models can process encrypted traffic more effectively, addressing one of the major challenges in modern network security.

Despite their capabilities, IDS and IPS face challenges in balancing precision and scalability. High false positive rates, where legitimate traffic is flagged as malicious, can overwhelm security teams and reduce operational efficiency [172]. Conversely, false negatives can leave organizations vulnerable to undetected threats. Researchers are exploring ensemble learning techniques, combining multiple models to enhance accuracy and reduce these issues [173].

The integration of IDS and IPS with Security Information and Event Management (SIEM) systems has further improved their effectiveness. SIEM platforms aggregate and analyse data from multiple sources, correlating security events to detect complex attack patterns [174]. This holistic approach provides a broader perspective on potential threats, enabling faster and more informed responses to security incidents [175].

Cloud and IoT environments also introduce challenges for IDS and IPS. These decentralized and dynamic ecosystems require solutions that can adapt to fluctuating traffic volumes and diverse device behaviours. Edge computing-based IDS/IPS systems are emerging as a promising solution, offering localized detection and response capabilities while reducing latency [176]. This adaptation ensures that IDS and IPS remain effective in securing modern, distributed networks.

On the other hand, IDS and IPS must evolve to address emerging threats and technologies. Future systems are expected to incorporate AI-driven threat intelligence, predictive analytics, and decentralized architectures to improve their detection and prevention capabilities. By leveraging advancements in machine learning, cloud-native security tools, and real-time analytics, IDS and IPS will remain integral to safeguarding digital infrastructures against evolving cyber threats [169].

5.2. Advanced Firewalls and Threat Intelligence Integration

Advanced firewalls, commonly referred to as next-generation firewalls (NGFWs), have revolutionized traditional network security by integrating application awareness, intrusion prevention, and threat intelligence into a single system. Unlike traditional firewalls that primarily focus on packet filtering, NGFWs provide granular traffic inspection at the application layer, enabling them to detect and block sophisticated attacks such as ransomware and phishing campaigns [177]. Their ability to differentiate between legitimate and malicious application traffic makes NGFWs indispensable in modern cybersecurity strategies [178].

One of the defining features of NGFWs is their deep packet inspection (DPI) capability, which examines the contents of data packets in real time. DPI enables NGFWs to identify threats embedded within encrypted traffic, an area where traditional firewalls struggle [179]. This functionality is particularly crucial as more than 80% of internet traffic is now encrypted, presenting both a challenge and an opportunity for modern network defences [180]. Machine learning (ML) algorithms integrated into NGFWs further enhance DPI by identifying patterns indicative of malicious behaviour [181].

Threat intelligence integration is another transformative aspect of NGFWs. By leveraging data from global threat intelligence feeds, NGFWs can identify emerging attack patterns and dynamically update their rules to block known malicious IPs, URLs, and domains [182]. This real-time adaptability ensures that organizations are protected against zero-day vulnerabilities and fast-

evolving threats. Furthermore, the automation of threat intelligence processing reduces the response time for mitigating security incidents.

Despite their advanced capabilities, NGFWs face challenges in scaling to meet the demands of high-speed, high-volume networks. Deep packet inspection and real-time analytics require substantial computational resources, which can impact performance and introduce latency in network operations [183]. Hardware acceleration techniques and optimized algorithms are being developed to address these limitations, allowing NGFWs to maintain high throughput without compromising security [176].

The adoption of NGFWs is also expanding into cloud and hybrid environments. Cloud-native NGFW solutions offer consistent security policies across on-premises and cloud infrastructures, enabling seamless protection of distributed applications and data [179]. These firewalls integrate with cloud service provider ecosystems, such as Amazon Web Services (AWS) and Microsoft Azure, to provide visibility and control over cloud workloads. This flexibility is critical for organizations embracing multi-cloud strategies [178].

Behavioural analytics is increasingly being incorporated into NGFWs to detect insider threats and lateral movement within networks. By analysing user and entity behaviours, NGFWs can identify deviations from normal activity patterns, such as unusual login attempts or data access anomalies [184]. These insights complement traditional threat detection methods, providing a more comprehensive view of potential security risks.

Looking forward, NGFWs are expected to leverage artificial intelligence (AI) and predictive analytics to anticipate attack trends and proactively implement countermeasures. AI-driven NGFWs will not only detect and respond to existing threats but also predict emerging vulnerabilities by analysing historical and real-time data [177]. These advancements will be particularly valuable as the threat landscape continues to evolve, requiring more adaptive and intelligent defence mechanisms.

The integration of NGFWs with Security Information and Event Management (SIEM) systems and threat hunting platforms will further enhance their effectiveness. This collaboration provides organizations with a unified view of their security posture, enabling faster and more coordinated responses to complex attack scenarios. As cyber threats grow in sophistication, NGFWs will remain a cornerstone of modern network security, ensuring robust and scalable protection for both traditional and cloud-native environments [182].

5.3. Machine Learning Applications in Network Security

Machine learning (ML) has transformed the landscape of network security, offering adaptive and intelligent solutions for detecting and mitigating cyber threats. Unlike traditional rule-based systems, ML models can analyse large datasets and learn patterns that signify potential malicious behaviour. This capability is particularly valuable in addressing advanced threats such as zero-day exploits and advanced persistent threats (APTs), where conventional methods often fail to identify novel attack vectors [169].

Supervised learning algorithms, such as Support Vector Machines (SVMs) and Random Forests (RF), are among the most widely used ML models in intrusion detection systems (IDS). These algorithms classify network traffic as benign or malicious by learning from labelled datasets of

previous attacks [166]. While these models are highly effective in detecting known threats, their reliance on labelled data limits their capacity to identify entirely new attack patterns [167]. This limitation has driven interest in unsupervised and semi-supervised learning methods.

Unsupervised learning techniques, such as clustering and anomaly detection, address the challenge of detecting unknown threats without relying on labelled data. Models like K-means clustering and autoencoders can identify deviations from normal network behaviour, flagging potential threats for further analysis [184]. Autoencoders, for instance, reconstruct input data and measure reconstruction errors to detect anomalies. These methods are particularly valuable for identifying insider threats and lateral movement within networks [170].

Deep learning has further advanced ML applications in network security. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks excel at analysing high-dimensional and sequential data, respectively. CNNs are effective for tasks like malware classification and packet inspection, while LSTMs are well-suited for detecting temporal anomalies in network traffic [171]. The integration of these models into real-time systems has significantly enhanced the detection of complex multi-stage attacks.

Another promising development is the application of reinforcement learning (RL) in network security. RL algorithms can autonomously learn optimal defence strategies by interacting with the network environment. For instance, RL-based systems can dynamically adjust firewall rules or allocate resources to mitigate ongoing attacks, providing a proactive and adaptive layer of security [185]. These systems are particularly effective in managing distributed denial-of-service (DDoS) attacks.

Despite their potential, ML-based security systems face several challenges. One significant issue is the susceptibility to adversarial attacks, where attackers manipulate input data to deceive ML models. For example, slight modifications to malicious traffic can cause an ML model to misclassify it as benign. Addressing these vulnerabilities requires the development of robust training methods and adversarial-resistant architectures [166].

The computational complexity of ML models also poses a challenge, particularly in large-scale networks with high traffic volumes. Training deep learning models requires significant computational resources and deploying them in real-time systems can introduce latency. Federated learning and edge computing are emerging as solutions to these challenges, enabling decentralized model training and localized decision-making without compromising performance [175].

As cyber threats continue to evolve, ML will remain a remarkable network security innovation. Organizations must adopt a multi-layered approach that combines ML-driven threat detection, robust adversarial defences, and real-time analytics to protect their digital infrastructures effectively. By leveraging advancements in AI and ML, businesses can stay ahead of cybercriminals and build resilient systems capable of withstanding future threats [176].

6. Conclusion

In this study, we describe the current state of the technologies and innovations we will use in the VESTA project. We examined the versatile techniques and technologies used in the detection, prevention and analysis of phishing attacks, ransomware and network attacks. Our work throughout

the project will be guided by a detailed examination of the various approaches that have proven effective in mitigating their threats.

Under the Phishing Detection and Prevention section, techniques such as email filtering, URL analysis and behavioral analysis, which play a critical role in identifying phishing attempts, are examined. Behavioral analysis, which encompasses email, website, and user interaction behavior, is emerging as a particularly effective method for detecting sophisticated phishing schemes. Content analysis, both textual and image-based, provides valuable insights into how phishing attacks can be recognized based on content characteristics. Machine learning and artificial intelligence, anomaly detection, user behavior analytics and endpoint monitoring are increasingly being used to improve the accuracy and speed of phishing detection.

In Advanced Email Security Measures, the role of advanced email security protocols is examined. These protocols can significantly improve the ability of systems such as IDS, IPS and SIEM to detect and prevent email-based attacks in real time.

In Ransomware Detection and Behavioral Analysis, the focus is on understanding the lifecycle of ransomware, from infection to encryption, which is critical to developing effective detection methods. By analyzing the behavior of ransomware, we can identify attack patterns and react before damage is done. Furthermore, both public and unpublished datasets, the number of ransoms they contain, the diversity of ransom families and the features they contain, have proven to be important for training machine learning models to predict and counter ransomware attacks.

Network Analysis has an important place in phishing and ransomware detection. Network intrusion detection and prevention, advanced firewalls and machine learning applications in network security provide a comprehensive defense mechanism against external and internal network threats. Integrating this mechanism with threat intelligence has become indispensable against evolving threats.

As a result, cyber threats such as phishing and ransomware continue to evolve in sophistication. By integrating machine learning, behavioral analytics and advanced security protocols, it may be possible to stay ahead of attackers. Ongoing research and technological advancements promise even more effective solutions to combat phishing, ransomware and network attacks, and the VESTA project will contribute to these solutions, ensuring a more secure digital environment for individuals and organizations.

References

[1]. Matt Kapko (2024) “Ransomware attacks surge despite international enforcement effort”, URL: https://www.cybersecuritydive.com/news/ransomware-surges-desite-global-effort/728534/

[2]. Desolda G, Ferro LS, Marrella A et al (2022) Human factors in phishing attacks: a systematic literature review. ACM Comput Surv. https://doi.org/10.1145/3469886

[3]. Neupane S, Fernandez IA, Mittal S, Rahimi S (2023), "Impacts and risk of generative AI technology on cyber defense", https://arxiv.org/abs/2306.13033

[4]. S. Jamal, H. Wimmer, and I. H. Sarker, “An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach,” Security and Privacy, p. e402, 2024. [Online]. Available: https://doi.org/10.1002/spy2.402

[5]. Aleroud, A., & Zhou, L. (2017). “Phishing environments, techniques, and countermeasures: A survey”, Computers & Security, 68, 160-196.

[6]. N. Abdelhamid, F. Thabtah and H. Abdel-jaber, "Phishing detection: A recent intelligent machine learning comparison based on models’ content and features," 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 2017, pp. 72-77, doi: 10.1109/ISI.2017.8004877.

[7]. J. Tanimu and S. Shiaeles, "Phishing Detection Using Machine Learning Algorithm," 2022 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 2022, pp. 317-322, doi: 10.1109/CSR54599.2022.9850316.

[8]. Antonidoss A., Arivukarasi M., (September 2019),"Artificial Intelligence Techniques for Phishing Detection". International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-11.

[9]. Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2021). Phishing email detection using natural language processing techniques: a literature survey. Procedia Computer Science, 189, 19-28.

[10]. Stone, A. (2007). Natural-language processing for intrusion detection. Computer, 40(12), 103-105.

[11]. S. Salloum, T. Gaber, S. Vadera and K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," in IEEE Access, vol. 10, pp. 65703-65727, 2022, doi: 10.1109/ACCESS.2022.3183083.

[12]. Blanzieri, EnricoBryl, Anton, "A survey of learning-based techniques of email spam filtering," Artificial Intelligence Review, Springer Netherlands, vol. 29, no.1, pp. 63-92, 2008.

[13]. Calzarossa, M. C., Giudici, P., & Zieni, R. (2023, July). Explainable Machine Learning for Bag of Words-Based Phishing Detection. In World Conference on Explainable Artificial Intelligence (pp. 531-543). Cham: Springer Nature Switzerland.

[14]. Chapelle, O., "Training a support vector machine in the primal," Neural Computation, vol. 19, no.5,pp. 1155-1178, 2007.

[15]. R. Dazeley, et al., "Consensus Clustering and Supervised Classi cation for Pro ling Phishing Emails in Internet Commerce Security," in Knowledge Management and Acquisition for Smart Systems and Services, Berlin Heidelberg, pp. 235-246, 2010.

[16]. Yearwood, JMammadov, MBanerjee, A, "Profiling Phishing Emails Based on Hyperlink Information," in 2010 International Conference on Advances in Social Networks Analysis and Mining, Odense,,IEEE, Denmark 2010, pp. 120-127.

[17]. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[18]. Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D., & Rodríguez-Galán, G. (2023). A phishing-attack-detection model using natural language processing and deep learning. Applied Sciences, 13(9), 5275.

[19]. S. Salloum, T. Gaber, S. Vadera and K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," in IEEE Access, vol. 10, pp. 65703-65727, 2022, doi: 10.1109/ACCESS.2022.3183083.

[20]. Bhowmick A., Hazarika S.M., (3 June 2016) , "Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends". arXiv:1606.01042v1

[21]. M. Abutaha, M. Ababneh, K. Mahmoud and S. A. -H. Baddar, "URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis," 2021 12th International Conference on Information and Communication Systems (ICICS), Valencia, Spain, 2021, pp. 147-152, doi: 10.1109/ICICS52457.2021.9464539.

[22]. J. Kang and D. Lee, "Advanced White List Approach for Preventing Access to Phishing Sites", 2007 International Conference on Convergence Information Technology (ICCIT 2007), pp. 491-496, 2007.

[23]. P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 1-5, doi: 10.1109/INFCOM.2010.5462216.

[24]. Li, Y., Xiong, K., & Li, X. (2018). Applying Machine Learning Techniques to Understand User Behaviors When Phishing Attacks Occur. ICST Transactions on Security and Safety, 0(0), 162809. https://doi.org/10.4108/eai.13-7-2018.162809.

[25]. Tamal, M. A., Islam, M. K., Bhuiyan, T., Sattar, A., & Nayem Uddin Prince. (2024). Unveiling suspicious phishing attacks: enhancing detection with an optimal feature vectorization algorithm and supervised machine learning. Frontiers in Computer Science, 6. https://doi.org/10.3389/fcomp.2024.1428013

[26]. Thakur, K., Ali, M. L., Obaidat, M. A., & Kamruzzaman, A. (2023). A Systematic Review on Deep-Learning-Based Phishing Email Detection. Electronics, 12(21), 4545. https://doi.org/10.3390/electronics12214545

[27]. Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., & Shukla, S. (2022). Applications of deep learning for phishing detection: a systematic literature review. Knowledge and Information Systems, 64(6), 1457–1500. https://doi.org/10.1007/s10115-022-01672-x

[28]. Ravindra, Salvi & Sanjay, Shah & Gulzar, Shaikh & Pallavi, Khodke. (2021). Phishing Website Detection Based on URL. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 589-594. 10.32628/CSEIT2173124.

[29]. Çolhak, Furkan, et al. "Phishing Website Detection through Multi-Model Analysis of HTML Content." arXiv preprint arXiv:2401.04820 (2024).

[30]. Opara, Chidimma, Yingke Chen, and Bo Wei. "Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics." Expert Systems with Applications 236 (2024): 121183.

[31]. Safi, Asadullah, and Satwinder Singh. "A systematic literature review on phishing website detection techniques." Journal of King Saud University-Computer and Information Sciences 35.2 (2023): 590-611.)

[32]. Tang, Lizhen, and Qusay H. Mahmoud. "A survey of machine learning-based solutions for phishing website detection." Machine Learning and Knowledge Extraction 3.3 (2021): 672-694.)

[33]. Rajeswary, C., and M. Thirumaran. "A comprehensive survey of automated website phishing detection techniques: a perspective of artificial intelligence and human behaviors." 2023

International conference on sustainable computing and data communication systems (ICSCDS). IEE,E, 2023.

[34]. Kulkarni, Aditya, Vivek Balachandran, and Tamal Das. "Phishing Webpage Detection: Unveiling the Threat Landscape and Investigating Detection Techniques." IEEE Communications Surveys & Tutorials (2024).

[35]. Chawla, Minal, and Siddarth Singh Chouhan. "A survey of phishing attack techniques." International Journal of Computer Applications 93.3 (2014).

[36]. Li, Yi, Kaiqi Xiong, and Xiangyang Li. "Applying machine learning techniques to understand user behaviors when phishing attacks occur." EAI Endorsed Transactions on Security and Safety 6.21 (2019): e3-e3.

[37]. Aldakheel, Eman Abdullah, et al. "A Deep learning-based innovative technique for phishing detection in modern security with uniform resource locators." Sensors 23.9 (2023): 4403.

[38]. Pinto, L., Brito, C., Marinho, V., & Pinto, P. (2022). Assessing the relevance of cybersecurity training and policies to prevent and mitigate the impact of phishing attacks. Journal of Internet Services and Information Security, 12(4), 23-38.

[39]. Kumaraguru, P., Cranshaw, J., Acquisti, A., Cranor, L., Hong, J., Blair, M. A., & Pham, T. (2009, July). School of phish: a real-world evaluation of anti-phishing training. In Proceedings of the 5th Symposium on Usable Privacy and Security (pp. 1-12).

[40]. S. Baki and R. M. Verma, "Sixteen Years of Phishing User Studies: What Have We Learned?," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1200-1212, 1 March-April 2023, doi: 10.1109/TDSC.2022.3151103.

[41]. Butavicius, M., Taib, R., & Han, S. J. (2022). Why people keep falling for phishing scams: The effects of time pressure and deception cues on the detection of phishing emails. Computers & Security, 123, 102937.

[42]. Dukarm, C., Dill, R., & Reith, M. (2019, July). Improving phishing awareness in the United States Department of Defense. In Proceedings of the 18th European Conference on Cyber Warfare and Security (2019), Nr Reading: Acad Conferences Ltd (pp. 172-177).

[43]. Jang, C., Lee, O., Mun, C. H. A. N. G. B. A. E., & Ha, H. (2022). An Analysis of Phishing Cases Using Text Mining. Journal of Theoretical and Applied Information Technology, 100(22), 6758-6773.

[44]. L'Huillier, G., Hevia, A., Weber, R., & Rios, S. (2010, May). Latent semantic analysis and keyword extraction for phishing classification. In 2010 IEEE international conference on intelligence and security informatics (pp. 129-131). IEEE.

[45]. Barraclough, P. A., Hossain, M. A., Tahir, M. A., Sexton, G., & Aslam, N. (2013). Intelligent phishing detection and protection scheme for online transactions. Expert systems with applications, 40(11), 4697-4706.

[46]. Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. Expert systems with applications, 53, 231-242.

[47]. Da Silva, C. M. R., Fernandes, B. J. T., Feitosa, E. L., & Garcia, V. C. (2022). Piracema. io: A rules-based tree model for phishing prediction. Expert Systems with Applications, 191, 116239.

[48]. Bhowmick A., Hazarika S.M., (3 June 2016) , “Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends”. arXiv:1606.01042v1

[49]. Sheneamer A., (2021), “Comparison of Deep and Traditional Learning Methods for Email Spam Filtering”. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 12, No.1

[50]. Altwaijry, N., Al-Turaiki, I., Alotaibi, R., & Alakeel, F. (2024). Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models. Sensors, 24(7), 2077.
[51]. Nahmias, D., Engelberg, G., Klein, D., & Shabtai, A. (2024). Prompted contextual vectors for spear-phishing detection. arXiv preprint arXiv:2402.08309.
[52]. A. S. Bozkir and M. Aydos, "Logosense: A companion hog based logo detection scheme for phishing web page and e-mail brand recognition", Computers & Security, vol. 95, pp. 101855, 2020.
[53]. Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, et al., "Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages", 30th {USENIX} Security Symposium ({USENIX} Security 21), 2021.
[54]. S. -C. Lin, P. -C. Wl, H. -Y. Chen, T. Morikawa, T. Takahashi and T. -N. Lin, "SenseInput: An Image-Based Sensitive Input Detection Scheme for Phishing Website Detection," ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, pp. 4180-4186, doi: 10.1109/ICC45855.2022.9838653.
[55]. Trinh, N. B., Phan, T. D., & Pham, V. H. (2022, December). Leveraging deep learning image classifiers for visual similarity-based phishing website detection. In Proceedings of the 11th International Symposium on Information and Communication Technology (pp. 134-141).
[56]. Jain, A. K., & Gupta, B. B. (2017). Phishing detection: analysis of visual similarity based approaches. Security and Communication Networks, 2017(1), 5421046.
[57]. Ford, J., & Berry, H. S. Feasibility of Machine Learning-Enhanced Detection for QR Code Images in Email-based Threats.
[58]. K. Tian, S. T. Jan, H. Hu, D. Yao and G. Wang, "Needle in a haystack: Tracking down elite phishing domains in the wild", Proceedings of the Internet Measurement Conference 2018, pp. 429-442, 2018.
[59]. W. Zhang, Q. Jiang, L. Chen and C. Li, "Two-stage elm for phishing web pages detection using hybrid features", World Wide Web, vol. 20, no. 4, pp. 797-813, 2017.
[60]. Kumar et al., (3 October 2018). “DeepImageSpam: Deep Learning based Image Spam Detection”. arXiv:1810.03977v1
[61]. W. Zhang, Q. Jiang, L. Chen and C. Li, "Two-stage elm for phishing web pages detection using hybrid features", World Wide Web, vol. 20, no. 4, pp. 797-813, 2017.
[62]. Abuhammed H.Z., Abuzaid A.A., (March 2022), “Image Spam Detection Using ML and DL Techniques”. International Journal of Advances in Soft Computing and its Applications.
[63]. Klimt, B., & Yang, Y. (2004, September). The enron corpus: A new dataset for email classification research. In European conference on machine learning (pp. 217-226). Berlin, Heidelberg: Springer Berlin Heidelberg.
[64]. D. Nahmias, ”Prompted Contextual Vectors for Spear Phishing Detection” github.com https://github.com/nahmiasd/Prompted-Contextual-Vectors-for-Spear-Phishing-Detection (accessed November 21, 2024)
[65]. Nazario. Phishing Corpus 2015. 2015. Available online: https://academictorrents.com/details/a77cda9a9d89a60dbdfbe581adf6e2df9197995a (accessed on 21 Nov 2024).
[66]. Naser Alam, Amith Khandakar, “Phishing Email Dataset,” Kaggle.com, 2024. https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset/data?select=SpamAssasin.csv (accessed Nov. 21, 2024).
[67]. D. Radev, CLAIR collection of fraud email ACL Data and Code Repository, 2008, [online] Available: https://aclweb.org/aclwiki .

[68]. Anonymous, “Phishing Email Curated Datasets,” Zenodo (CERN European Organization for Nuclear Research), Sep. 2023, doi: https://doi.org/10.5281/zenodo.8339690 .
[69]. “Phishing Email Dataset,” www.kaggle.com. https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset
[70]. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. Information retrieval, 6, 49-73.
[71]. Ealvaradob/phishing-dataset · Datasets at Hugging Face — huggingface.co, https://huggingface.co/datasets/ealvaradob/phishing-dataset , [Accessed 21-11-2024].
[72]. Tamal, Maruf (2023), “Phishing Detection Dataset”, Mendeley Data, V1, doi: 10.17632/6tm2d6sz7p.1
[73]. Bountakas, P., & Xenakis, C. (2023). Helphed: Hybrid ensemble learning phishing email detection. Journal of network and computer applications, 210, 103545.
[74]. Mohammad, R.; McCluskey, T.L.; Thabtah, F. UCI Machine Learning Repository: Phishing Websites Data Set. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 21 November 2024).
[75]. Gallo, L., Gentile, D., Ruggiero, S., Botta, A., & Ventre, G. (2024). The human factor in phishing: Collecting and analyzing user behavior when reading emails. Computers & Security, 139, 103671.
[76]. Omar, A. R., Taie, S., & Shaheen, M. E. (2023). From Phishing Behavior Analysis and Feature Selection to Enhance Prediction Rate in Phishing Detection. International Journal of Advanced Computer Science and Applications, 14(5).
[77]. Kaushik, P., & Rathore, S. P. S. (2023, August). Deep Learning Multi-Agent Model for Phishing Cyber-attack Detection. In IJRITCC (Vol. 11, No. 9s, pp. 680-686).
[78]. M. Srivastava, D. Khandelwal and N. Swarup, "Email Spam Monitoring System," 2024 4th International Conference on Intelligent Technologies (CONIT), Bangalore, India, 2024, pp. 1-7, doi: 10.1109/CONIT61985.2024.10626718.
[79]. Andriu, A. V. (2023). Adaptive Phishing Detection: Harnessing the Power of Artificial Intelligence for Enhanced Email Security. Romanian Cyber Secur. J, 5(1), 3-9.
[80]. Bountakas, P., & Xenakis, C. (2023). Helphed: Hybrid ensemble learning phishing email detection. Journal of network and computer applications, 210, 103545.
[81]. D. L. Cook, V. K. Gurbani and M. Daniluk, "Phishwish: A stateless phishing filter using minimal rules Financial Cryptography and Data Security", Springer-Verlag, pp. 182-186, 2008.
[82]. N. Chou, R. Ledesma, Y. Teraguchi and J. C. Mitchell, "Client-side defense against web-based identity theft", NDSS. The Internet Society, 2004.
[83]. L. A. T. Nguyen, B. L. To, H. K. Nguyen and M. H. Nguyen, "Detecting phishing web sites: A heuristic URL-based approach", 2013 International Conference on Advanced Technologies for Communications (ATC 2013), pp. 597-602, 2013.
[84]. Cloudflare, "What are DMARC, DKIM, and SPF?". https://www.cloudflare.com/learning/email-security/dmarc-dkim-spf/
[85]. Postmark, “SPF record: Protect your domain reputation and email delivery”, 2016 (updated in 2024). https://postmarkapp.com/guides/spf
[86]. Postmark, “What Is DKIM? DomainKeys Identified Mail Explained”, 2017 (updated in 2024). https://postmarkapp.com/guides/dkim
[87]. Fortinet, “What Is DMARC? How Does DMARC Work?”. https://www.fortinet.com/resources/cyberglossary/dmarc

[88]. ICANN, “DNSSEC – What Is It and Why Is It Important?”. https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en
[89]. Cloudfare, “How Does DNSSEC Works?”. https://www.cloudflare.com/learning/dns/dnssec/how-dnssec-works/
[90]. Microsoft Learn, “Overview of file sharing using the SMB 3 protocol in Windows Server”, 2023. https://learn.microsoft.com/en-us/windows-server/storage/file-server/file-server-smb-overview
[91]. Microsoft Learn, “Remote Desktop Services (Remote Desktop Services) - Win32 apps”, 2020. https://learn.microsoft.com/en-us/windows/win32/termserv/terminal-services-portal
[92]. GitHub Repository, “rekall: Rekall Memory Forensic Framework”. https://github.com/google/rekall
[93]. GitHub Repository, “volatility: An advanced memory forensics framework”. https://github.com/volatilityfoundation/volatility
[94]. YARA, “YARA's documentation”. https://yara.readthedocs.io/en/latest/
[95]. GitHub, “Yara Rules”. https://github.com/Yara-Rules/rules
[96]. Microsoft Learn, “Why do I need Microsoft Defender for Office 365?”, 2024. https://learn.microsoft.com/en-us/defender-office-365/mdo-about
[97]. Microsoft Learn, “Exchange Online Protection (EOP) overview”, 2024. https://learn.microsoft.com/en-us/defender-office-365/eop-about
[98]. Google, “Gmail Email Security & Privacy Settings”. https://safety.google/gmail/
[99]. IT Security, “Darktrace Transforms Security Operations and Cyber Resilience”, 2024. https://itsecuritywire.com/news/darktrace-transforms-security-operations-and-improves-cyber-resilience-with-launch-of-darktrace-activeai-security-platform/
[100]. CISCO, “Cisco Live 2024: Cisco Announces New AI-powered Innovations and Investments to Help Customers Unlock a More Connected and Secure Future”, 2014. https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2024/m06/cisco-live-2024-cisco-announces-new-ai-powered-innovations-and-investments-to-help-customers-unlock-a-more-connected-and-secure-future.html

[101]. Elastic Security, “AI-driven SIEM Solution & Security Analytics”. https://www.elastic.co/security/siem
[102]. Fortinet, “What is Splunk? Key Benefits and Features of Splunk”. https://www.fortinet.com/resources/cyberglossary/what-is-splunk
[103]. Exabeam, “The AI-Driven Exabeam Security Operations Platform: Revolutionizing Threat Detection, Investigation, and Response”, 2024. https://www.exabeam.com/blog/infosec-trends/the-ai-driven-exabeam-security-operations-platform-revolutionizing-threat-detection-investigation-and-response/
[104]. Securonix, “2024 Gartner Critical Capabilities for Security Information and Event Management”. https://www.securonix.com/resources/2024-gartner-critical-capabilities-for-siem/
[105]. Securonix, “Securonix Unveils Next Wave of AI-Reinforced Capabilities With Launch of Cyber Data Fabric and Noise Canceling SIEM”, 2024. https://www.securonix.com/press_release/securonix-eon-next-wave-features-launch/
[106]. LogRhythm SIEM, “Advanced Intelligence Engine”. https://docs.logrhythm.com/lrsiem/7.13.0/advanced-intelligence-engine

[107]. KnowBe4, “Security Awareness Training Features”. https://www.knowbe4.com/products/security-awareness-training/features
[108]. Cofense, “PhishMe Solution Architecture”. https://cofense.com/phishme-solution-architecture
[109]. F. Aldauiji, O. Batarfi and M. Bayousef, "Utilizing Cyber Threat Hunting Techniques to Find Ransomware Attacks: A Survey of the State of the Art," in IEEE Access, vol. 10, pp. 61695-61706, 2022
[110]. O. Delgado-Mohatar, and J.M. Sierra-Cámara, and E. Anguiano, "Blockchain-based semi-autonomous ransomware", Future Generation Computer Systems, Volume 112, 2020, Pages 589-603.
[111]. Ani Petrosyan, "Businesses worldwide affected by ransomware 2018-2023", Mar 28, 2024 https://www.statista.com/statistics/204457/businesses-ransomware-attack-rate/
[112]. IBM Security, "Cost of a Data Breach Report 2024", 2024. https://www.ibm.com/reports/data-breach
[113]. S. Razaulla et al., "The Age of Ransomware: A Survey on the Evolution, Taxonomy, and Research Directions," IEEE Access, vol. 11, pp. 1234-1256, Apr. 2023.
[114]. J. Ispahany et al., "Ransomware Detection Using Machine Learning: A Review," IEEE Access, vol. 12, pp. 1234-1250, May 2024.
[115]. M. Cen et al., "Ransomware early detection: A survey," Computer Networks, vol. 239, p. 110138, Feb. 2024.
[116]. K. Begovic et al., "Cryptographic ransomware encryption detection: Survey," Computers & Security, vol. 132, p. 103349, Sep. 2023.
[117]. M. ur Rehman et al., "A Systematic Literature Review of Ransomware Detection Methods," ICOCI 2023, pp. 80-95, Jan. 2024.
[118]. "Mitre ATT\&CK for Enterprise", Visited: Aug 08, 2024. https://attack.mitre.org/versions/v15/matrices/enterprise/
[119]. I. Gulatas et al., "Malware threat on edge/fog computing environments," IEEE Access, vol. 11, pp. 33584-33606, 2023.
[120]. M. Nkongolo et al., "UGRansome1819: A Novel Dataset for Anomaly Detection and Zero-Day Threats," Information, vol. 12, no. 10, Art. no. 405, Sep. 2021.
[121]. D. Arp et al., "DREBIN: Effective and Explainable Detection of Android Malware," University of Göttingen, Germany and Siemens CERT, Munich, Germany. Available: https://www.sec.cs.tu-bs.de/pubs/2014-ndss.pdf .
[122]. M. M. Adnan et al., "Sine cosine reptile search algorithm with grid search SVM based ransomware detection," ICICACS 2024, Feb. 2024.
[123]. D. Sgandurra et al., "Automated Dynamic Analysis of Ransomware," arXiv preprint, Sep. 2016.
[124]. D. S. Keyes et al., "EntropLyzer: Android Malware Classification and Characterization," RDAAPS 2021, May 2021.
[125]. A. Rahali et al., "DIDroid: Android Malware Classification and Characterization," ICCNS2020, Nov. 2020.
[126]. A. H. Lashkari et al., "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets," ICCST 2018, Oct. 2018.
[127]. R. Harang and E. M. Rudd, "SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection," arXiv preprint, Dec. 2020. DOI: 10.48550/arXiv.2012.

[128]. "Ransomware Detection Dataset (2023)," Kaggle. Available: https://www.kaggle.com/datasets/amdj3dax/ransomware-detection-data-set
[129]. MLRD: Machine Learning Ransomware Detection," GitHub repository. Available: https://github.com/securycore/MLRD-Machine-Learning-Ransomware-Detection
[130]. M. Hirano et al., "RanSAP: An open dataset of ransomware storage access patterns," Forensic Sci. Int.: Digit. Investig., vol. 40, pp. 301314, Mar. 2022.
[131]. H. Zhang et al., "Early Ransomware Detection Through Kernel-Level Behavioral Analysis," IEEE Trans. Inf. Forensics Security, vol. 19, pp. 6113-6127, Jun. 2024. DOI: 10.1109/TIFS.2024.3410511.
[132]. A. Huertas Celdr´an et al., "Behavioral fingerprinting to detect ransomware in resource-constrained devices," Comput. Security, vol. 117,Art. no. 103510, Oct. 2023. DOI: 10.1016/j.cose.2023.103510
[133]. A. H. N. Almoqbil, "Anomaly detection for early ransomware and spyware warning in nuclear power plant systems based on FusionGuard," J. Comput. Security, vol. 23, pp. 2377-2394, Apr. 2024.
[134]. C. C. Moreira et al., "A comprehensive analysis combining structural features for detection of new ransomware families," J. Inf. Security Appl., vol. 81, Art. no. 103716, Mar. 2024. Available: https://doi.org/10.1016/j.jisa.2024.103716 .
[135]. J. von der Assen et al., "GuardFS: a File System for Integrated Detection and Mitigation of Linux-based Ransomware," arXiv preprint, arXiv:2401.17917, Jan. 2024. Available: https://doi.org/10.48550/arXiv.2401.17917 .
[136]. X. Deng et al., "Ransomware early detection using deep reinforcement learning on portable executable header," Cluster Comput., vol. 27, pp. 1867-1881, Jun. 2023.
[137]. C. B. N. and B. S. H., "Revolutionizing ransomware detection and criticality assessment: Multiclass hybrid ML and semantic similarity- based end2end solution," Multimedia Tools Appl., vol. 83, pp. 39135- 39168, Oct. 2023.
[138]. D. Warren Fernando and N. Komninos, "FeSAD ransomware detection framework with ML using adaptation to concept drift," Comput. Security, vol. 137, Art. no. 103629, Feb. 2024. Available: https://doi.org/10.1016/j.cose.2023.103629 .
[139]. G. Ciaramella et al., "Explainable Ransomware Detection with Deep Learning Techniques," J. Comput. Virol. Hacking Tech., vol. 20, pp. 317–330, 2024.
[140]. W. Z. A. Zakaria et al., "Early detection of Windows cryptographic ransomware," JARSE, vol. 39, no. 2, Feb. 2024. DOI:10.1016/j.jarse.2024.01.006
[141]. A. Alqahtani and F. T. Sheldon, "eMIFS: A Normalized Hyperbolic Ransomware Deterrence Model," Sensors, vol. 24, no. 6, p. 1728, Mar. 2024. DOI: 10.3390/s24061728.
[142]. K. Vaisakhkrishnan et al., "Deep Learning for Attack Detection in Medical IoT," Procedia Comput. Sci., vol. 235, pp. 2498-2507, 2024. DOI: 10.1016/j.procs.2024.04.235.
[143]. M. S. Rahman et al., "Ransomware Attack Detection using Machine Learning Approaches," INOCON 2024, Mar. 2024. DOI: 10.1109/IN-OCON60754.2024.10512276.
[144]. T. Radhakrishna and N. E. Majd, "Edge computing ransomware detection in IoT," ICNC 2024, Feb. 2024. DOI: 10.1109/ICNC59896.2024.10556351.
[145]. S. Zhang et al., "Early Detection and Defense Countermeasure Inference of Ransomware," IJACSA, vol. 14, no. 10, pp. 1-12, Oct. 2023. DOI:10.14569/IJACSA.2023.0141095.
[146]. M. A. Ayub et al., "RWArmor: a static-informed dynamic analysis approach for early detection," Int. J. Inf. Secur., vol. 23, pp.533-556, Sep. 2023. Available: https://link.springer.com/article/10.1007/s10207-023-06795-9 .

[147]. J. Lee et al., ”A Study on Countermeasures against Neutralizing Technology: Encoding Algorithm-Based Ransomware Detection Methods Using ML,” Electronics, vol. 13, no. 6, p. 1030, Mar. 2024. DOI: 10.3390/electronics13061030.
[148]. Y. Rbah et al., ”A fog-based attack detection model using deep learning,” ICAISE 2023, vol. 838, pp. 506-511, Jan. 2024.
[149]. Schneier, B. (2015). Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World. W.W. Norton & Company. ISBN 978-0393244816
[150]. Rose, S., et al. (2020). Zero Trust Architecture. National Institute of Standards and Technology (NIST) Special Publication 800-207. https://doi.org/10.6028/NIST.SP.800-207
[151]. Aminu, M., et al. (2024). Enhancing Cyber Threat Detection through Real-time Threat Intelligence and Adaptive Defense Mechanisms. International Journal of Computer Applications Technology and Research Volume 13–Issue 08, 11 – 27, 2024. https://doi.org/10.1109/ACCESS.2020.2976111
[152]. Ofoegbu, K., et al. (2023). Empowering users through AI-driven cybersecurity solutions: enhancing awareness and response capabilities. Engineering Science & Technology Journal, Volume 4, Issue 6. https://doi.org/10.51594/estj.v4i6.1528
[153]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy. https://doi.org/10.1109/SP.2010.25
[154]. Rose, S., et al. (2020). Zero Trust Architecture. National Institute of Standards and Technology (NIST) Special Publication 800-207. https://doi.org/10.6028/NIST.SP.800-207
[155]. Kainerstorfer, M., et al. (2011). Software security for small development teams: a case study. iiWAS'2011 - The 13th International Conference on Information Integration and Web-based Applications and Services. https://doi.org/10.1145/2095536.2095590
[156]. Schneier, B. (2015). Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World. W.W. Norton & Company. ISBN 978-0393244816
[157]. Chandola, V., et al. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58. https://doi.org/10.1145/1541880.1541882
[158]. Dash, B., et al. (2022). Threats and Opportunities with AI-based Cyber Security Intrusion Detection: A Review. International Journal of Software Engineering & Applications 13(5). https://doi.org/10.5121/ijsea.2022.13502
[159]. Garcia-Teodoro, P., et al. (2009). Anomaly-based network intrusion detection: Techniques, systems, and challenges. Computers & Security, 28(1), 18-28. https://doi.org/10.1016/j.cose.2008.08.003
[160]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy. https://doi.org/10.1109/SP.2010.25
[161].
[162]. Kainerstorfer, M., et al. (2011). Software security for small development teams: a case study. iiWAS'2011 - The 13th International Conference on Information Integration and Web-based Applications and Services. https://doi.org/10.1145/2095536.2095590
[163]. Dash, B., et al. (2022). Threats and Opportunities with AI-based Cyber Security Intrusion Detection: A Review. International Journal of Software Engineering & Applications 13(5). https://doi.org/10.5121/ijsea.2022.13502.

[164]. Garcia-Teodoro, P., et al. (2009). Anomaly-based network intrusion detection: Techniques, systems, and challenges. Computers & Security, 28(1), 18-28. https://doi.org/10.1016/j.cose.2008.08.003

[165]. Bhuyan, M. H., et al. (2014). Network anomaly detection: Methods, systems, and tools. IEEE Communications Surveys & Tutorials, 16(1), 303-336. https://doi.org/10.1109/SURV.2013.052213.00046

[166]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy. https://doi.org/10.1109/SP.2010.25

[167]. Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences 9(20):4396. https://doi.org/doi.org/10.3390/app9204396

[168]. Kim-Hung, L., et al. (2022). IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT. Electronics, 11(4), 524. https://10.3390/electronics11040524

[169]. Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176. https://doi.org/10.1109/COMST.2015.2494502

[170]. Eskin, E., et al. (2002). A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. Detecting Intrusions in Unlabeled Data. Springer Nature. https://doi.org/10.1007/978-1-4615-0953-0_4

[171]. Aminu, M., et al. (2024). Enhancing Cyber Threat Detection through Real-time Threat Intelligence and Adaptive Defense Mechanisms. International Journal of Computer Applications Technology and Research Volume 13–Issue 08, 11 – 27, 2024. https://doi.org/10.1109/ACCESS.2020.2976111

[172]. Kim-Hung, L., et al. (2022). IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT. Electronics, 11(4), 524. https://10.3390/electronics11040524

[173]. Gokhale, S. (2007). Architecture-Based Software Reliability Analysis: Overview and Limitations. IEEE Transactions on Dependable and Secure Computing. Volume: 4, Issue: 1. https://doi.org/10.1109/TDSC.2007.4

[174]. Garcia-Teodoro, P., et al. (2009). Anomaly-based network intrusion detection: Techniques, systems, and challenges. Computers & Security, 28(1), 18-28. https://doi.org/10.1016/j.cose.2008.08.003

[175]. Dash, B., et al. (2022). Threats and Opportunities with AI-based Cyber Security Intrusion Detection: A Review. International Journal of Software Engineering & Applications 13(5). https://doi.org/10.5121/ijsea.2022.13502

[176]. Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences 9(20):4396. https://doi.org/doi.org/10.3390/app9204396

[177]. Cremer, F., et al. (2022). Cyber risk and cybersecurity: a systematic review of data availability. The Geneva Papers on Risk and Insurance - Issues and Practice, 47, 698-736. https://doi.org/10.1057/s41288-022-00266-6

[178]. Palo Alto Networks. (2022). Next-Generation Firewall Technology Overview. Retrieved from https://www.paloaltonetworks.com/

[179]. Dash, B., et al. (2022). Threats and Opportunities with AI-based Cyber Security Intrusion Detection: A Review. International Journal of Software Engineering & Applications 13(5). https://doi.org/10.5121/ijsea.2022.13502

[180]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy. https://doi.org/10.1109/SP.2010.25

[181]. Aminu, M., et al. (2024). Enhancing Cyber Threat Detection through Real-time Threat Intelligence and Adaptive Defense Mechanisms. International Journal of Computer Applications Technology and Research Volume 13–Issue 08, 11 – 27, 2024. https://doi.org/10.1109/ACCESS.2020.2976111

[182]. Antonakakis, M., et al. (2017). Understanding the Mirai botnet. Proceedings of the USENIX Security Symposium. ISBN 978-1-931971-40-9

[183]. Bhuyan, M. H., et al. (2014). Network anomaly detection: Methods, systems, and tools. IEEE Communications Surveys & Tutorials, 16(1), 303-336. https://doi.org/10.1109/SURV.2013.052213.00046

[184]. Chandola, V., et al. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

[185]. Ibitoye, O., et al. (2019). The Threat of Adversarial Attacks on Machine Learning in Network Security -- A Survey. Computer Science. https://doi.org/10.48550/arXiv.1911.02621