Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

RESEARCH ARTICLE

Zero-Day Vulnerability Prevention with Recursive Feature Elimination and Ensemble Learning

Mike Nkongolo Wa Nkongolo*

[1]Department of Informatics, University of Pretoria, Gauteng, South Africa

Correspondence
*Mike Nkongolo Wa Nkongolo, Faculty of Engineering, Built Environment and Information Technology, Pretoria 0028, South Africa. Email: mike.wankongolo@up.ac.za

Abstract

This research addresses the challenge of identifying and preventing novel network threats by optimising the UGRansome dataset for real-time anomaly detection. Utilising a hybridised approach involving naïve tree-based ensemble machine learning and recursive feature elimination (RFE), the study achieves a balanced accuracy of 97%, with naïve Bayes (NB) emerging as the most effective classifier. The proposed framework, combining gradient boosting (GB) and random forest (RF) as base models with NB as a blender, proves successful in detecting and preventing zero-day vulnerabilities. UGRansome makes a substantial impact by successfully preventing more than 100 kbps of malicious traffic using features extracted by the RFE technique. This is achieved through the utilisation of uniform resource locators (URLs) extracted by the RFE model, surpassing the performance of existing Intrusion Detection System (IDS) datasets. Secure shell attacks are notably thwarted, showcasing the dataset's efficacy in enhancing network security. The study highlights significant advancements in intrusion detection techniques. The NB model exhibits exceptional performance, surpassing other models in accuracy, precision, and recall across all classes, especially in zero-day vulnerability classification. Additionally, the proposed naïve tree-based ensemble model demonstrates remarkable accuracy and stands out as the top-performing technique among all models studied. Implementation of the UGRansome properties-based rule resulted in discernible changes in traffic classification, reducing unknown traffic while increasing unclassified traffic, warranting further investigation.

KEYWORDS:
UGRansome dataset, network security, machine learning, ensemble learning, deep packet inspection, policy and charging rules function

1 | INTRODUCTION

The field of intrusion detection systems (IDSs) has experienced a significant increase in the number of network attacks over the last two decades, with malicious activities proliferating across various network systems. Detecting these attacks presents two significant challenges for IDSs. Firstly, IDSs must detect unknown threats to prevent revenue loss and secure corporate assets[1]. Secondly, detection processes must minimise false alarms and unknown traffic classification rates to reduce inaccuracy in classifying malicious activities[2]. These challenges have significant implications for detecting and preventing novel network attacks such as advanced persistent threats (APT) and zero-day attacks[3]. Despite the increasing prevalence of novel network threats, the network security landscape lacks effective IDSs that are capable of accurately recognising and preventing APT and zero-day vulnerabilities. The inefficiency arises from the absence of adequately designed and innovative datasets essential for training and testing IDSs and implementing diverse network shaping and filtering rules[4]. Network shaping refers to the practice of controlling the speed or bandwidth of network traffic[4]. This is often implemented to manage and prioritise data flow, ensuring that certain types of traffic or applications receive specific levels of bandwidth[4]. Network shaping rules are configurations that dictate how the available bandwidth is allocated and distributed across different types of traffic. In contrast, network filtering rules involve controlling the passage of data packets based on specific criteria[4]. Filtering rules define conditions under which packets are allowed or denied passage through a network (Figure 1). This could include rules based on source or destination internet protocol (IP) addresses, port numbers, or other attributes (Figure 1). Filtering rules are commonly used for security purposes to permit or block certain types of traffic[4] (Figure 1).

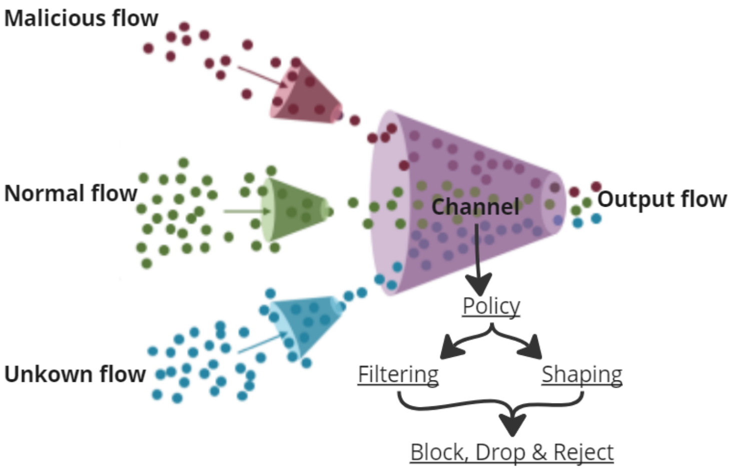


FIGURE 1 Filtering and shaping rules

The optimal IDS should incorporate a range of shaping and filtering rules or policies to promptly block abnormal network issues in real-time[5] (Figure 1). To achieve this goal, the network engineer must discern irregular patterns to configure the policy or rule[4]. For instance, to thwart access to illicit websites, the policy or rule should encompass malevolent uniform resource locators (URLs) patterns associated with these sites. Subsequently, the network engineer can eliminate these patterns from the network traffic, effectively prohibiting access to such sites[4]. Similarly, malevolent ports, internet addresses, and protocols can be configured to dismiss or discard atypical network traffic matching their patterns. Nevertheless, there are numerous emerging threats, such as phishing and mobile attacks, that remain challenging to detect due to the absence of new patterns that can be employed to implement networking policies and protect the network from such threats. This article explores the development of a new IDS that can accurately identify and prevent unknown and new network threats in real time using innovative networking policies. The central aim of this research is to enhance the capabilities of the IDS by incorporating a distinctive set of patterns derived from an anomaly detection dataset named UGRansome[6] to thwart malicious network activities. To achieve this goal, a compilation of uncommon patterns from the UGRansome dataset was utilised to configure the network policy[4,6]. The IDS has been configured to disallow any network traffic that aligns with the identified malicious patterns from the crafted UGRansome dataset, thereby identifying novel and menacing network activities. This study has optimised the UGRansome dataset for real-time detection and prevention of aberrant network traffic, fortifying the organisation against malicious attacks. The network intrusion detection problem (NIDP) is defined as an intrusion made by a novel network threat that accesses the information system (IS) and performs illegal actions such as misusing confidential information or encrypting relevant files[7]. Different systems have been designed to detect network threats, but they suffer from various issues: the identification of false IP addresses, encrypted network flow, and unknown intrusions[8,7,2]. Therefore, to mitigate the problem, researchers in the network security community[2],[9] have proposed incorporating machine learning (ML) techniques into the IDS to improve the inaccuracies of the detection performance[10].

These ML-based approaches aim to enhance the system's ability to identify and classify various types of attacks, such as novel and evolving threats, unknown patterns, and subtle anomalies, thereby bolstering the overall effectiveness and efficiency of intrusion detection mechanisms. Nevertheless, most of the suggested techniques used a single ML classifier, which still suffers from a false alarm rate[10,9]. Hence, various studies[11,12,13] have proposed the use of ML based on ensemble with feature selection[14]. Ensemble learning can be defined as the process of utilising various ML algorithms that are strategically combined to resolve a specific computational problem[14]. With ensemble learning algorithms, an improvement in terms of the prediction or classification of network threats is achieved by reducing the likelihood of a single ML classifier[15]. An ensemble learning is different from a feature selection algorithm, which can be thought of as a feature extractor method that provides the most relevant attribute of a given dataset to optimise the classification results[16]. Recursive feature elimination (RFE) is a feature selection technique used in ML to select a subset of features by recursively removing the least important ones[17]. It is often applied in conjunction with an ML model to identify and retain the most relevant features for better model performance. A general outline of how RFE works has been presented as follows:

1. Build a model: Train an ML model on the entire set of features.

2. Rank features: Assess the importance of each feature in the model. This can be achieved using coefficients, feature importance scores, or other relevant metrics depending on the type of model used[17].

3. Eliminate features: Remove the least important feature(s) from the dataset.

4. Repeat: Iterate steps 1-3 with the reduced feature set. This process is recursive and continues until a predefined number of features is reached or performance criteria are met.

5. Evaluate performance: Assess the model's performance using the selected subset of features.

The RFE is commonly used with models that provide feature importance scores, such as linear models, tree-based models (like decision trees and RF), or support vector machines (SVM)[17]. By iteratively eliminating the least important features, RFE helps identify the most relevant features for a given predictive task, potentially improving model interpretability and performance[17]. It also aids in reducing overfitting by focusing on the most informative features. The optimal number of features to retain is often determined through cross-validation or other performance metrics. While RFE can be a powerful tool for feature selection, the choice of the ML algorithm and the criteria for feature importance play a crucial role in its effectiveness.

The proposed methodology of this research combined both RFE and ensemble learning to improve and optimise the classification accuracy of novel network-threatening behaviour. In this study, the paramount features generated by the RFE have been utilised to configure the shaping and filtering rules to prevent zero-day vulnerabilities in real time. Zero-day vulnerabilities, often referred to as zero-days, are security flaws or weaknesses in software applications that are unknown to the software vendor and, therefore, lack a patch or fix[18].

The term "zero-day" indicates that the developers have had zero days to address and resolve the issue, making it a critical and potentially exploitable weakness. A network IDS is considered accurate when the intrusion rate is low in terms of false alarms and performs a high classification rate in terms of accuracy[19,18]. To implement a robust network IDS, an increase in accuracy rate and a decrease in false alarms are expected due to the classification of network events. The classification is achieved by the decision component of the IDS, which determines if networking features are abnormal or not[20].

The main issue in testing the performance of current IDSs is the recognition of zero-day vulnerabilities. A zero-day exploit is a hidden or unknown malicious intrusion[21]. These types of intrusions occur without being detected by the IDS. As a result, IDS vendors have zero days to develop keys to recognise them. Zero-day describes a novel malware that hackers utilise to attack networks by targeting system vulnerabilities. The term zero-day is commonly used in literature to describe a situation where developers have recently discovered a vulnerability in a system[18]. During this stage, attackers can exploit the network's vulnerabilities with ease, as no effective security measures have been implemented[22]. This makes zero-day intrusion a serious security threat for any organisation.

1.1 | Problem Statement

Ensemble learning and RFE methods have not been fully used in the detection and prevention of zero-day attacks. Legacy datasets are also used in experiments reported in the current IDS literature.

To this end, this study will optimise a new set of the UGRansome dataset to assess the classification performance of the ensemble learning and RFE in the detection and prevention of zero-day threats and use the most relevant features produced by the RFE to enforce various shaping and filtering rules on the network to block malicious traffic. This research will also compare the performance of the UGRansome dataset with existing datasets. The core problem of this research is the detection and prevention of novel malicious network concerns to optimise the security of an IDS. The core focus of the study is to optimise the UGRansome with a set of attributes that can be used to reject or drop malicious activities on the network. The benefit of detecting zero-day vulnerabilities is to protect information technology (IT) systems, data, and programs from cyber threats, such as sabotage, fraudulent activities, or espionage. Cybersecurity is important because it protects all kinds of data, whether it is personal information, intellectual property, or government information. Adequate network security policies such as robust firewalls, advanced threat detection systems, and regular vulnerability assessments can protect individuals and businesses from ransomware, zero-day attacks, and APT such as Cozy Bear and OceanLotus[23]. The greatest advantage of this study is that it provides a cybersecurity solution to tackle zero-day vulnerabilities. The proposed solution leverages the UGRansome dataset's unique features, and configured shaping and filtering rules to counteract zero-day vulnerabilities effectively. This demonstrates the solution's capacity to safeguard businesses across various industries from emerging cybersecurity threats. Moreover, internet data leaves businesses vulnerable to different cyber attacks that take advantage of unsecured access points. For instance, South Africa has been listed fifth in the top 10 countries globally afflicted by cybercriminal activities (Figure 2).

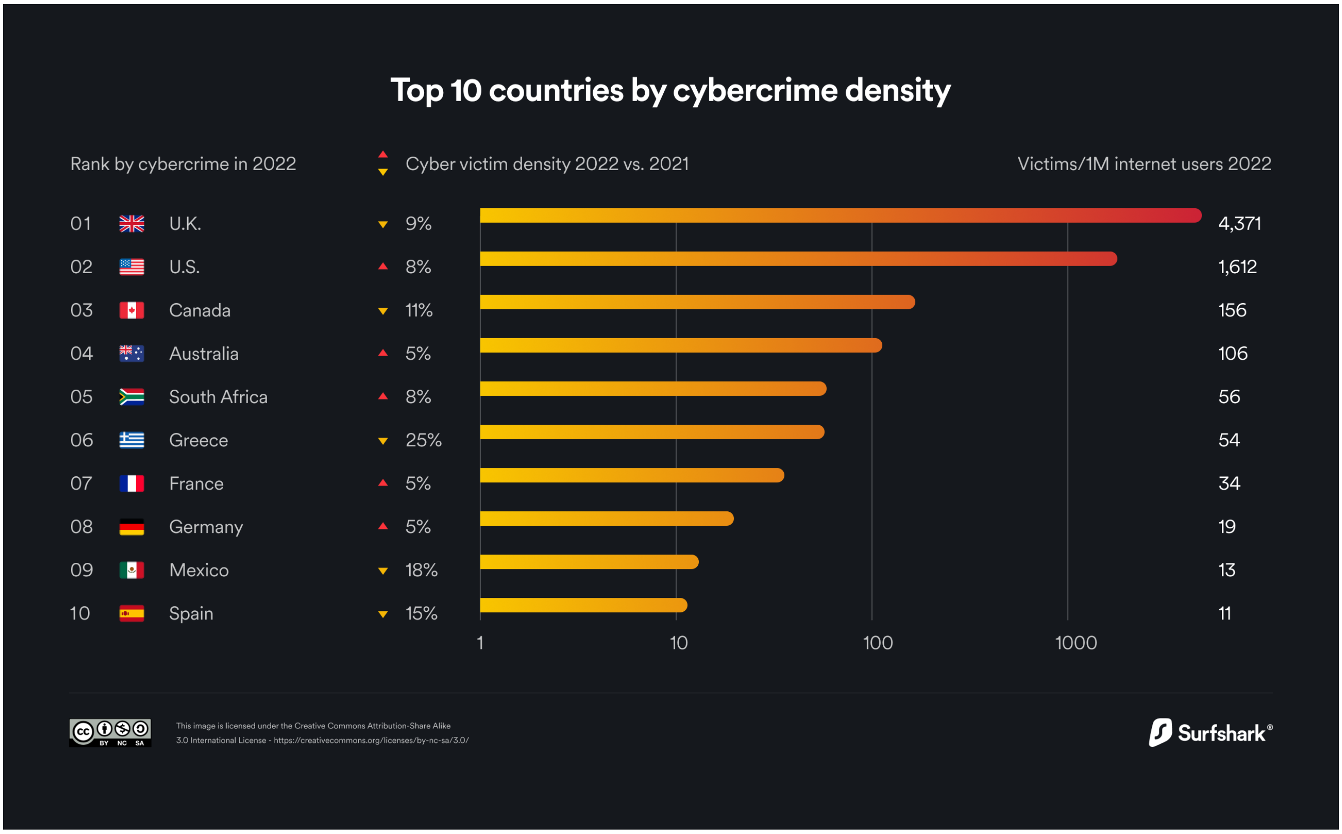


FIGURE 2 Top ten countries by cybercrime density[24]

This ranking is based on the number of cybercrime victims per one million internet users, as reported to the Federal Bureau of Investigation (FBI) during 2021 and 2022[24]. Therefore, to protect personal information, various countries worldwide have instituted measures such as the General Data Protection Regulation (GDPR)[25]. For instance, the GDPR implemented by the European Union (EU), aims to strengthen the protection of individuals' data and enhance their control over their information[25]. It establishes strict guidelines for organisations handling personal data, including requirements for consent, data breach notification, and the rights of individuals to access, rectify, and erase their data.

This development stems from the fact that data security has become an essential aspect of human rights and privacy. Figure 3 illustrates the number of countries that have fully enacted data protection, those with pending enactment, and those with no data protection initiatives as of 2023.

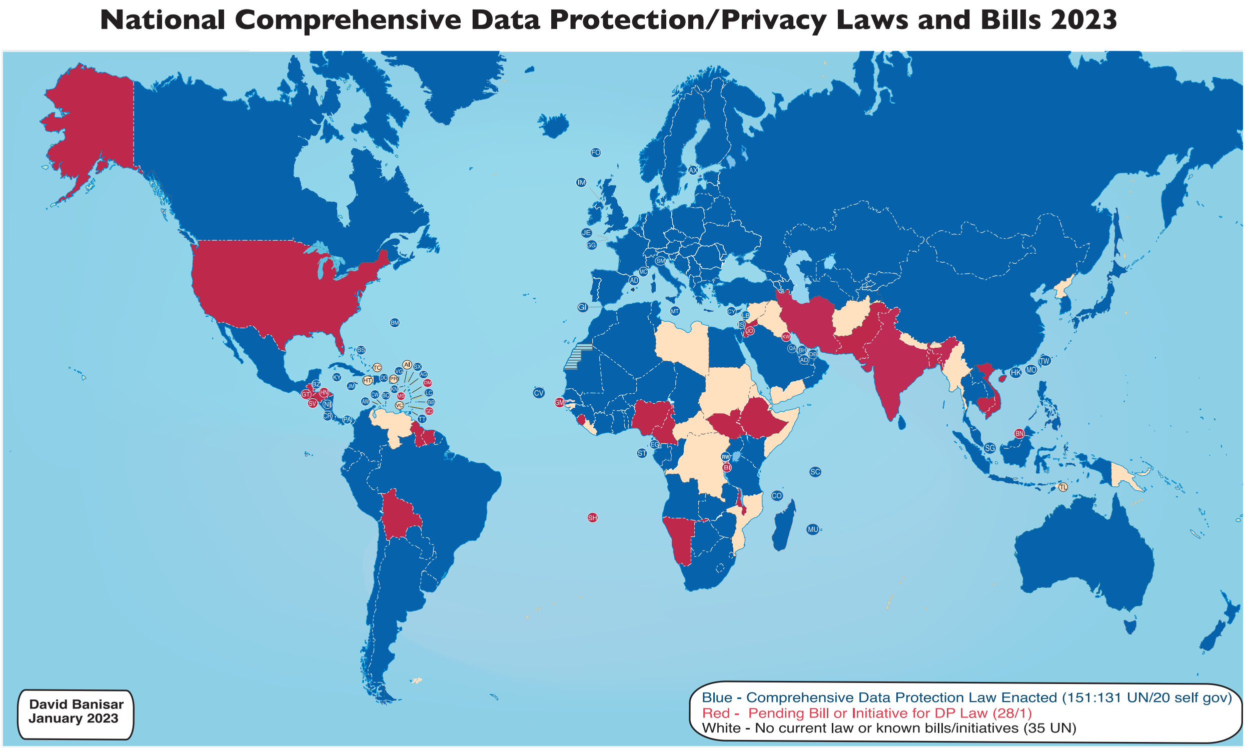

National Comprehensive Data Protection/Privacy Laws and Bills 2023
David Banisar
January 2023
Blue - Comprehensive Data Protection Law Enacted (151:131 UN/20 self gov)
Red - Pending Bill or Initiative for DP Law (28/1)
White - No current law or known bills/initiatives (35 UN)

FIGURE 3 National data protection laws and bills[24]

Figure 4 illustrates the growth of cybercrime worldwide[24]. These figures depict the necessity of safeguarding information in response to global apprehensions concerning data security (Figure 2 to Figure 4). Given the rising amount of information exchanged in a network, adopting zero-day vulnerability prevention has become indispensable to protect confidential data. Hence, this paper strives to assess the proficiency of an IDS in leveraging cyclostationary data to safeguard information, particularly in the context of network security.

1.2 | Research Question

Given the outlined research problem, the primary research question can be articulated as follows:

- How do ensemble learning and RFE contribute to identifying novel network threats while reducing false alarms, avoiding misclassification of traffic as unknown, and simultaneously establishing a secure configuration based on anomalous patterns?

To answer the main research question, the study emphasises the importance of optimising the UGRansome dataset[6]. This study approaches the main research question by optimising the UGRansome dataset because the focus is on the recognition of comprehensive, up-to-date, and labelled training sets with various novel threatening behaviour types that can be blocked by current IDSs. This can only be possible if we optimise a novel dataset such as UGRansome. The manifestation of ensemble learning and the RFE will depend on the classification results achieved on the optimised dataset.

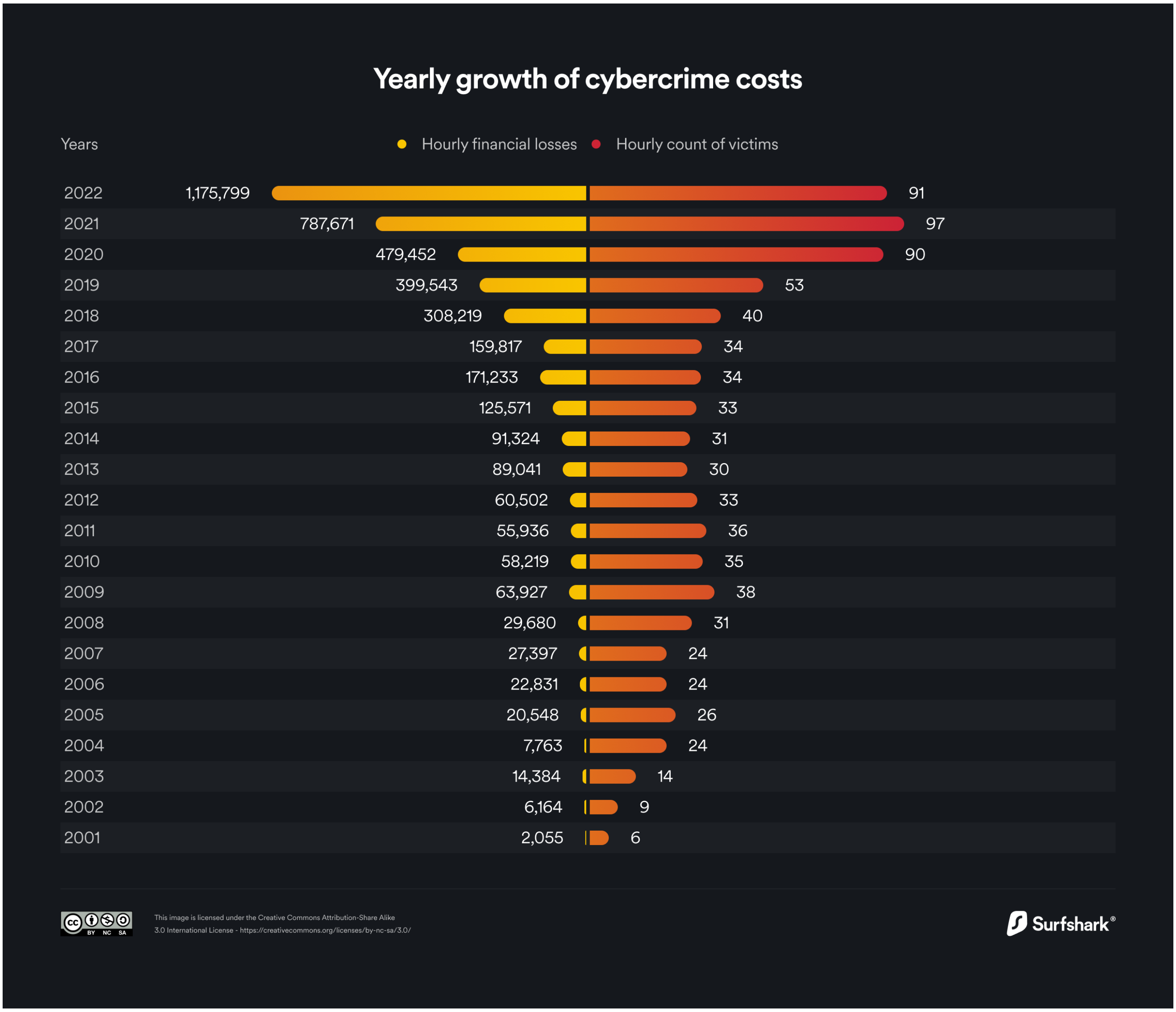


FIGURE 4 Yearly growth of cybercrime costs[24]

The advantage of this approach is that one can utilise the attributes provided by the RFE to enforce security on the IDS by creating different shaping and filtering rules.

1.3 | Fundamental Assumptions

Based on the aforementioned research question, the following fundamental assumptions are derived:

- Ensemble learning and the RFE will increase the classification rate of zero-day vulnerabilities on the optimised UGRansome dataset.
- Anomaly detection through data visualisation will spot abnormal features of the optimised UGRansome dataset.
- The UGRansome attributes extracted by the RFE can be used by the IDS to block abnormal traffic.
- This research posits that the IDS can only provide sufficient results based on the quality and number of features extracted by the RFE technique.

- The ensemble learning and RFE framework can only feasibly yield successful performance based on the optimisation scheme efficiency.

- This study posits that feature extraction and feature optimisation are the most integral phases in this research, as those phases have a direct effect on the eventual classification, prediction, and prevention rates.

1.4 | Delineation and Limitations

The study will not create a real-time IDS and it will not compare various feature extraction algorithms or consider the creation of new classifiers to address the problem of zero-day vulnerabilities, classification, and prevention. By optimising the UGRansome dataset or using the ensemble learning and RFE framework, the main research question will be answered.

1.5 | Demarcation

In this research, the UGRansome dataset was subjected to a series of rigorous pre-processing and encoding steps, rendering it comprehensible to the discerning scheme of ML algorithms. Furthermore, the RFE, a pinnacle of feature selection, was set in motion to discern and extract the most relevant features for the subsequent stacking ensemble learning phase, which served as the bedrock for classification and prediction. In this intricate ensemble process, the formidable alliance of gradient boosting (GB) and random forest (RF) reigned as base models (member models), with the ever-discerning naive Bayes (NB) stepping in as the meta-model (blender), seamlessly harmonising the distinctive strengths of the base models. The pivotal experiment unfurled with the UGRansome dataset at its core, poised to unveil its superiority over existing datasets. Additional layers of complexity can be added to this profound exploration with the introduction of other ML algorithms, including distinguished tree-based models such as extreme gradient boosting (XGBoost) and extra trees. Upon the culmination of the experimentation, the discerned paramount features, as ascertained through the collaborative efforts of ensemble learning and the RFE, assumed the pivotal role of crafting intricate filtering and shaping network rules or policies to block malicious traffic. These policies were meticulously tailored to preemptively thwart the progress of malevolent network traffic, thus laying the foundation for a resilient and fortified network security infrastructure. In essence, the demarcation within this research extended beyond mere theoretical delineation; it evolved into an empirical odyssey that paved the way for the practical application of remarkable discoveries within the field of applied cybersecurity.

1.6 | Research Objectives

This research follows the following objectives:

- Optimise a new set of the UGRansome dataset to achieve the objective of the recognition and prevention of zero-day vulnerabilities.

- Compute an ensemble learning and RFE model on UGRansome for zero-day vulnerabilities threat recognition.

- Evaluate the performance of the proposed model with the following metrics: precision, recall, F1 score, sensitivity, specificity, and accuracy.

- Evaluate the proposed UGRansome model's performance against that of other models operating on established datasets for comparison.

- Use the most optimal features of UGRansome that have been retrieved by the RFE and configure an existing IDS that will block malicious traffic in real time.

1.7 | Original Research Contribution

The proposed theoretical methodology for optimising and evaluating UGRansome will be showcased through a series of meticulous experiments and comprehensive simulations, underscoring the novelty of this research. This work goes beyond merely explaining the manifestation of RFE and ensemble learning in preventing zero-day vulnerabilities.

Key performance objectives encompass reducing false alarms, restraining the classification of traffic as unknown, and simultaneously establishing a secure configuration based on anomalous UGRansome patterns. To delve into the details, the research novelty unfolds in configuring shaping and filtering rules to block malicious traffic and mitigate unknown traffic. This is achieved by utilising the most important features identified by the RFE. The RFE-derived features play a pivotal role in shaping the rules and guiding the system to discern and respond to malicious patterns effectively. By incorporating these features into the shaping and filtering rules, the research establishes a robust defense mechanism, minimising the chances of false alarms, curtailing misclassification, and enhancing the overall security posture against zero-day vulnerabilities in the UGRansome context.

This article is structured into four sections: Section 2 covers the background, discussing related works. In Section 3, the methodology is presented, focusing on the optimised UGRansome dataset. Section 4 delves into the results, covering the outcomes of ensemble learning RFE as well as results concerning shaping and filtering rules. Finally, Section 5 comprises the conclusion, which highlights unexpected anomalies and outlines directions for future research.

2 | BACKGROUND

The recent targeting of critical infrastructure by unknown attacks has highlighted the necessity for a deeper understanding of threat patterns and landscapes[26]. However, achieving this is challenging due to limited access to publicly available data. This section aims to explicate the rationale behind creating and optimising a network anomaly detection dataset, as well as providing a broad overview of its most significant features. Specifically, the section outlines the data structure of an anomaly detection dataset, followed by a discussion of the packet inspection component of the IDS, which can be utilised for real time testing of the dataset. The primary focus of this section is on the identification of useful and relevant network features that can be utilised to develop an anomaly detection dataset. The ultimate goal of this dataset is to enhance network security through the identification of abnormal patterns. This objective is directly linked to the research question. The section aims to provide a systematic analysis of abnormal network feature patterns, which will aid in the detection of new network threat anomalies. Additionally, the IDS is introduced to illustrate how abnormal patterns can be utilised to create various rules on the network to detect network concerns.

2.1 | Cyclostationary Features

Cyclostationary features in an IDS refer to specific patterns or characteristics in network traffic that exhibit cyclostationary behaviour[27]. Cyclostationarity is a property of network signals that exhibit statistical variations over time, but their statistical properties repeat periodically. In the context of an IDS, cyclostationary features are utilised to detect anomalous or malicious network activity[28,6]. By analysing the cyclostationary properties of network traffic, an IDS can identify patterns or deviations that indicate potential security breaches or attacks. Cyclostationary features are characterised by abnormal and long-term evolution traffic patterns that can lead to various types of intrusion, including phishing and secure shell attacks[29]. Unlike normal features, cyclostationary features exhibit abnormal properties that vary cyclically over time[30,29]. Cyclostationary features can be extracted from various aspects of network traffic, such as packet inter-arrival times, packet ports, or protocol-specific addresses. These features are then analysed using techniques such as spectral analysis or cyclostationary analysis to detect any abnormal or malicious activity that may indicate an ongoing attack or intrusion attempt[29]. Incorporating cyclostationary features into an IDS enhances its ability to detect sophisticated attacks that may not be easily identified by traditional signature-based or anomaly-based detection methods. By focusing on the cyclostationary properties of network traffic, an IDS can provide a more robust and accurate detection of intrusions, ultimately improving the security of the network infrastructure. Stochastic cyclostationary traffic can be divided into different samples, denoted as $T_1, T_2, ..., T_n$, with the network threat hidden in the abnormal traffic sample that exhibits cyclic patterns c_t[29].

$$T = \sum_{t=1}^{T_n} c_t = 0 \tag{1}$$

T is the set of cyclostationary features, and T_n is malicious when $c_t = 0$, and normal when $c_t = 1$[29].

$$T = \sum_{t=1}^{T_n} c_t = 1 \tag{2}$$

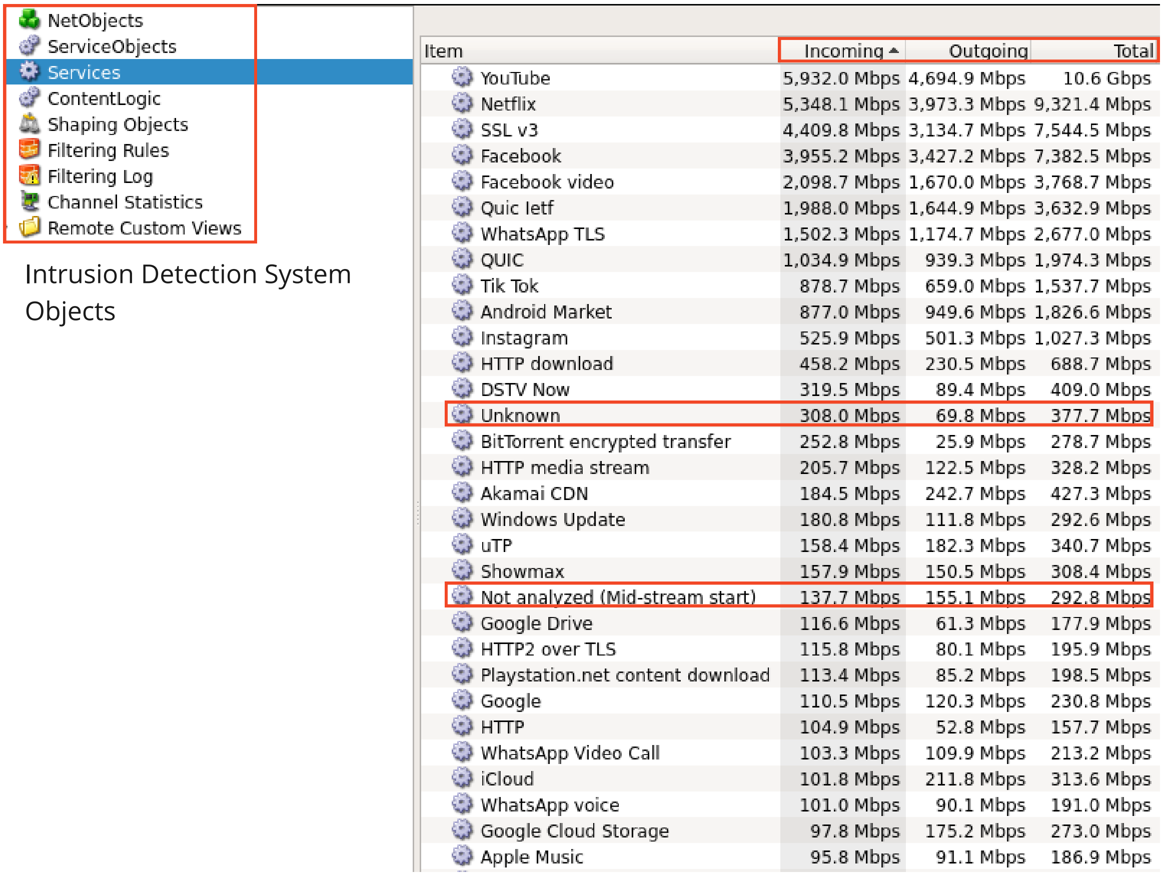
The identification and categorisation of cyclostationary malware involve the detection and separation of malicious samples into distinct and adjacent partitions made of 0 and 1. These partitions are then used to analyse and study the detected malware. Each point within a partition T_n represents a cyclostationary feature pattern associated with normal (1) or malicious behaviour (0). Cyclostationary attacks differ from traditional attacks in terms of their underlying characteristics and detection methodologies[29]. In contrast to traditional network attacks that are no longer prevalent, cyclostationary attacks employ updated protocols and exploit the periodicity in signal properties, making their detection more challenging using traditional methods. On the other hand, detecting cyclostationary attacks requires specialised analysis techniques that focus on the cyclostationary properties of signals[29].

These techniques may involve cyclostationary analysis, spectral analysis, or other advanced signal-processing methods that can capture the periodic variations in the attack behaviour[31,27]. The defence against cyclostationary attacks often requires the integration of specialised detection algorithms and feature extraction methods that can effectively capture and analyse the cyclostationary properties of signals. Cyclostationary attacks exhibit distinct behaviour patterns, require specialised detection techniques, involve higher levels of sophistication, and may require different defence mechanisms compared to traditional attacks[29]. Understanding these differences is crucial for developing effective detection and prevention strategies against cyclostationary attacks. Cyclostationary traffic can be classified into several categories, including connection, timestamp, protocol, ports, and addresses.

This study used three main types of cyclostationary protocols of the UGRansome dataset[6]: User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Internet Control Message Protocol (ICMP)[32]. Unlike some other protocols, UDP does not inherently exhibit cyclostationary properties[33] but it is worth mentioning that the application or data transmitted over UDP can introduce cyclostationary properties if the payload or pattern of the data exhibits periodic behaviour. For example, if an application using UDP generates data with an inherent periodicity or if the payload itself has cyclostationary characteristics, those properties could be observed in the overall UDP traffic. TCP is a connection-oriented protocol used for reliable data transmission over networks[34]. As a protocol, TCP does not exhibit inherent cyclostationary properties in its design but it is important to note that the behaviour of TCP traffic can exhibit statistical properties that may appear cyclostationary under certain conditions. These properties can arise due to the congestion control mechanisms and underlying algorithms employed by TCP. For example, variations in the round-trip time (RTT) and congestion window size may introduce patterns that exhibit periodic or quasi-periodic behaviour in the statistical properties of TCP traffic. Furthermore, the behaviour of TCP traffic can also be influenced by the applications or data being transmitted over it. If the data exhibits cyclostationary characteristics, such as periodic data patterns or periodic data rate variations, those characteristics may be observed in the statistical properties of the TCP traffic. ICMP messages are typically encapsulated within IP packets and are used for tasks such as ping, traceroute, and network error notifications[35]. Moreover, ICMP messages do not exhibit cyclostationary properties. ICMP is primarily a message-based protocol that provides control and informational messages rather than data transmission[36]. These messages are typically small and contain specific information related to network diagnostics or error reporting. It is worth noting that ICMP traffic, like any network traffic, can be influenced by various factors, such as network congestion, routing patterns, or the behaviour of the applications generating ICMP messages.

These factors can introduce statistical variations in the timing or arrival patterns of ICMP messages, potentially leading to some degree of cyclostationarity in the statistical properties of ICMP traffic. While ICMP messages do not possess inherent cyclostationary properties, the statistical properties of ICMP traffic can exhibit some periodic or quasi-periodic behaviour, depending on the underlying network conditions and the behaviour of the applications generating ICMP messages. Novel network attacks exploit these statistical variations to infiltrate a system. For example, ransomware can use different ports, addresses, and protocols at different timestamps to attack a specific network[35]. Such attacks can be difficult to detect due to the variability of connection types (unicast or multicast), protocols (TCP or UDP), and addresses (IP or URL). Therefore, to construct an effective anomaly detection dataset for zero-day threat detection, the dataset's properties must exhibit a balanced distribution and proportion of various types of cyclostationary behaviour. The cyclostationarity of the aforementioned protocol and its exploitation by zero-day attacks highlight the need for a robust anomaly detection dataset to ensure the security of a network. Figure 5 illustrates the classification of cyclostationary traffic using an IDS. By using such a system, the partitioned cyclostationary features can be stratified and studied for malware detection. The visual representation in Figure 5 displays the combined volume of inbound and outbound network traffic and identifies several categorised services, including YouTube, Netflix, Apple, and Facebook. Interestingly, the unidentified traffic (377.7 Mbps) surpasses the volume of unclassified traffic (292.8 Mbps), indicating a shortfall in detecting anomalous and cyclostationary patterns.

An optimal solution is to use significant features from a cyclostationary dataset and configure the IDS objects to create filtering rules that can reject malicious traffic. These rules aim to reduce unknown traffic and block abnormal traffic, ultimately enhancing network security.



Item	Incoming ▲	Outgoing	Total
YouTube	5,932.0 Mbps	4,694.9 Mbps	10.6 Gbps
Netflix	5,348.1 Mbps	3,973.3 Mbps	9,321.4 Mbps
SSL v3	4,409.8 Mbps	3,134.7 Mbps	7,544.5 Mbps
Facebook	3,955.2 Mbps	3,427.2 Mbps	7,382.5 Mbps
Facebook video	2,098.7 Mbps	1,670.0 Mbps	3,768.7 Mbps
Quic Ietf	1,988.0 Mbps	1,644.9 Mbps	3,632.9 Mbps
WhatsApp TLS	1,502.3 Mbps	1,174.7 Mbps	2,677.0 Mbps
QUIC	1,034.9 Mbps	939.3 Mbps	1,974.3 Mbps
Tik Tok	878.7 Mbps	659.0 Mbps	1,537.7 Mbps
Android Market	877.0 Mbps	949.6 Mbps	1,826.6 Mbps
Instagram	525.9 Mbps	501.3 Mbps	1,027.3 Mbps
HTTP download	458.2 Mbps	230.5 Mbps	688.7 Mbps
DSTV Now	319.5 Mbps	89.4 Mbps	409.0 Mbps
Unknown	308.0 Mbps	69.8 Mbps	377.7 Mbps
BitTorrent encrypted transfer	252.8 Mbps	25.9 Mbps	278.7 Mbps
HTTP media stream	205.7 Mbps	122.5 Mbps	328.2 Mbps
Akamai CDN	184.5 Mbps	242.7 Mbps	427.3 Mbps
Windows Update	180.8 Mbps	111.8 Mbps	292.6 Mbps
uTP	158.4 Mbps	182.3 Mbps	340.7 Mbps
Showmax	157.9 Mbps	150.5 Mbps	308.4 Mbps
Not analyzed (Mid-stream start)	137.7 Mbps	155.1 Mbps	292.8 Mbps
Google Drive	116.6 Mbps	61.3 Mbps	177.9 Mbps
HTTP2 over TLS	115.8 Mbps	80.1 Mbps	195.9 Mbps
Playstation.net content download	113.4 Mbps	85.2 Mbps	198.5 Mbps
Google	110.5 Mbps	120.3 Mbps	230.8 Mbps
HTTP	104.9 Mbps	52.8 Mbps	157.7 Mbps
WhatsApp Video Call	103.3 Mbps	109.9 Mbps	213.2 Mbps
iCloud	101.8 Mbps	211.8 Mbps	313.6 Mbps
WhatsApp voice	101.0 Mbps	90.1 Mbps	191.0 Mbps
Google Cloud Storage	97.8 Mbps	175.2 Mbps	273.0 Mbps
Apple Music	95.8 Mbps	91.1 Mbps	186.9 Mbps

FIGURE 5 Classification of network traffic using an IDS

2.2 | Analysis of Cyclostationary Features

A form of attack, known as the cyclostationary attack, has emerged, which encrypts data files and restricts access[37]. This attack is difficult to detect due to its various penetration methods, which may include a script with malicious attachments, such as suspected files. For instance, ransomware uses techniques like web applications, email links, and attachments to penetrate a network. Other types of intrusion like phishing, wiretapping, denial of service (DoS), and probes have also evolved[37].

A comparative study by Hindy et al[38] revealed that IDSs report an alarming percentage of unknown network flow and anomalies. As such, it has become imperative to understand the scale of intrusion techniques used in cyclostationary attacks to prevent them. Signature and behavioural analyses can be used to detect cyclostationary features. Signature-based methods rely on the real time monitoring of traffic to detect anomalies[39,40]. Relevant features such as network objects, system logs, and diagnostics are analysed. On the other hand, behavioural analysis involves constructing a limited set of comprehensive and cyclostationary features to optimise the IDS in terms of detecting novel network concerns.

2.3 | Cyclostationary Features Taxonomy

The KDD99 and NSL-KDD datasets are commonly known as legacy and cyclostationary datasets that can be utilised to classify cyclostationary features[41,38]. Figure 6 displays the unbalanced distribution of normal network threat categories, such as multihop, Neptune, Nmap, and Perl, within the KDD99 dataset. This highlights the need for more robust datasets that can better represent the current network threat landscape.

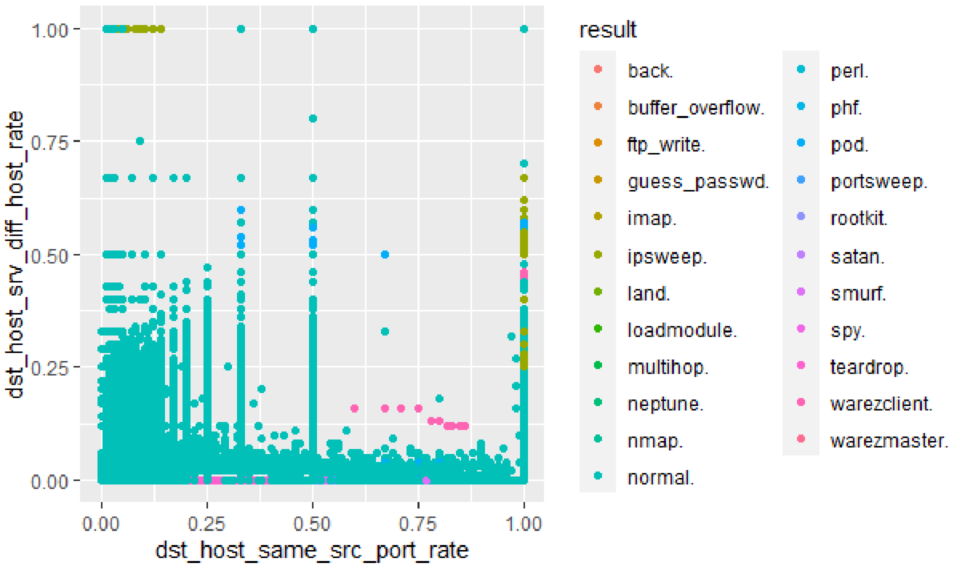


FIGURE 6 The KDD99 distribution

To analyse the cyclostationarity of the KDD99 dataset, Figure 6 visualises the difference in traffic between the source and destination ports. However, due to its unbalanced nature, with features skewed towards the normal category as may be seen in the bottom left side of Figure 6 , this legacy dataset does not provide effective options to study novel network threats. As shown in Figure 6 , the classification and prediction of features are biased towards the normal category. Apart from port traffic, the network flag can also be utilised to detect cyclostationarity, where each flag anomaly can correspond to different attacks. The prediction computation can be expressed as follows:

$$Feature \leftarrow anomaly_{names} \leftarrow attack^{type}_{predictor}$$

Figure 7 demonstrates various erroneous connection flags and showcases a sequence of stateless protocols, including SH, SF, S3, S1, S0, S1, S3, RSTR, RSTO, and REJ from the NSL-KDD dataset, to identify which flag serves as a robust indicator of cyclostationary attacks. Figure 7 demonstrates an effective approach to predict probing and DoS attacks. Network flags serve the purpose of indicating the state of network transactions being encapsulated. For example, the push (PSH) flag is utilised by the TCP/IP to indicate that the string buffer is empty, while the acknowledgment (ACK) flag indicates that a message has been received[34].

Similarly, the finish (FIN) flag signals the termination of a network process. Given the description of cyclostationary features, it is important to investigate alternative data processing and encoding methods to enhance the detection of novel network threat anomalies[38,42]. To predict such attacks, the present study employed the following formula:

$$Feature \leftarrow cyclostationary_{threats} \leftarrow novel^{type}_{predictor}$$

The aim is to enhance the overall learning process of the framework, enabling it to achieve precise outcomes in identifying new network attacks. Table 1 outlines various types of features found in IDS datasets, which offer diverse categories of cyclostationary patterns that have undergone changes and advancements. To detect anomalies in the Internet of Things (IoT),

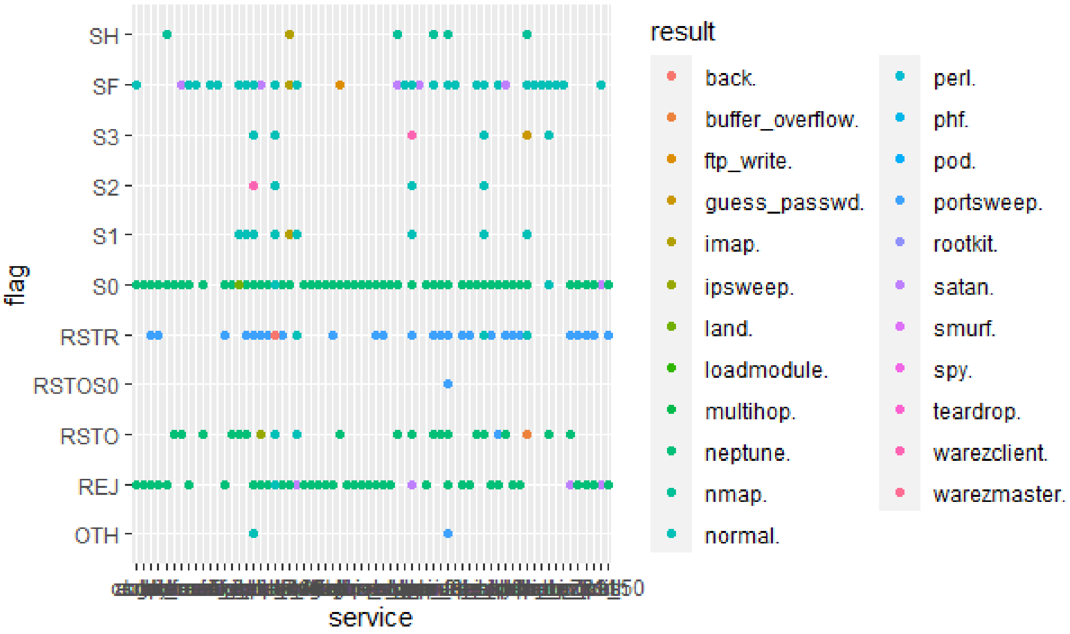


FIGURE 7 Network's flag anomaly

these features can be utilised by configuring rules in the IDS to block malicious servers, databases, operating systems, and network objects, which have been infected by cyclostationary attacks. These attacks utilise various cyclostationary features such as ports, addresses, throughput, and timestamps (Table 1). To optimise IDS performance for the recognition of zero-day threats, the study suggests optimising new balanced datasets that focus on such attacks. Furthermore, to generate comprehensive and well-designed cyclostationary datasets for the detection and recognition of novel network threat anomalies, additional features such as domain name servers (DNS), output strings, threats, and process information should also be taken into consideration. We outline a classification of novel network threats into three categories (signature, synthetic signature, and anomaly). These categories are part of the UGRansome dataset[6] and are utilised by the proposed ensemble learning and RFE to forecast the type of malicious activity.

2.4 | Synthetic Signature Attacks

The UGRansome dataset includes synthetic signature (SS) threats that aim to disrupt the computer network's performance[6]. This type of attack generates multiple TCP connections by creating packets or frames that originate from the attacker. The attack script uses the victim's network traffic to finetune its behaviour by specifying open network ports and calculating a packet rate to overwhelm incoming traffic[47]. Synthetic signature attacks mimic the cyclostationary properties of both known and unknown threats, including DoS, botnets, distributed DoS (DDoS), scan, and nerisbotnet[48]. Although signatures have been developed and updated to identify known threat types, such as DoS attacks, detecting unknown threats remains challenging due to the lack of novel detection keys.

2.5 | Signature Attacks

The UGRansome dataset includes the signature (S) attack category[6]. This type of attack leverages various network patterns, such as traffic flow and byte sequences, to infiltrate the system. Anti-viruses can mitigate this type of intrusion by recognising these patterns using different signatures. IDSs can detect signature attacks as their pattern recognition keys are available and frequently updated[49]. Furthermore, one can develop a specific key to effectively detect them because their behaviours are well understood.

TABLE 1 Existing cyclostationary features

Dataset	Feature	Limitation	Source
CAIDA	Port	Unbalanced Duplicate	- Ferdiana et al. (2020)[10]
CICIDS2017	Bugs	Zero logs Noisy	- Hindy et al. (2020)[38]
NF-BoT-IoT-v2	Timestamps	Unbalanced Format	- Ahsan, Rifat, Chowdhury, & Gomes (2022)[43]
UNSWNB-15	Packets	Drop analysis Packet inspection	M. Nkongolo, van Deventer, & Kasongo (2021)[6] M. M. Nkongolo, van Deventer, & Kasongo (2021)[6]
WSN-DS	Latency	Legacy dataset Novel patterns	Hindy et al. (2020)[38] -
Sperotto	Logs	Legacy dataset Documentation	- Gaurav et al. (2022)[44]
MAWI	Throughput	Unbalanced Noisy	- Chiche & Meshesha (2021)[45]
UNB ISCX	Protocol	TCP restricted Redundant	Kumar et al. (2020)[11] -
CTU-13	Addresses	Format Obfuscation	- Le et al. (2022)[33]
ADFA-LD12	Session	Capturing Tools	- Krishna & Bhanu (2021)[46]
InSDN	Packets	Big data Availability	M. Nkongolo, van Deventer, & Kasongo (2021)[6] -
UGRansome	Timestamp	Duplication Cryptographic	This work M. Nkongolo and M. Tokmak (2023)[18]
Ransomware	Attacks	Encryption	M. Nkongolo and M. Tokmak (2023)[18]

2.6 | Anomalous Attacks

The anomalous (A) attack class refers to a group of unknown network threats for which detection keys have not yet been developed[23]. To detect such attacks, an ML framework is typically employed that compares the traffic patterns of unknown traffic against known traffic patterns incorporated within the framework. Such a framework is trained based on the hardware and application configurations. Although ML approaches offer better detection rates compared to signature-based IDSs and enable the detection of unknown attacks, they may generate false alarms. The UGRansome dataset includes anomalous threats and serves as an anomaly detection dataset that can be used to study novel network attack behaviour to enhance network security[6].

2.7 | Enhancing IDS Proficiency Through Synthetic Signature Training

In the context of ML implementation, the synthetic signature (SS) category within the UGRansome dataset can serve as a valuable resource for training IDS to identify zero-day threats in both the signature (S) and anomaly (A) categories. This means that the SS category contributes to the comprehensive training of the IDS, enhancing its ability to recognise novel and previously unseen network threats. The incorporation of the SS category within the IDS training framework serves as a pivotal mechanism for augmenting the system's ability to identify and respond to both established and previously unobserved network threats.

This strategic inclusion empowers the IDS to transcend the constraints of solely recognising known threats, typified by the S category, and embrace the dynamic landscape of novel anomalies found in the A category. For instance, when confronted with an unprecedented threat, commonly referred to as a zero-day threat, the IDS can rely on the insights acquired from the SS category. This category is constructed to emulate the behavioral characteristics of diverse threats, known and unknown alike. In an ML analogy, it acts as a training ground for the IDS, allowing it to familiarise itself with the intricate patterns and subtleties of hitherto unencountered network threats. This enrichment of the IDS's training dataset facilitates its evolution into a more adept and versatile security guardian. By harnessing the training potential of the SS category, the IDS can further sharpen its proficiency in the realms of threat recognition and classification. It no longer restricts its vigilance to known threats but extends its capabilities to the early detection of emerging and unfamiliar anomalies, thereby reinforcing network security and ensuring resilience against the ever-evolving landscape of cybersecurity challenges.

2.8 | Active Network Intelligence

This section debates the application of an IDS to detect novel network attacks. The importance of using cyclostationary features to improve network security is also presented. An effort is made to define and introduce the IDS used, as well as its general impact in terms of the detection of novel network concerns. This discussion is linked to the main research question and is related to how an optimised anomaly detection dataset feeds into the necessity of improving the security of a network. The discussion is important in terms of the real time testing of the UGRansome dataset for zero-day vulnerabilities prevention. Intrusion detection solutions furnish scalable network intelligence technology, which is suitable for all types of networks, by analysing packets in the traffic[50]. The IDS allows network architects to gather subscriber experience intelligence to enforce policy on their networks[23]. The network topology, user license agreement, and system settings can impact the functionality delivered by any IDS. The typical deployment of a successful IDS will have the following properties:

- **Deep packet inspection**: Inspects IP packets and efficiently classifies their content using a data stream recognition definition language (DRDL)[23]. The DRDL analyses packets and presents them in a comprehensive and real-time manner based on the network traffic[23]. The IDS uses DPI as an active intelligence technique to examine computers, clients, and servers' identities by managing subscriber usage, monitoring the programs transmitting and receiving traffic (also known as active network intelligence services), and identifying the cyclostationary properties specific to each service, such as the sender/receiver in a packet flooding process[23]. The list of services identified by the DRDL is frequently updated. The network operator can also define network traffic recognition for specific services.

- **Policy engine**: Maps IP addresses to user identities in real time and provides the required infrastructure to manage and create subscriber profiles and active sessions[23]. The policy engine distributes the network user and session information to data plane elements to enforce applications and subscribers' policies.

- **Database**: Gathers traffic information from the network to build statistics based on the configurations and settings of statistics objects and rules. The statistics viewer object (SVO) explores the capabilities of the stored statistics features, which are presented in the form of charts and graphs[23].

- **Engineering insights**: Provides an interface to view the features stored in the database. Statistical data is extracted from network traffic running through the IDS. The data is then pushed to the insights data storage component. Engineering insights is a type of web-based visualisation tool for viewing information and cyclostationary features extracted from the network traffic[51]. The proposed methodology uses engineering insights to analyse the network traffic quality and trends.

The IDS deployment can be used for any purpose where it is useful to keep track of the traffic flow in a network. The following are examples of such purposes:

- **Reporting and analytics**: Examines network traffic features and represents the information in a simplistic view. Network operators can then gain a deeper comprehension of the data traffic. They can also manage the analysed traffic and impose various restrictions for policy enforcement[52].

- **Traffic management**: Includes fair split-shaping functionality, which enhances the subscriber network experience at a reduced cost[53].

- **Policy control**: Enforces restrictions on the network to reject, block, and drop malicious activities such as unwanted websites, ports, and IP addresses[23].

2.9 | About Deep Packet Inspection

Deep packet inspection (DPI) forms the core intelligence of the IDS and can be deployed on a variety of virtual platforms and hardware (Figure 8). The network intelligence component of DPI performs packet logic and flow inspection on IP packets and groups their content (Figure 9). The results from this ongoing packet analysis are presented in real time[52]. Using the software, users examine client and server identities by managing subscriber usage data and monitoring the applications transmitting and receiving traffic. DPI can identify the cyclostationary properties specific to each service, such as the sender or receiver of a packet. The graphic view represents the core feature of the network intelligence functionality. Admin users have the privilege of determining what modules or functionality are available to normal users. For instance, all traffic shaping functionality is invisible to users without explicit permission to configure rules set by the system admin. The system administrators can also configure the database permissions for normal users, and monitor the installation of licenses that are necessary to enforce exclusive policies, like channel sharing detection, content logic classification, and geolocation detection.

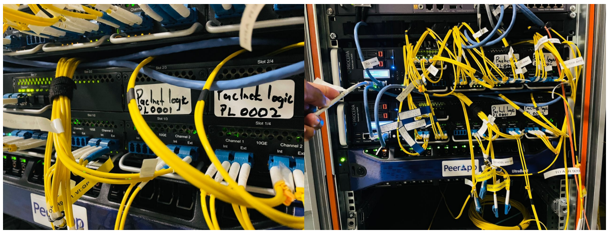
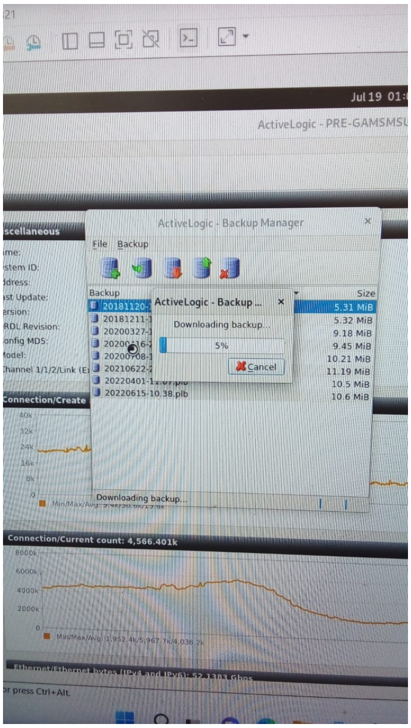


FIGURE 8 The DPI hardware component



```
Jul 19 01:
ActiveLogic - PRE-GAMSMSL

ActiveLogic - Backup Manager

scellaneous
me:
stem ID:
ddress:
ast Update:
ersion:
RDL Revision:
onfig MD5:
Model:
Channel 1/1/2/Link (E

Backup                                  Size
20181120-                           5.31 MiB
20181211-1                          5.32 MiB
20200327-1                          9.18 MiB
20200416-2                          9.45 MiB
20200708-1                         10.21 MiB
20210622-2                         11.19 MiB
20220401-11.07.plb                   10.5 MiB
20220615-10.38.plb                   10.6 MiB

ActiveLogic - Backup ...
Downloading backup...
5%

Downloading backup...

Connection/Create
Min/Max/Avg:

Connection/Current count: 4,566.401k
Min/Max/Avg: 1,952.4k/5,967.7k/4,038.2k

Ethernet/Ethernet bytes (IPv4 and IPv6): 52.1383 Gbps
or press Ctrl+Alt.
```

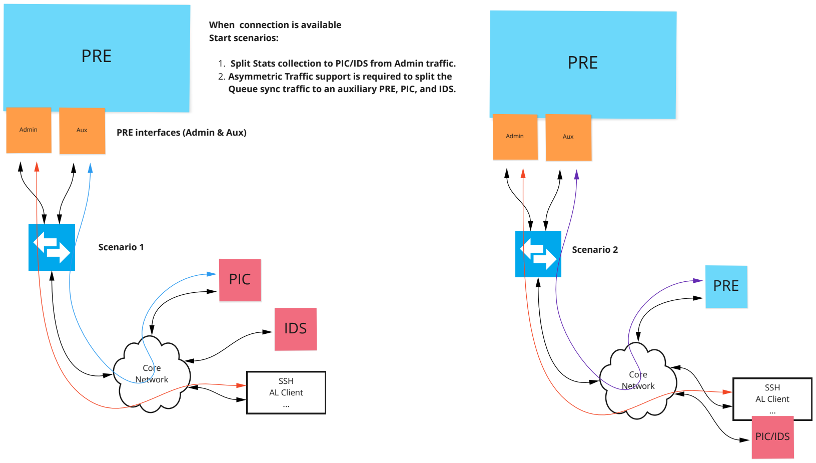
FIGURE 9 The DPI software component

2.10 | Basic DPI Data Flow

The basic flow of DPI data is presented in Algorithm 1. This algorithm represents the sequential steps involved in processing DPI data. The DPI system can stratify network traffic using subscriber devices and geolocations[8]. The network operator can monitor the subscriber's traffic activity in real time using the packet-logic rule enforcement (PRE) module (Figure 10). The PRE starts by splitting the collected statistics into the packet internet container (PIC) and IDS to support asymmetric traffic. It then uses the traffic via two interfaces (admin or aux). In some instances, asymmetric traffic is divided into a second PRE with additional PIC and IDS (Figure 10), where the PIC serves as the IDS database.

Algorithm 1 Basic flow of DPI data

```
1: Receive a packet.
2: Analyse the packet to determine the following:
       • Does the packet start a new connection, or does it belong to an existing connection?
       • Does the packet connection match any defined rules?
3: Enforce all rules to which the packet's connection applies.
4: If the packet has not been rejected or dropped during the enforcement of the rules, forward it.
```



When connection is available
Start scenarios:

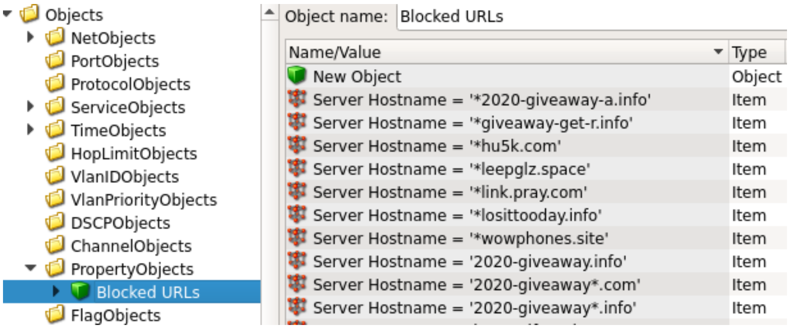
1. Split Stats collection to PIC/IDS from Admin traffic.
2. Asymmetric Traffic support is required to split the Queue sync traffic to an auxiliary PRE, PIC, and IDS.

FIGURE 10 The PRE module of the DPI

2.11 | Enforcing Rules

A rule is a condition set on the IDS to monitor and secure the network traffic[54]. The rule has various cyclostationary properties. A property object is usually created on the IDS to contain a text field that has the name of the cyclostationary feature. The network operator can then apply a condition to block, drop, or reject any traffic related to the cyclostationary feature[10]. A property object named blocked URLs is created in Figure 11 with a list of items or URLs assigned to a server. Any network traffic related to these URLs will be restricted to specific actions that can be set to reject, drop, rewrite, inject, divert, enrich, or accept the packets (Figure 12). The rule should be enabled to relinquish the action. It should contain advanced options to invoke the created property object. The system administrator will determine if additional rules can also be processed by the same action (Figure 12). The study intends to extract a list of cyclostationary features from the UGRansome dataset and create different network property objects to detect and block the network flow that uses malicious addresses, ports, and protocols.

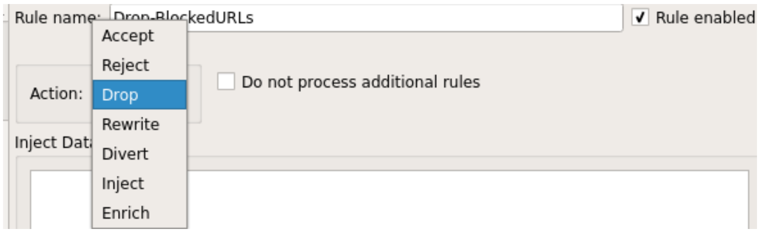
The blocked traffic will also be visualised by the IDS to detect abnormal flow and perform network traffic management. The rule implemented in figures 11 , 12 , and 13 can be read in Algorithm 2. The condition set in Figure 13 will drop all the website traffic, which uses the URLs configured in the property object.



Object name: Blocked URLs

Name/Value	Type
New Object	Object
Server Hostname = '*2020-giveaway-a.info'	Item
Server Hostname = '*giveaway-get-r.info'	Item
Server Hostname = '*hu5k.com'	Item
Server Hostname = '*leepglz.space'	Item
Server Hostname = '*link.pray.com'	Item
Server Hostname = '*losittoday.info'	Item
Server Hostname = '*wowphones.site'	Item
Server Hostname = '2020-giveaway.info'	Item
Server Hostname = '2020-giveaway*.com'	Item
Server Hostname = '2020-giveaway*.info'	Item

FIGURE 11 A property object with URLs

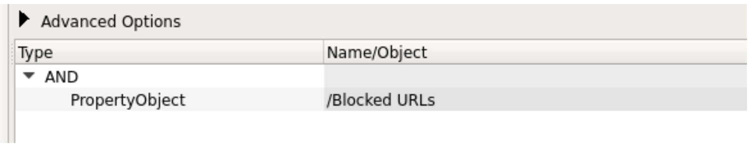

Rule name: Drop-BlockedURLs ☑ Rule enabled

Action: Drop

☐ Do not process additional rules

Inject Dat

FIGURE 12 Dropping rule


Advanced Options

Type	Name/Object
AND	
PropertyObject	/Blocked URLs

FIGURE 13 A property object condition

Algorithm 2 DPI data processing

```
Additional Rule:
IF Property Object == Blocked URLs WHEN URLs == Abnormal
AND Action == Drop
THEN Drop Network Traffic of URLs
ELSE Forward Network Traffic of URLs.
```

2.12 | Background Summary

We have discussed the concepts needed to detect and classify novel network threats by introducing the notion of cyclostationary features. These features were analysed and stratified based on their intrinsic properties, such as timestamp, latency, services, traffic, and sessions. This background section has defined cyclostationary features and discussed various approaches required to detect them. These features were grouped into connection, timestamps, protocol, ports, and addresses. The objective was to present the most cyclostationary features that will be useful in the optimisation of an IDS. Furthermore, the section has provided a taxonomy of cyclostationary features using legacy datasets such as KDD99 and NSL-KDD. The drawback of existing datasets was provided with their corresponding cyclostationary features. The abnormal network concerns were divided into signature, synthetic signature, and anomaly threats to facilitate the prediction and classification of novel network concerns. Additionally, the IDS was presented as an active network intelligence that performs DPI to test the UGRansome dataset in real time for zero-day vulnerabilities prevention. Lastly, this section demonstrated the usefulness of enforcing rules on the network using cyclostationary features to block or drop malicious traffic. The DPI can be thought of as an active IDS component that will be used to test the UGRansome dataset in real time and examine anomalous traffic identities. Section 3 will discuss the application of ensemble learning and RFE for the classification and prevention of novel network threats using the UGRansome dataset.

3 | METHODOLOGY

In 2021, Nkongolo et al.[6] introduced a significant contribution to the field of cybersecurity: the UGRansome dataset (Table 2 and Figure 17 -18). This dataset has received citations from Suthar et al. (2022)[55], Janicke, and Ferrag (2022)[56], Komisarek et al. (2023)[57], Gil Bravo[58], and Ramahlosi and Akanbi (2023)[59]. Furthermore, in their work, N. Zhang et al. (2023)[60] and M. Tokmak (2022)[22] discussed various evaluation models to assess the effectiveness of the UGRansome dataset. Rege and Bleiman (2023)[26] defined the UGRansome as a detection-based ransomware dataset, while Shankar, George, S, and Madhuri (2023)[42] stated that the UGRansome was designed to improve the detection of unknown network attacks, and was specifically created to include previously unexplored attacks (Figure 14). According to them[42], further research in network security should prioritise detecting unknown UGRansome attacks (Figure 15). What sets UGRansome apart from other datasets in the IDS landscape is its comprehensive coverage of previously unexplored ransomware attack types (Figure 16)[61]. Shankar, George, S, and Madhuri (2023)[42] therefore recommended the use of the UGRansome dataset while Singh et al. (2023)[62] reported that the UGRansome performed better in terms of classification accuracy when compared to previous datasets used in similar experiments or research. Lastly, Okafor et al. (2023)[63] delved into the application of the UGRansome, utilising a qualitative technique that employs an analytical case study method to explore the intricate realm of healthcare cybersecurity. This sector, frequently singled out for its invaluable and sensitive data, was the focal point of their investigation. Our research thus aims to contribute significantly to network security by utilising a stacking ensemble learning to further evaluate this dataset and enhance its capability to detect zero-day vulnerabilities in real time. The UGRansome has proven to be an invaluable resource for identifying and countering ransomware attacks, even those considered zero-day threats[42,55,64].

Within its dataset, it encompasses a spectrum of malware categories, including Signature (S), Anomaly (A), and Synthetic Signature (SS), with meticulously labeled instances of well-known ransomware variants such as Locky, CryptoLocker, JigSaw, EDA2, TowerWeb, Flyper, Razy, and WannaCry, as well as APT (Figure 14 , 15 , 16)[26]. Figure 17 depicts the three predictive classes (S, SS, and A) and their corresponding labels. Each predictive class is linked to specific ransomware types, including but not limited to Locky, CryptoLocker, APT, SamSam, and Globe. To delve deeper into the dataset's characteristics, we direct our attention to Table 2 , which provides a concise overview of its key attributes. The UGRansome dataset stands as a vital tool for researchers and cybersecurity professionals in the ongoing battle against ransomware threats within critical infrastructure. A ZIP file was acquired via download from the following URL: https://doi.org/10.13140/RG.2.2.23570.07363/1. This archive houses a dataset, consisting of 207 533 rows, stored in comma-separated values (CSV) format, albeit without any initial column headings (Figure 18). To facilitate further analysis, the dataset's headers were subsequently renamed by the specified attributes delineated in Table 2 , encompassing labels such as timestamp, protocol, flag, ransomware family, clusters, and more. To prepare the raw data for analysis, we employed a statistical approach to address issues such as data messiness and duplicate entries (Figure 18). Utilising the Python Data prep package and its comprehensive reporting function, which offers a thorough examination of the entire dataset and its variables, we obtained the following findings. As illustrated in Figure 18 (left side), no missing cells were identified, but a redundancy rate of 28.2% was observed. In response to this discovery, we proceeded to eliminate the duplicate entries, comprising a total of 58 491 rows.

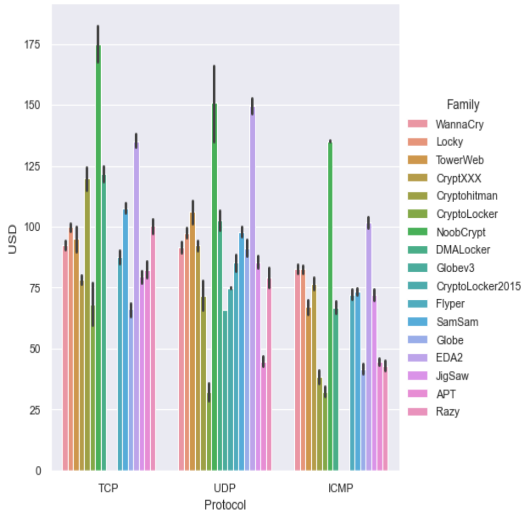


FIGURE 14 Ransomware financial impacts in the UGRansome dataset

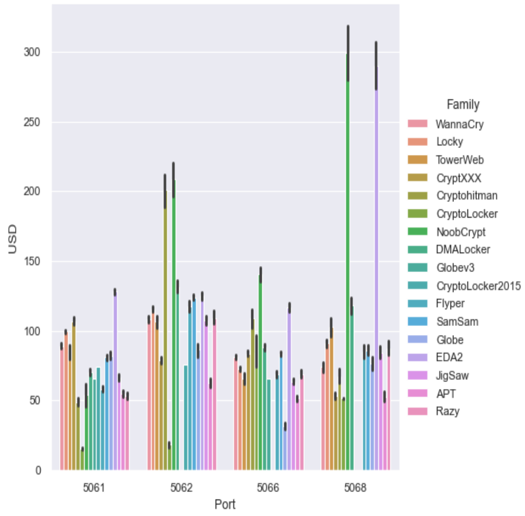


FIGURE 15 Ransomware ports in the UGRansome dataset

Subsequently, we re-evaluated the redundancy rate, as depicted in Figure 18 (right side), revealing that the cleaned dataset exhibited a 0.0% redundancy rate. This outcome indicated that the data was now prepared for processing and rigorous analysis. The resultant clean dataset, complete with column names, was then exported, encompassing 149 043 rows, making it ready for data processing and RFE. A series of mathematical transformations were implemented on these features (149 043 rows) to mitigate their skewed distributions, ultimately seeking to achieve either a normal distribution or a less-skewed distribution (Figure 19).

3.1 | Recursive Feature Elimination

RFE is a fundamental technique in data analysis, commonly employed to obtain a representative subset of data from a larger dataset[65]. In this context, Algorithm 3 illustrates the process of RFE utilised in this research. The objective is to randomly select 1 000 data points from the original UGRansome dataset, which can be particularly useful for various tasks, including data

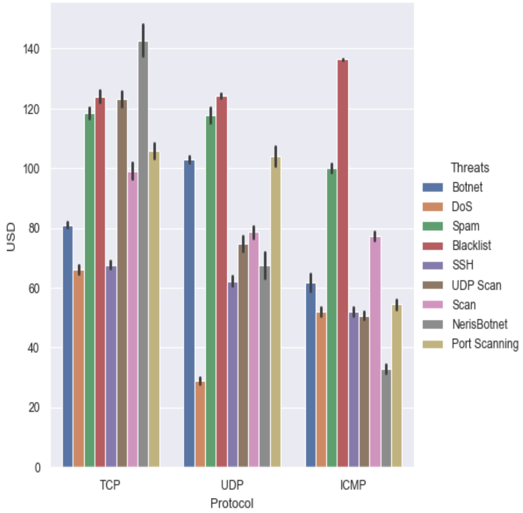


FIGURE 16 Financial impact of various threats in the UGRansome dataset

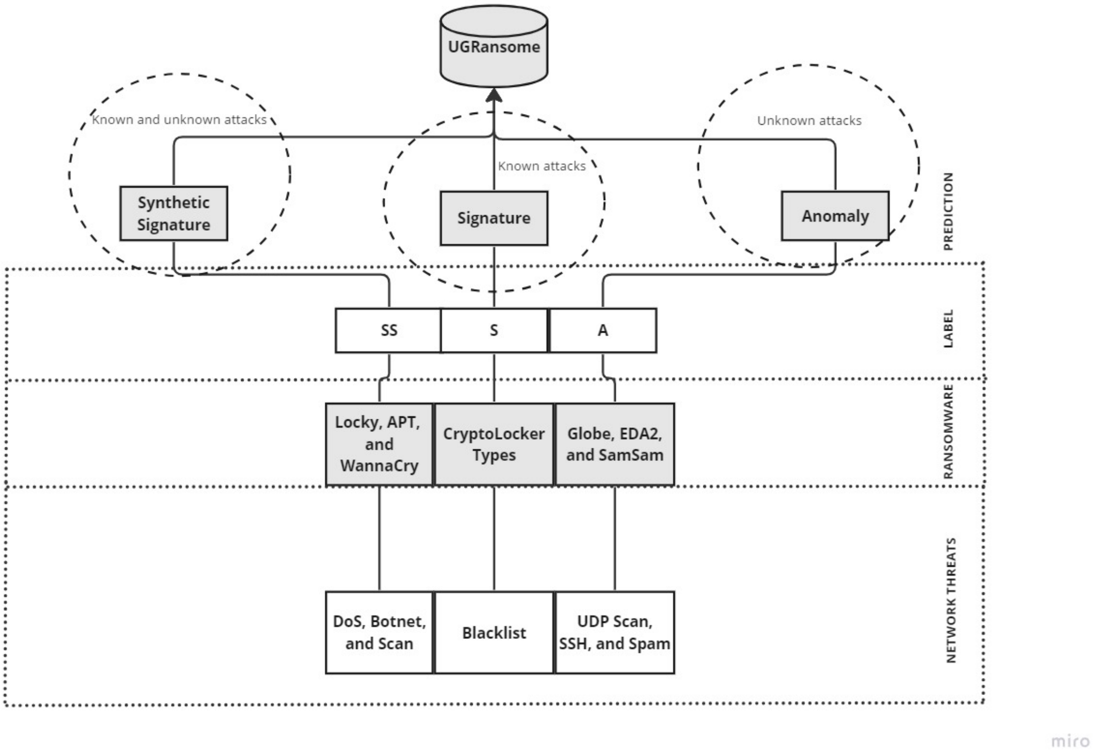
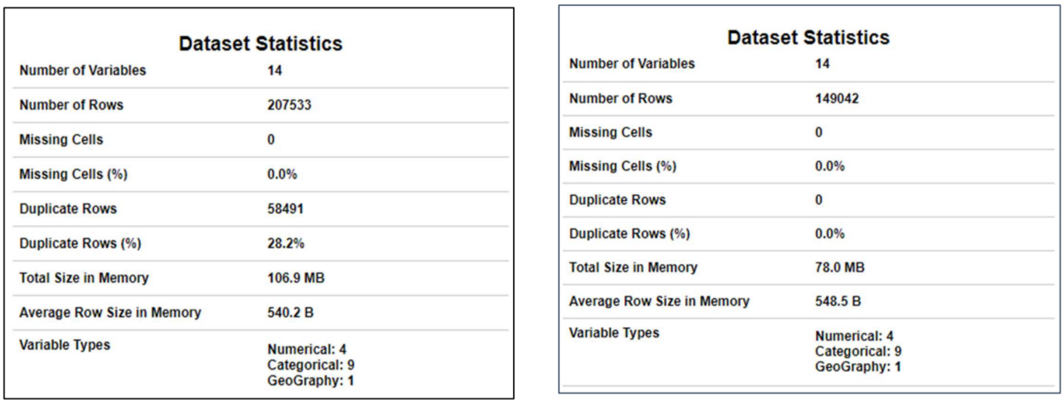


FIGURE 17 The UGRansome model

processing, model training, and statistical analysis. Let's denote the original feature set as X, the target variable as y, and the estimator as E. The goal is to select a subset of features F_{selected} containing k features as follows (Figure 20):

1. **Data splitting:** The dataset X is split into training and testing sets: $X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}}$.

2. **RFE initialisation:** RFE is initialised with the chosen estimator E and the specified number of features to select, k.



Dataset Statistics

Number of Variables	14
Number of Rows	207533
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	58491
Duplicate Rows (%)	28.2%
Total Size in Memory	106.9 MB
Average Row Size in Memory	540.2 B
Variable Types	Numerical: 4 Categorical: 9 GeoGraphy: 1

Dataset Statistics

Number of Variables	14
Number of Rows	149042
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	78.0 MB
Average Row Size in Memory	548.5 B
Variable Types	Numerical: 4 Categorical: 9 GeoGraphy: 1

FIGURE 18 The dataset statistics and insights

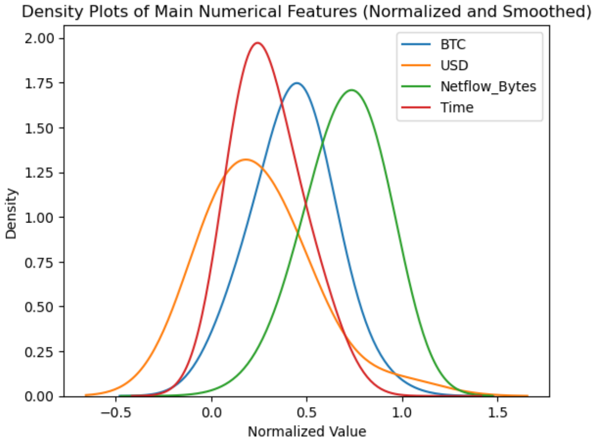


FIGURE 19 Normalised and smoothed numerical features

3. **RFE fitting:** RFE is fitted to the training data: $rfe.\text{fit}(X_{\text{train}}, y_{\text{train}})$.

4. **Feature selection:** The selected features are determined based on the support from RFE: $F_{\text{selected}} = X.\text{columns}[rfe.\text{support\_}]$.

5. **Data transformation:** The training and testing data are transformed to include only the selected features:

$$X_{\text{train\_selected}} = rfe.\text{transform}(X_{\text{train}})$$

$$X_{\text{test\_selected}} = rfe.\text{transform}(X_{\text{test}})$$

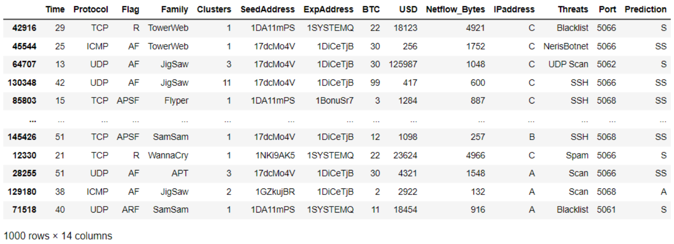
6. **Output:** The algorithm returns the selected features F_{selected} (Figure 22).

3.2 | Data Processing and Optimisation

After the RFE, the data was processed to transform categorical variables into numerical values to prepare it for ML computations (Figure 21). Duplicate values were removed (Figure 18), and any data entries with negative timestamps were also filtered out during this data preprocessing phase (Figure 19). This rigorous data preparation ensures the dataset is clean and suitable for the proposed ensemble learning.

TABLE 2 Attributes of the UGRansome dataset

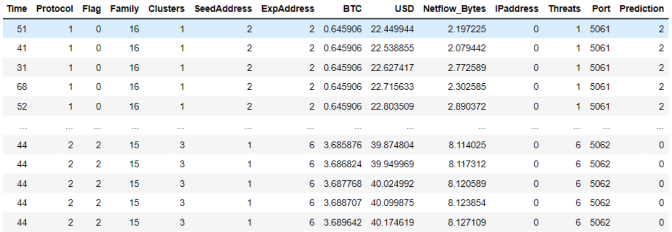
Attribute	Meaning	Type	Example
Time	Timestamp of network attacks	Numeric	50s
Protocol	Network protocol	Categorical	TCP
Flag	Connection status	Categorical	ACK
Family	Ransomware family	Categorical	WannaCry, JigSaw, SamSam, Locky
Clusters	Malware groups	Numeric	1-12
SeedAddress	Ransomware links	Categorical	1SYSTEMQ, 1GZkujBR, 1KZKcvx4, 1DA11mPS
ExpAddress	Ransomware links	Categorical	18e372GN, 1BonuSr7, 1DiCeTjB, 1AEoHYZ, 1NKi9AK5, 17dcMo4V, 1Lc7xTpP
BTC	Ransomware Bitcoin transactions	Numeric	90.0
USD	Ransomware USD transactions	Numeric	32 465
Netflow Bytes	Bytes transferred in network flow	Numeric	45 389
IP Address	IP addresses	Categorical	Class A
Threats	Malware	Categorical	Blacklist, SSH, Botnet, Spam, Scan
Port	Network port number	Numeric	5062
Prediction	Outcomes of predictive models (target variable)	Categorical	A, S, and SS



	Time	Protocol	Flag	Family	Clusters	SeedAddress	ExpAddress	BTC	USD	Netflow_Bytes	IPaddress	Threats	Port	Prediction
42916	29	TCP	R	TowerWeb	1	1DA11mPS	1SYSTEMQ	22	18123	4921	C	Blacklist	5066	S
45544	25	ICMP	AF	TowerWeb	1	17dcMo4V	1DiCeTjB	30	256	1752	C	NerisBotnet	5066	SS
64707	13	UDP	AF	JigSaw	3	17dcMo4V	1DiCeTjB	30	125987	1048	C	UDP Scan	5062	S
130348	42	UDP	AF	JigSaw	11	17dcMo4V	1DiCeTjB	99	417	600	C	SSH	5066	SS
85803	15	TCP	APSF	Flyper	1	1DA11mPS	1BonuSr7	3	1284	887	C	SSH	5068	SS
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145426	51	TCP	APSF	SamSam	1	17dcMo4V	1DiCeTjB	12	1098	257	B	SSH	5068	SS
12330	21	TCP	R	WannaCry	1	1NKi9AK5	1SYSTEMQ	22	23624	4966	C	Spam	5066	S
28255	51	UDP	AF	APT	3	17dcMo4V	1DiCeTjB	30	4321	1548	A	Scan	5066	SS
129180	38	ICMP	AF	JigSaw	2	1GZkujBR	1DiCeTjB	2	2922	132	A	Scan	5068	A
71518	40	UDP	ARF	SamSam	1	1DA11mPS	1SYSTEMQ	11	18454	916	A	Blacklist	5061	S

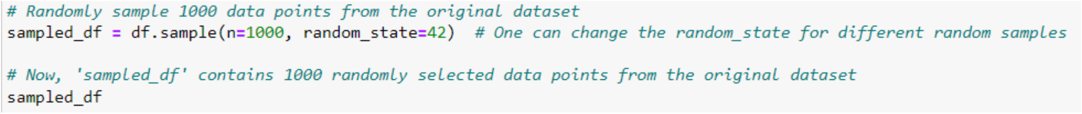
1000 rows × 14 columns

FIGURE 20 RFE results



Time	Protocol	Flag	Family	Clusters	SeedAddress	ExpAddress	BTC	USD	Netflow_Bytes	IPaddress	Threats	Port	Prediction
51	1	0	16	1	2	2	0.645906	22.449944	2.197225	0	1	5061	2
41	1	0	16	1	2	2	0.645906	22.538855	2.079442	0	1	5061	2
31	1	0	16	1	2	2	0.645906	22.627417	2.772589	0	1	5061	2
68	1	0	16	1	2	2	0.645906	22.715633	2.302585	0	1	5061	2
52	1	0	16	1	2	2	0.645906	22.803509	2.890372	0	1	5061	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
44	2	2	15	3	1	6	3.685876	39.874804	8.114025	0	6	5062	0
44	2	2	15	3	1	6	3.686824	39.949969	8.117312	0	6	5062	0
44	2	2	15	3	1	6	3.687768	40.024992	8.120589	0	6	5062	0
44	2	2	15	3	1	6	3.688707	40.099875	8.123854	0	6	5062	0
44	2	2	15	3	1	6	3.689642	40.174619	8.127109	0	6	5062	0

FIGURE 21 Encoded RFE results



```python
# Randomly sample 1000 data points from the original dataset
sampled_df = df.sample(n=1000, random_state=42)  # One can change the random_state for different random samples

# Now, 'sampled_df' contains 1000 randomly selected data points from the original dataset
sampled_df
```

FIGURE 22 Python RFE code

3.3 | Ensemble Learning

Ensemble learning combines predictions from multiple ML models to enhance overall performance[66]. It is a powerful technique for tackling complex problems.

Algorithm 3 RFE algorithm

```
Training dataset $X$, Target variable $y$, Estimator $E$ (RF), Number of features to select $k$, Selected features $F_{\text{selected}}$
Step 1: Split the dataset into training and testing sets $X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}} \leftarrow$ train_test_split($X, y, test\_size = 0.2$)
Step 2: Initialise the RFE with the estimator and the number of features to select $rfe \leftarrow$ RFE($E, n\_features\_to\_select = k$)
Step 3: Fit RFE on the training data $rfe$.fit($X_{\text{train}}, y_{\text{train}}$)
Step 4: Get the selected features $F_{\text{selected}} \leftarrow X$.columns[$rfe$.support_]
Step 5: Transform the training and testing data to include only the selected features $X_{\text{train\_selected}} \leftarrow rfe$.transform($X_{\text{train}}$)
$X_{\text{test\_selected}} \leftarrow rfe$.transform($X_{\text{test}}$)
Step 6: Output the selected features return $F_{\text{selected}}$
```

In our context, ensemble learning involves combining the strengths of the tree-based models such as RF and GB[67]. There are several types of ensemble learning methods, each with its approach to combining the individual models. Table 3 presents some common types of ensemble learning methods. This study uses the *stacking* ensemble approach.

3.4 | Proposed Stacked Ensemble Model

To achieve optimal zero-day threat detection, we propose a stack ensemble model that leverages the capabilities of gradient boosting (GB) and random forest (RF) as base models, with naive Bayes (NB) serving as the blender or meta-model. This stack ensemble model presented in Figure 23 is designed to extract valuable insights from each base model and make an informed prediction. The stacking approach involves training the base models independently and then combining their predictions using the NB meta-model. This hierarchical approach harnesses the strengths of each base model (RF and GB) while mitigating their weaknesses. The components of the framework illustrated in Figure 23 will be discussed in the upcoming sections. For our zero-day vulnerabilities detection system, we introduce a naive stack ensemble model that effectively leverages the capabilities of individual base models such as RF and GB to enhance prediction accuracy using NB as a meta-model (Figure 23)[71]. This ensemble model is designed to address the complexities of zero-day threat prevention by stacking two tree-based models. The ensemble learning will be evaluated with a confusion matrix (Table 4). The confusion matrix provides a detailed breakdown of the model's predictions, including true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP) (Table 4)[71].

TP are cases where the model correctly identifies attacks, whether they are anomalies (attacks without signatures), signature attacks (attacks with known signatures), or synthetic signature attacks (attacks with both known signatures and unknown components)[6]. In other words, TP represents the number of correct attack predictions. TN are cases where the model correctly identifies non-attacks, such as normal network behavior[6]. In this context, TN indicates that the model accurately recognises instances that do not attack, regardless of whether they involve anomalies, signature attacks, or synthetic signature attacks. TN represents correct non-attack predictions. FN occurs when the model mistakenly classifies attacks as non-attacks[6]. For instance, if an attack is misclassified as normal behavior, it would be considered a FN. In the context of anomaly, FN implies the failure to detect these stealthy attacks. For signature attacks, FN means failing to identify them as attacks. Similarly, for synthetic signature attacks, FN indicates that the model failed to recognise these complex attacks. FP represents cases where the model incorrectly predicts an attack when it is not an attack[6]. This can occur when the model misclassifies normal behavior as an attack. In the context of anomaly attacks, FP means incorrectly flagging benign activities as attacks. For signature attacks, FP implies the false identification of normal activities as malicious. In the case of synthetic signature attacks, FP means the incorrect classification of non-attacks as attacks. TP and TN reflect correct predictions, while FN and FP represent errors in the model's predictions[6]. These definitions apply to various types of attacks, including anomalies, signature-based attacks, and synthetic signature attacks, making them applicable in a cybersecurity context where different threat scenarios may occur[6]. The proposed ensemble learning will be evaluated using the following ML evaluation metrics:

- **Precision (P)**: measures the accuracy of positive predictions[71]. It is defined as:

$$P = \frac{TP}{TP + FP} \tag{3}$$

- **Accuracy (A)**: measures the overall correctness of predictions[71]. It is defined as:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

TABLE 3 Types of ensemble learning methods

Method	**Description**	**Examples**
Bagging	Combining predictions from multiple instances of the same base model trained.	RF
Boosting	Sequentially training weak learners, and combining their predictions.	GB, XGBoost
Stacking	combining base model predictions through a meta-model trained using the predictions of base models.	This work
Voting	Multiple models independently predict output, and the final prediction is based on a majority vote.	Hard/Soft voting
Random Subspace	Selects random subsets of features for each base model in the ensemble.	[68]
Random Patches	Selects random subsets of data instances and features for training each base model.	[69]
GB	Sequentially train models to correct errors of previous models by optimizing a loss function.	XGBoost
AdaBoost	Assigns different weights to training instances and base models.	AdaBoost
Bootstrapped Ensembles	Create subsets of data for each base model using bootstrapping techniques.	Bagging
Bayesian Model Averaging	Assigns a probability distribution over models and combines predictions.	[70]

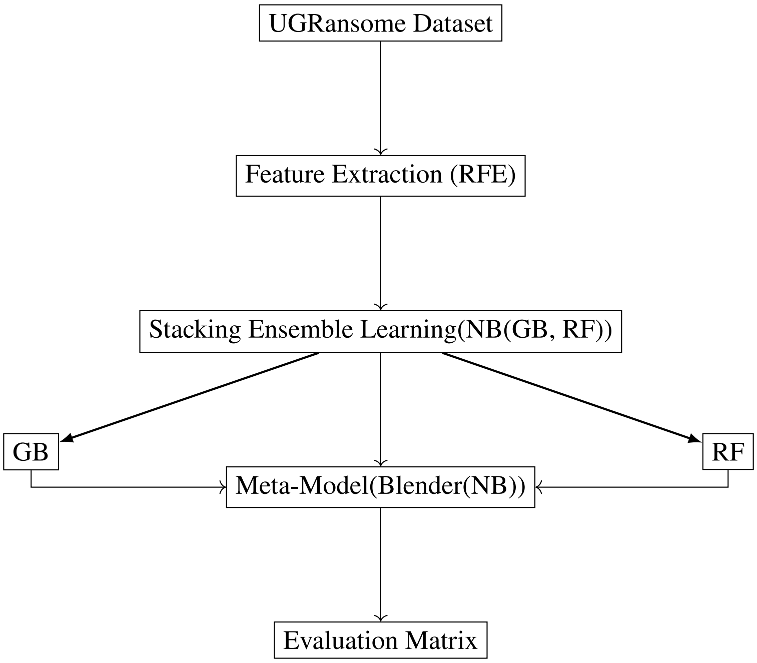


FIGURE 23 Proposed ML framework

TABLE 4 Confusion matrix

	Actual positive	**Actual negative**
Predicted positive (Outcome correct)	TP	FP
Predicted negative (Outcome correct)	FN	TN

- **F1 Score (F1)**: it is the harmonic mean of precision and recall[71]. It is defined as:

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5}$$

- **Recall (R)**: measures the model's ability to identify all relevant instances[71]. It is defined as:

$$R = \frac{TP}{TP + FN} \tag{6}$$

Meta-Model

To harmonise the predictions from our base models, we employ a meta-model, known as NB (Figure 23). The role of the meta-model is to combine the outputs of the base models, making the final prediction. Mathematically, the meta-model is defined as:

$$\text{NB}(X) = \frac{1}{n} \sum_{i=1}^{n} \text{Decision}_i(X) \tag{7}$$

The NB is the meta-model that blends the predictions of base models (RF and GB) by aiming to achieve a robust and accurate zero-day vulnerability recognition system (Figure 23).

Stacking with a tree-based ensemble learning: Stacking, or stacked generalisation, is an ensemble learning technique that combines multiple base models to improve predictive performance[72]. In the proposed tree-based ensemble learning, this approach involves the use of two algorithms as base models (Figure 23). We propose the following breakdown of our stacking ensemble learning presented in Figure 23 :

1. **Base models:** The process begins with a set of base models, typically tree-based models like GB and RF.

2. **Training phase:** In this phase, we train each of these base models on the same UGRansome dataset to predict the target variable (A, S, and SS) (Figure 17 and Table 2)[6].

3. **Meta-model:** In this process, we use an NB meta-model, often referred to as a blender, which naively takes the predictions from the base models as its input features[72].

4. **Meta-model training:** In this phase, we train the meta-model on these predictions to produce a final prediction. The NB learns to combine the outputs of the base models, effectively determining a weighted average of their predictions.

5. **Predictions:** When making predictions on new data, the proposed scheme uses the base models to predict the target variable. Then, feed these predictions into the meta-model to obtain the final prediction. Stacking leverages the strengths of individual base models, allowing them to work together to make more accurate final predictions. It is a potent technique for enhancing model performance, especially in situations involving complex datasets or challenging prediction tasks[73]. In the context of tree-based ensemble learning, stacking was highly effective as it combines the strengths of GB and RF, leading to superior predictive performance.

3.5 | Machine Learning

We present the selected ML models in this section. GB is an ensemble learning technique that builds a series of weak learners (usually decision trees) sequentially[6]. Each subsequent model corrects the errors of the previous one. The final model is a weighted sum of the individual weak models.

Mathematical representation: The loss function is minimised iteratively:

$$\text{Loss} = \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + h_m(x_i)) + \Omega(h_m)$$

Here, L is the loss function, y_i is the true label, $F_{m-1}(x_i)$ is the predicted value from the previous models, $h_m(x_i)$ is the new model's prediction, and $\Omega(h_m)$ is a regularisation term[74]. RF builds multiple decision trees and merges their predictions[6]. It introduces randomness by considering a random subset of features at each split, reducing overfitting.

Mathematical representation: RF involves constructing T decision trees:

$$\text{RF prediction} = \frac{1}{T} \sum_{t=1}^{T} \text{Decision Tree}_t(x)$$

NB is a probabilistic classification algorithm based on Bayes' theorem[6]. It assumes that features are conditionally independent given the class label.

Mathematical representation: The probability of a class given features $P(y|X)$ is calculated using Bayes' theorem:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Decision Trees (DTs) recursively split the dataset based on feature conditions, aiming to create homogeneous subsets in terms of the target variable[6].

4 | EXPERIMENTAL RESULTS

In this section, we commence by showcasing the outcomes of feature extraction through RFE and the subsequent ensemble learning results. The experimentation will proceed by employing the most pertinent features identified by the RFE to establish shaping and filtering rules, aiming to thwart zero-day vulnerabilities in real-time. Figure 24 illustrates feature importance as determined by the RFE Gini impurity[74]. In the context of feature importance, Gini impurity of the RFE quantifies how well a feature separates classes or categories within the UGRansome dataset[75]. Features that lead to better separation and lower impurity are considered more important as they contribute more to the decision-making process in the extraction tasks. The correlation matrix of the extracted features is visually represented in Figure 24 . It reveals a significant correlation coefficient of

0.26 between the ransomware cluster and predicted Bitcoins (BTC) transactions. This finding underscores a robust association between specific ransomware attack types and distinctive patterns within cryptocurrency transactions.

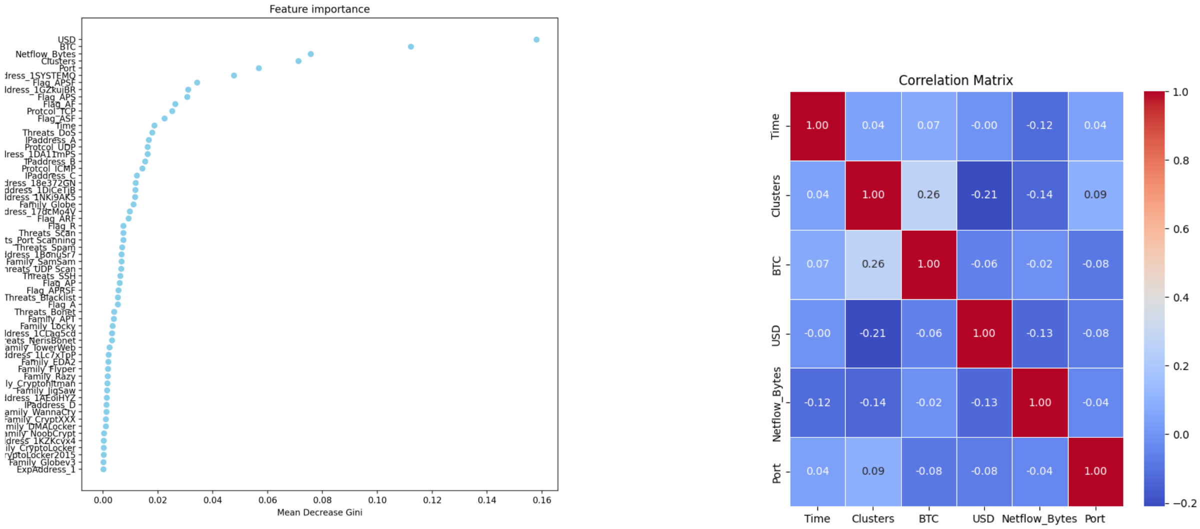


FIGURE 24 Relevant features and their correlations

4.1 | Machine Learning Results

We will showcase the ML results achieved by the selected base models, namely GB and RF. Furthermore, we will present the outcomes of the meta-model, NB. Lastly, we will unveil the results of our ensemble learning model. For each of these algorithms, we will provide comprehensive insights into the associated evaluation metrics, including precision, accuracy, F1 score, and recall. The confusion matrix of the proposed ensemble learning will also be presented. Table 5 provides a summary of the ML models utilised and their corresponding parameter configurations. It includes information on label encoding, the train-test split, and the specifications for each model, such as max depth, number of estimators, learning rate, and other relevant parameters. Additionally, it highlights the stacking classifier with its component models and the final estimator, NB.

TABLE 5 Model and parameter specifications

Model	Parameters
Encoding Train-Test Split	Label Encoding() Test Size: 20%
Decision Tree (DT) Random Forest (RF) Gradient Boosting (GB)	Max Depth: 5 Max Depth: 9, Estimators: 100, Max Features: 1 000 Estimators: 100, Learning Rate: 1.0, Max Depth: 1, Random State: 42
Stacking Classifier	**Estimators:** - RF (Max Depth: 9, Estimators: 100, Max Features: 1 000) - GB (Estimators: 100, Learning Rate: 1.0, Max Depth: 1, Random State: 42) **Final Estimator:** NB (BernoulliNB())

4.2 | Decision Tree Results

The DT model seems to be effective at identifying Signature attacks (S) and Synthetic Signature (SS) attacks with 98% and 100% of recall but may need some improvement in recognising Anomaly attacks (A), depending on the specific context and goals of the IDS application (Table 6).

TABLE 6 DT classifier report

	Precision	Recall	F1-Score	Support
Anomaly (A)	0.9848	0.8784	0.9286	74
Signature (S)	0.9467	0.9861	0.9660	72
Synthetic Signature (SS)	0.9153	1.0000	0.9558	54
Accuracy: 0.9500				200
Macro Avg	0.9489	0.9548	0.9501	200
Weighted Avg	0.9523	0.9500	0.9494	200

4.3 | Gradient Boosting Results

The DT model exhibits strong predictive performance compared to the GB model. It demonstrates high precision and recall for all three classes: Anomaly (A), Signature (S), and Synthetic Signature (SS) (Table 6). With an accuracy of 95%, the DT model excels in correctly classifying instances across these classes, yielding a well-balanced F1 score. However, in Table 7 , the GB model's results are somewhat less favorable. While it maintains a reasonable accuracy of 72.5%, its precision and recall values for the Anomaly (A) class are lower (63%-66%), indicating more false negatives and fewer true positives. This suggests that the GB model may be less effective at detecting zero-day attacks. On the other hand, it performs relatively well in identifying S and SS attacks. The DT model outperforms the GB model in terms of classification accuracy balanced precision, and recall across different attack types. The choice of model can significantly impact the effectiveness of a cybersecurity system in detecting and classifying different types of attacks.

TABLE 7 GB classifier report

Class	Precision	Recall	F1-Score	Support
Anomaly (A)	0.6912	0.6351	0.6620	74
Signature (S)	0.8125	0.9028	0.8553	72
Synthetic Signature (SS)	0.6346	0.6111	0.6226	54
Accuracy: 0.7250				200
Macro Avg	0.7128	0.7163	0.7133	200
Weighted Avg	0.7196	0.7250	0.7209	200

4.4 | Naive Bayes Results

The NB model outperforms both DT, RF, and GB in terms of accuracy, precision, and recall across all classes (Table 8). It achieves the highest accuracy (96%) and F1 scores, indicating a better overall classification performance (Table 8). The DT model also performs well but has slightly lower accuracy (95%). In contrast, the GB model exhibits lower precision and recall, particularly for the Anomaly (A) class. The NB model seems to be the most effective algorithm for classifying these types of attacks.

TABLE 8 NB classifier report

	Precision	**Recall**	**F1-Score**	**Support**
Anomaly (A)	0.9855	0.9189	0.9510	74
Signature (S)	0.9600	1.0000	0.9796	72
Synthetic Signature (SS)	0.9464	0.9815	0.9636	54
Accuracy: 0.9650				200
Macro Avg	0.9640	0.9668	0.9648	200
Weighted Avg	0.9658	0.9650	0.9647	200

4.5 | Ensemble Learning Results

The ensemble model shows the highest accuracy (97%) and competitive precision, recall, and F1 score values, making it the top-performing model among all the models (Table 9 and Figure 25). NB also performed well, closely following the ensemble model. DT and GB have lower accuracy and F1 scores, indicating that they are less effective for this classification task. The confusion matrix depicted in Figure 26 provides insights into the classification outcomes of attacks across three distinct categories: Anomaly (A), Signature (S), and Synthetic Signature (SS). The matrix showcases a remarkable level of accuracy, characterised by high TP rates across all categories, signifying the correct classification of the majority of attacks (Figure 26). Specifically, the Signature (S) and Synthetic Signature (SS) categories demonstrate outstanding precision, with minimal classification errors, featuring just one FP for the SS category (Figure 26). In contrast, the Anomaly (A) category, while generally accurate, shows a slightly elevated number of FN (2 and 3 instances), implying the misclassification of a few Anomaly (A) attacks as non-anomaly (Figure 26). These classification results are robust and dependable, particularly for the Signature (S) and Synthetic Signature (SS) attack types, encompassing 72 and 53 instances, respectively (Figure 26).

TABLE 9 Ensemble learning report

	Precision	**Recall**	**F1-Score**	**Support**
Anomaly (A)	0.93	0.99	0.96	70
Signature (S)	1.00	0.97	0.99	74
Synthetic Signature (SS)	0.98	0.95	0.96	56
Accuracy: 0.97				200
Macro Avg	0.97	0.97	0.97	200
Weighted Avg	0.97	0.97	0.97	200

The obtained results shed light on the effectiveness of selected ML models in recognising zero-day vulnerabilities threats. These threats are categorised as Anomaly (A), Synthetic Signature (SS), and Signature (S) (Figure 25 and Figure 26). Among the models evaluated, the proposed ensemble model, which combines the strengths of individual base models, showcased the highest accuracy of 97% (Figure 25). This exceptional performance signifies its proficiency in identifying the most challenging zero-day threats. Furthermore, the ensemble model exhibited outstanding precision and recall for all three threat categories, with F1 scores close to 1.0 for Signature attacks (S) (Table 9). NB, while not as accurate as the ensemble model, also achieved notable results.

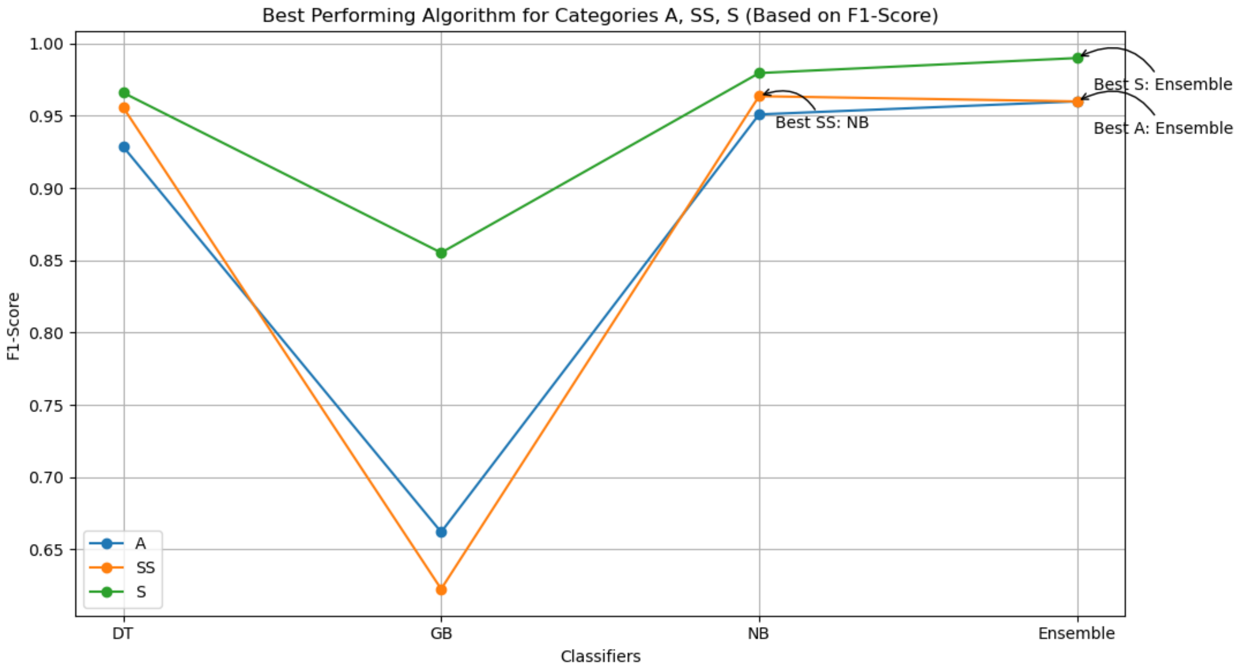


FIGURE 25 Algorithmic comparative analysis

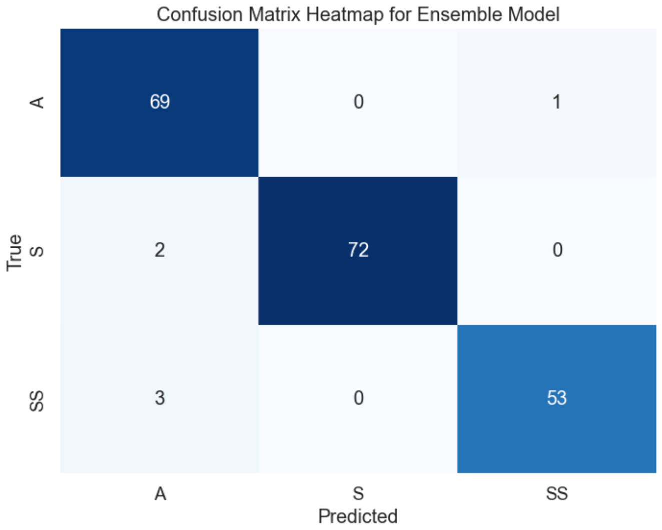


FIGURE 26 The ensemble learning confusion matrix

These findings imply that the combination of models in the ensemble and the advanced boosting techniques employed by GB is particularly well-suited for the task of recognising zero-day vulnerabilities threats. On the other hand, DT and GB showed lower accuracy and F1 scores, indicating that they may not be the ideal choice for tackling such sophisticated threats. This experiment underscores the significance of ensemble learning in enhancing the security posture against zero-day threats and illustrates the need for selecting relevant features for the task of effectively mitigating advanced cybersecurity challenges using ML models. In the upcoming section, we employ the most pertinent features identified by the RFE (Figure 24) to evaluate the UGRansome dataset in real-time.

4.6 | Intrusion Prevention Simulation

In the simulation experiments, the study ensured that the virtual private network (VPN) tunnel was properly configured and maintained to establish a secure connection between the computer and the IDS server. The VPN tunnel's performance and uptime were monitored to ensure a consistent and secure connection, minimising any potential risks or downtime. To establish a remote connection to the IDS, the following SSH command was used:

$$ssh - p\,42002\; ids@16.23.19.19$$

Here, the p flag denotes the specific port to be used (e.g. 42002), while 16.23.19.19 represents the server's IP address, and ids indicates the IDS server's name. The IDS server includes a DPI component that captures network traffic. The IDS server can be viewed as a complex system composed of several interconnected processes or daemons that work together to ensure the security of the network. Each daemon performs specific tasks, and they communicate with each other to extract and process information. The workflow used to test UGRansome in real time using the IDS is depicted in Figure 27 . This approach highlights the effectiveness of DPI with UGRansome data in improving network security by identifying and mitigating potential threats. The research designed a property object that includes extracted features of UGRansome such as seed or expended addresses, flags, and ports. By incorporating these features, the study creates a rule that can either reject or drop any abnormal traffic, reducing unknown traffic classification and ultimately improving network security. With this approach, the study could ensure that any malicious traffic utilising UGRansome features will be rejected, thus mitigating potential network threats. The research uses the IDS to visualise the traffic that triggered the rule set, offering real-time testing of UGRansome. In this experiment, the extracted UGRansome data is uploaded to the IDS file manager, where it can be processed and analysed to detect any malicious activity (Figure 28). As illustrated in Figure 29 , the study has generated a property object. The aforementioned property, labelled infected sites, encompasses all the seed or expended addresses contained in the UGRansome dataset, such as 1DA11mPS, 1diceyg, and 4ePEyktk (see Figure 29 where * means applied to all specified addresses). This meticulous approach serves to enhance network security by blocking all traffic that will use the specified UGRansome addresses in the property object. In the field of cybersecurity, it is crucial to have such mechanisms in place that can detect and prevent potential threats from entering a network. One such mechanism is the proposed use of rules to filter network traffic based on UGRansome addresses. In this case, the research utilised the seed or expended addresses of the UGRansome dataset to create a property object named infected sites (Figure 29).

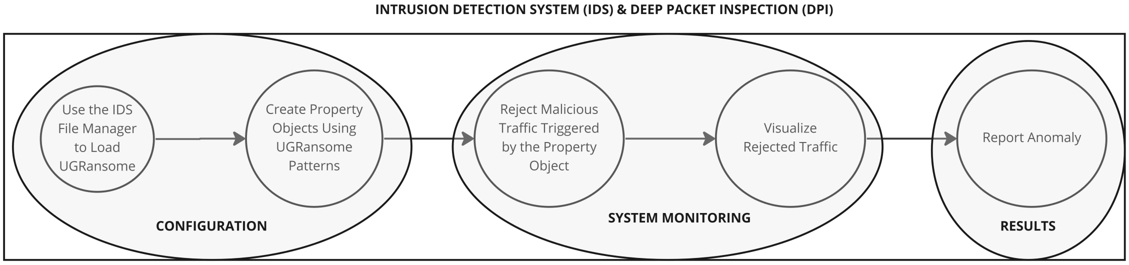
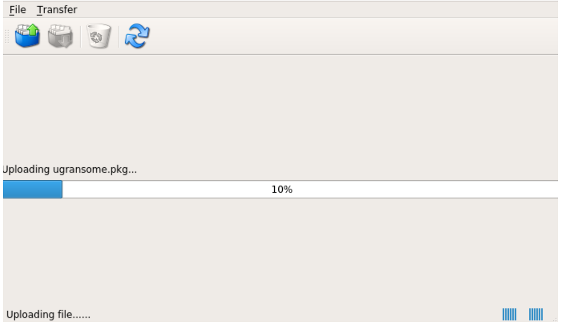

INTRUSION DETECTION SYSTEM (IDS) & DEEP PACKET INSPECTION (DPI)
Use the IDS File Manager to Load UGRansome
Create Property Objects Using UGRansome Patterns
CONFIGURATION
Reject Malicious Traffic Triggered by the Property Object
Visualize Rejected Traffic
SYSTEM MONITORING
Report Anomaly
RESULTS

FIGURE 27 Real time testing workflow using UGRansome

To further enhance network security, the research sets a rule to reject all network traffic that utilises UGRansome addresses, protocols, flags, and ports (Figure 30 and 32). This approach ensures that any malicious traffic that uses UGRansome features will be rejected, thereby mitigating potential threats to the network. The use of rules to filter network traffic based on certain criteria is a critical component of network security[76]. By incorporating features such as seed or expended addresses, protocols, flags, and ports, the research creates effective rules that help prevent potential threats from entering the network. To ensure the effectiveness of UGRansome in blocking malicious network concerns, the rejected traffic is subjected to visualisation for analysis, as shown in Figure 31 . The visualisation is crucial in determining the total traffic triggered by the rejection rule. Results obtained indicate that the total infected sites reached more than 100 kbps, with outgoing traffic surpassing incoming traffic. These findings are a clear indication of the efficiency of UGRansome in mitigating malicious network threats[76].

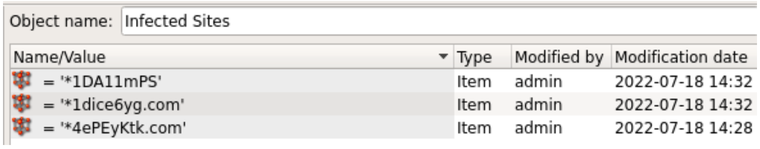


Jploading ugransome.pkg...

10%

Uploading file......

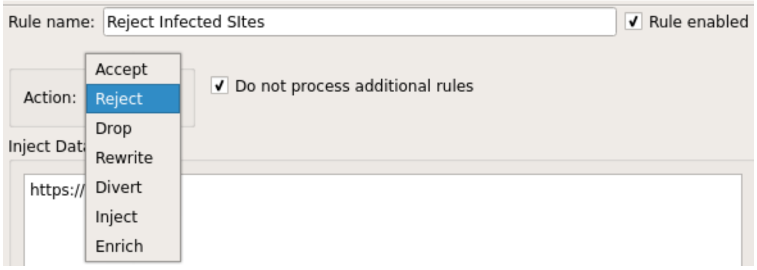
FIGURE 28 Uploading the UGRansome dataset in the IDS



Object name: Infected Sites

Name/Value	Type	Modified by	Modification date
= '*1DA11mPS'	Item	admin	2022-07-18 14:32
= '*1dice6yg.com'	Item	admin	2022-07-18 14:32
= '*4ePEyKtk.com'	Item	admin	2022-07-18 14:28

FIGURE 29 The UGRansome property object



Rule name: Reject Infected SItes ☑ Rule enabled

Action: Reject

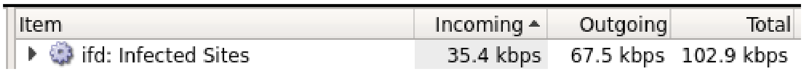
☑ Do not process additional rules

Inject Dat

https://

FIGURE 30 The rejection rule

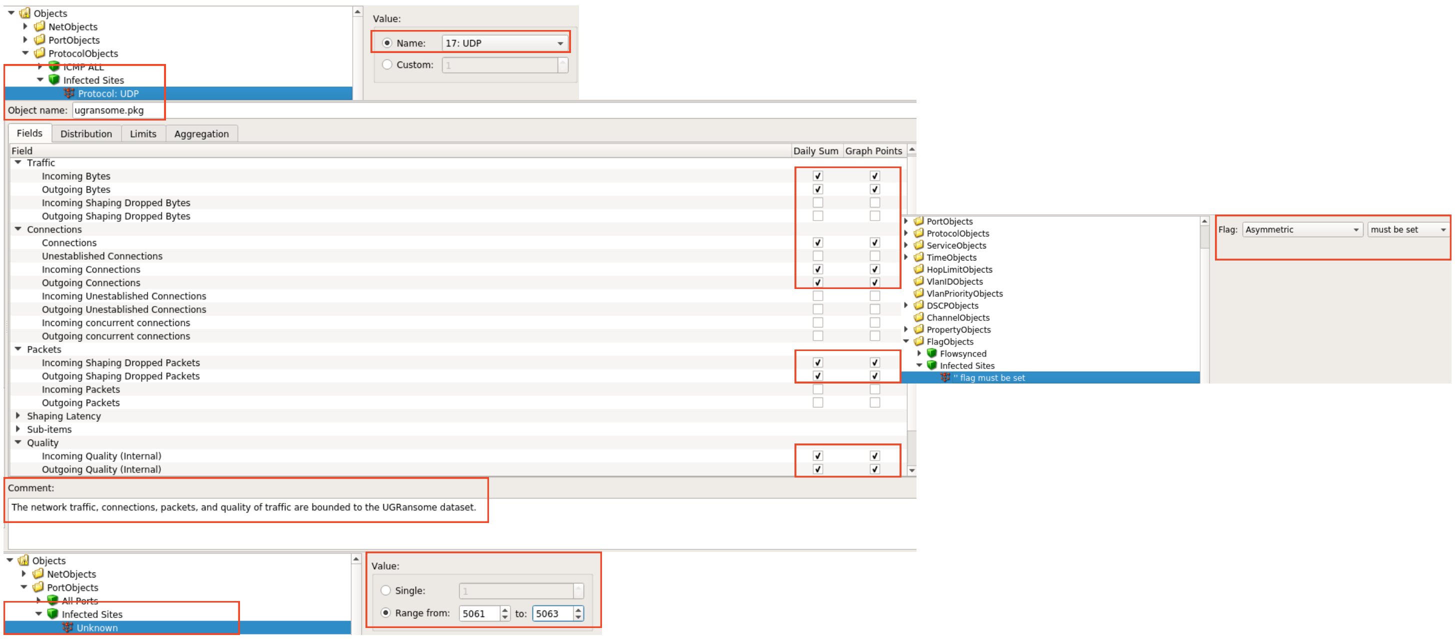
The IDS configuration using UGRansome is illustrated in Figure 32 . The system is configured with three network objects: protocol, port, and flag. The protocol object is designed to only reject UDP traffic, but it can also be configured to reject TCP or ICMP traffic. The IDS can block malicious traffic by utilising network objects to reject the flow that uses a seed or expended addresses of the UGRansome data (Figures 30 and 32). This discussion is related to the configuration and implementation of an IDS rule to detect and block malicious traffic related to UGRansome malware (Figure 32). The IDS uses DPI to analyse network traffic and identify packets that match the specified rule.



Item	Incoming	Outgoing	Total
ifd: Infected Sites	35.4 kbps	67.5 kbps	102.9 kbps

FIGURE 31 Rejected malicious traffic of infected sites

The rule is designed to reject traffic from specific IP addresses and ports associated with UGRansome malware, as well as any flags that indicate malicious behaviour (Figure 32). The rule is composed of several elements, including a property object named infected sites, which contains a list of known UGRansome seeds and expended addresses.



Value:
● Name: 17: UDP
○ Custom: 1

Object name: ugransome.pkg

Field	Daily Sum	Graph Points
Traffic		
Incoming Bytes	☑	☑
Outgoing Bytes	☑	☑
Incoming Shaping Dropped Bytes	☐	☐
Outgoing Shaping Dropped Bytes	☐	☐
Connections		
Connections	☑	☑
Unestablished Connections	☐	☐
Incoming Connections	☑	☑
Outgoing Connections	☑	☑
Incoming Unestablished Connections	☐	☐
Outgoing Unestablished Connections	☐	☐
Incoming concurrent connections	☐	☐
Outgoing concurrent connections	☐	☐
Packets		
Incoming Shaping Dropped Packets	☑	☑
Outgoing Shaping Dropped Packets	☑	☑
Incoming Packets	☐	☐
Outgoing Packets	☐	☐
Shaping Latency		
Sub-items		
Quality		
Incoming Quality (Internal)	☑	☑
Outgoing Quality (Internal)	☑	☑

Comment:
The network traffic, connections, packets, and quality of traffic are bounded to the UGRansome dataset.

Flag: Asymmetric must be set

Value:
○ Single: 1
● Range from: 5061 to: 5063

FIGURE 32 The IDS configuration using the UGRansome attributes

The action is set to reject, which instructs the IDS to drop any packets that match the rule (Figure 30). The protocol object is set to UDP to target the specific type of network protocol used by UGRansome (Figure 32). The port object is configured with the UGRansome malicious port ranges of 5061–5063 and 5065–5067 (Figure 32). The flag object is also set to asymmetric, which indicates that the packet could be altered or modified in some way, a common characteristic of malware traffic (Figure 32). Finally, the rule is limited to traffic that is bound to the ugransome.pkg file representing features extracted by the RFE technique (Figure 32). The effectiveness of the IDS rule is demonstrated by the visualisation of the rejected traffic, which shows 102.9 kbps of infected flow being blocked (Figure 31). This highlights the importance of UGRansome features in detecting and blocking malicious traffic on a network. It is important to note that the rule can also be configured to accept traffic if needed, as demonstrated in Figure 30 . This flexibility allows the IDS to be tailored to the specific needs and requirements of the network being protected. The upcoming section will detail the outcomes related to diverse shaping and filtering rules and the utilisation of UGransome to diminish unknown traffic.

4.7 | Shaping and Filtering Rules Results

The shaping and filtering rules are detailed in Tables 10 and 11 . Table 10 provides a breakdown and description of shaping rules as well as blocked malicious traffic. The effectiveness of the IDS in detecting network anomalies is further supported by Figure 33 . The incoming internal quality flow, in particular, does not show any significant fluctuations when compared to the other rules. Furthermore, there is a direct relationship between the number of established connections and the quality of the incoming internal traffic. This observation is indicative of the fact that the network traffic flow is being protected from any abnormalities. The IDS configuration, therefore, appears to be effective in maintaining the quality of the network traffic flow. Figure 33 presents a dashboard of the IDS configured with the proposed rule. The visualisation produced by the IDS indicates that the network traffic flow is being effectively monitored and that no abnormal traffic patterns have been detected. The results obtained from the IDS configuration suggest that the shaping rule is effective in maintaining the quality of the network traffic flow and protecting it from any abnormalities. The plot illustrated in Figure 34 shows the fluctuation of blocked traffic for different rules based on two zero-days vulnerabilities denoted as "A" and "SS". The x-axis represents various rules, including incoming, outgoing, total, estimated connections, unestablished connections, and incoming internal quality (Tables 11 and 10).

Upon examining the plot, we can discern the rules within shaping and filtering that exhibit the most blocked URLs and infected sites by assessing the tallest peaks for each category. Elevated peaks in the plot indicate rules that caused more blocked traffic

TABLE 10 Description of rules

Rule	Description
Incoming	Incoming traffic measured in kilobits per second (kbps)
Outgoing	Outgoing traffic measured in kilobits per second (kbps)
Total	Total of incoming and outgoing traffic
In. CPS	Incoming connections per service measured in kbps
Out. CPS	Outgoing connections per service measured in kbps
Est. connections	Established connections measured in kbps
Unest. connections	Unestablished connections measured in kbps
In int Quality	Quality of incoming internal traffic measured in kbps

TABLE 11 Description of features used to configure rules and blocked traffic (Figure 24)

Block URLs

Extracted RFE features: address, threats, flag, ransomware
1SYSTEMQ, APT, AP, expAddress_1, cryptoLocker2015, ARF, 1GZku-jBR, NoobCrypt, flag_R, 1KZKcvx4, DMALocker, flag_APS, CryptXXX, flag_AF, WannaCry, 1DA11mPS, JigSaw

Infected sites

Extracted RFE features: address and ransomware
18e372GN, SamSam, 1BonuSr7, Globe, 1DiCeTjB, Cryptoitman, 1AEoHYZ, Razy, 1NKi9AK5, Flyper, 17dcMo4V, EDA2, 1Lc7xTpP, Locky

(URLs or infected sites) for the specified zero-day vulnerabilities "A" and "SS". Determining the most blocked categories like Signature (S), Synthetic Signature (SS), and Anomaly (A) can be achieved by identifying the highest peaks within the S, SS, and A categories, and one can ascertain the rules contributing the most to blocked traffic. This plot highlights the dynamic nature of cybersecurity threats. Unknown and unpatched vulnerabilities, known as zero-day vulnerabilities, pose significant risks. The fluctuation in blocked traffic for the "A" and "SS" categories might signify varying attack intensities or frequencies targeting diverse zero-day vulnerabilities. These implications stress the importance of adaptive and proactive security measures. The plot underscores the necessity for continual monitoring, agile responses, and adaptive security frameworks to counter emerging threats. It emphasises the need for predictive analytics, machine learning, and artificial intelligence-driven security to effectively anticipate and mitigate potential zero-day vulnerabilities. In essence, this plot signifies the evolving nature of cybersecurity threats, demanding a holistic and adaptable approach to safeguard against emerging vulnerabilities, particularly zero-day threats, in an ever-changing landscape.

4.8 | Unexpected Anomaly and Unknown Traffic Reduction

During testing of UGRansome in real time, the study identified a packet drop anomaly where network traffic decreased upon uploading UGRansome data into the IDS to reject infected traffic (Figure 35). The root of this issue is unknown, and further investigation from a packet inspection perspective is needed to determine its cause. The author resolved this incompatibility issue by compressing and translating UGRansome into a pkg format, although some cyclostationary features were lost in the compression process. This study provides insight into the classification of novel network threats and the use of ML algorithms for this purpose. It also highlights the need for further investigation into packet drop anomalies presented in Figure 35 and the importance of preserving cyclostationary features during dataset compression. The study can be criticised for not delving deeper into the underlying causes of the identified anomalies, and for not exploring alternative approaches to resolve the packet drop issue. Additionally, the study's findings may not be generalisable to other datasets or contexts. The study has identified an anomaly in the network related to the rules that were not triggered, suggesting that there were no outgoing packet drops or latency issues in the network (Figure 36).

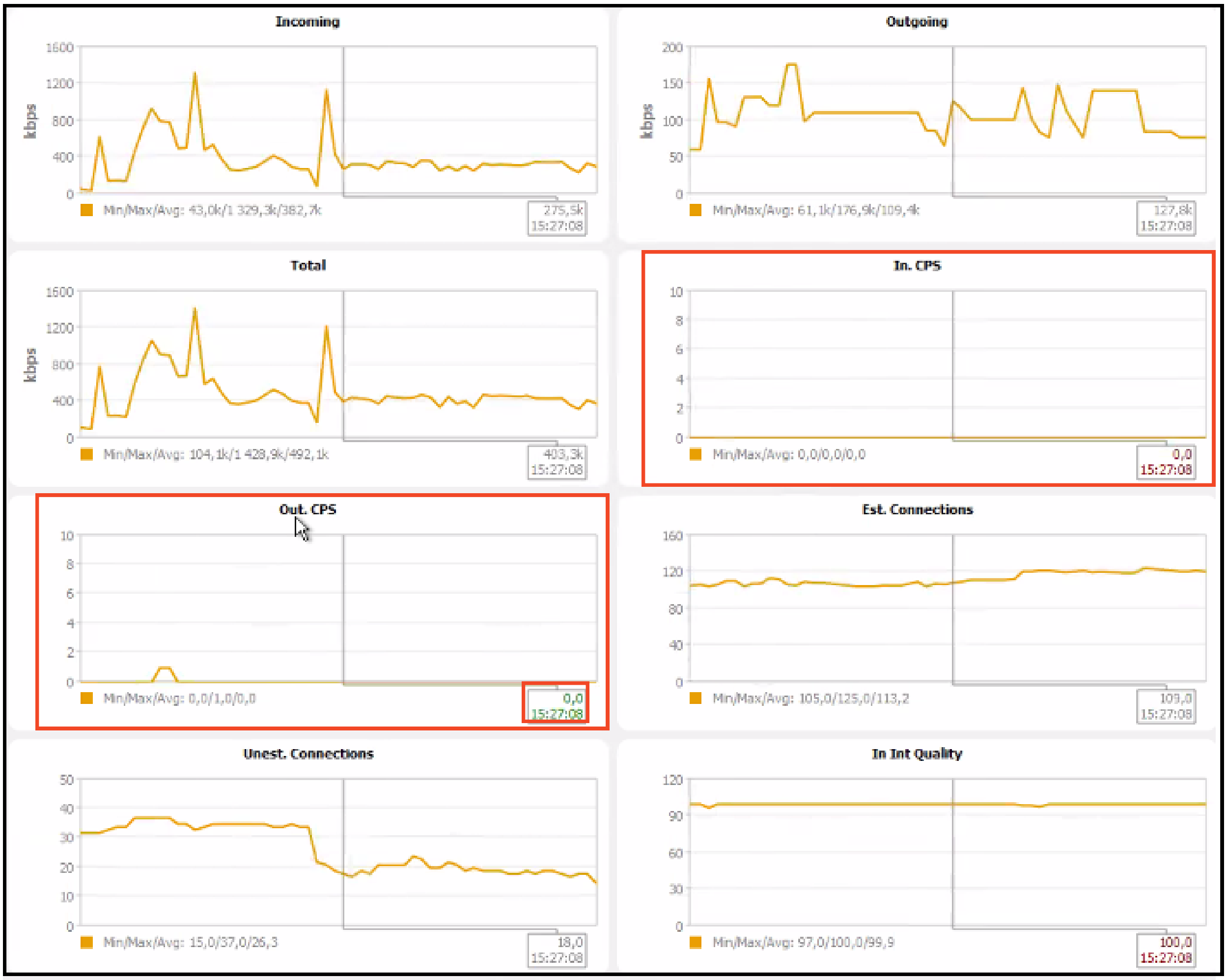


FIGURE 33 Real-time analysis of network flow using UGRansome (Table 10)

Moreover, the IDS generated a warning when UGRansome attributes were updated, as shown in Figure 37 . These warnings indicate that the DPI needs to manage more traffic queues than before, which could potentially impact its performance with packet drop shown in Figure 36 and Figure 38 . The DPI queue is a critical component of the IDS that manages the flow properties of network traffic in real time. The UGRansome dataset has been observed to cause anomalous changes to the traffic entries in the DPI queue (Figure 37). The warning message produced by the IDS indicates that the k-value store queue is full because the cache values are exhausted (Figure 38), which results in the loss of some flow properties (Figure 35 and Figure 36). This situation requires troubleshooting measures to adjust the key-value store parameters and increase the size of the queue. Further investigation is required to identify and analyse the factors that contribute to anomalous changes in the DPI queue. This research could also involve the development and testing of new algorithms and techniques to optimise IDS queue management and flow property storage. Moreover, the identification of anomalies is dependent on the quality and quantity of the RFE data (Figure 38 and Figure 37), as well as the algorithms and parameters used for analysis. Therefore, there is a need for further research to develop more robust methods for anomaly detection and investigate the impact of different parameters and algorithms on the performance of the IDS. Upon the implementation of rules utilising the properties of UGRansome, a notable reduction in unknown traffic was observed, as depicted in Figure 39 . It is noteworthy that the rate of unclassified traffic, i.e. traffic that was not being analysed, increased from 378.0 kbs to 591.0 kbs after the implementation of rules, as illustrated in Figure 39 . This unexpected outcome calls for further investigation to understand the underlying causes and implications of the observed phenomenon. Despite this, it can be argued that the reduction in unknown traffic is a significant advantage as it enhances the security of the network by enabling better monitoring and the detection of potential threats.

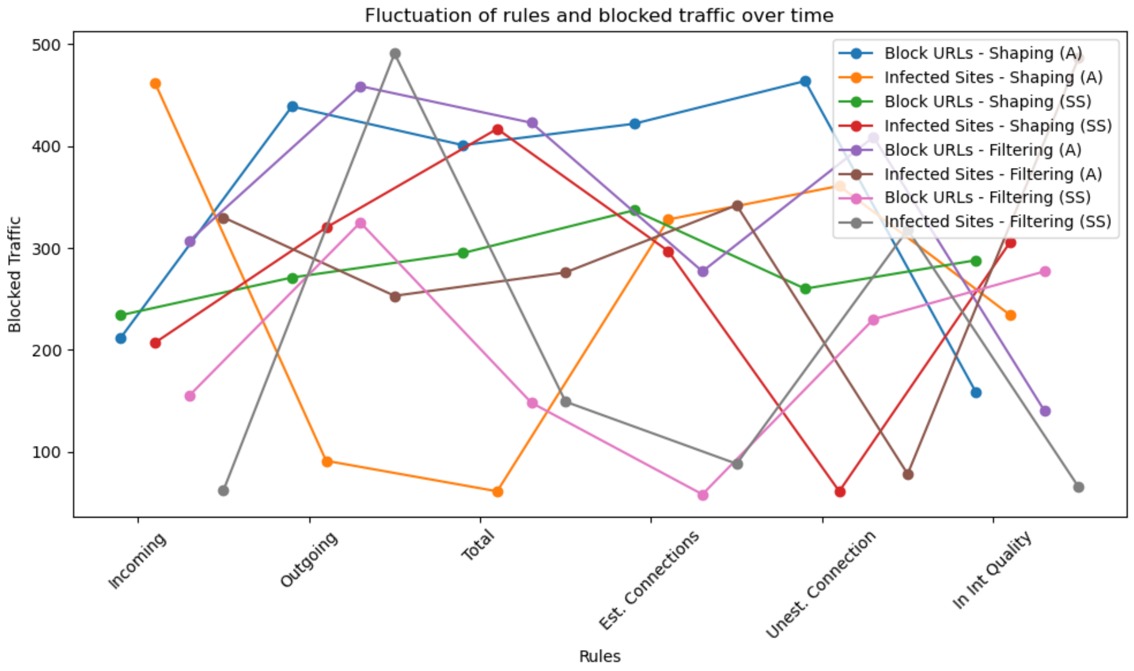


FIGURE 34 Network flow analysis using UGRansome

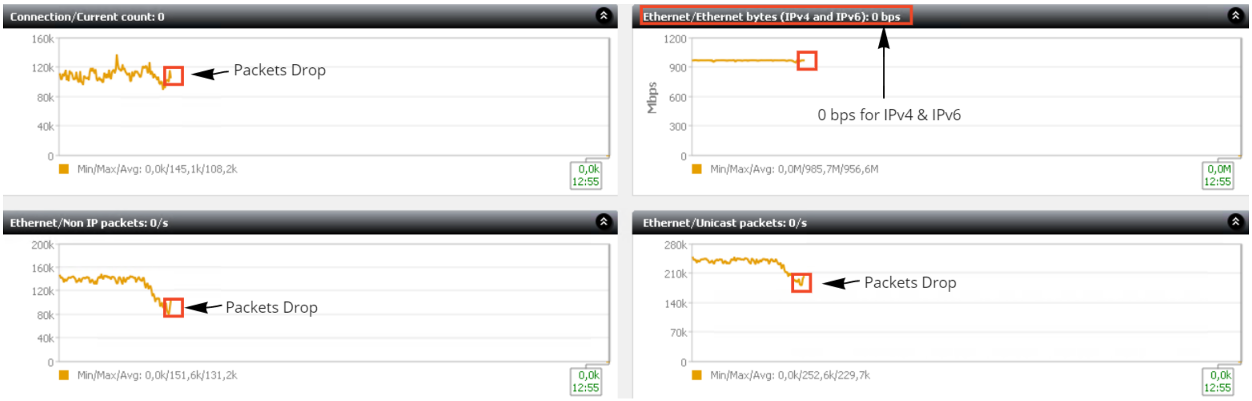

Connection/Current count: 0
160k
120k
80k
40k
0
Packets Drop
Min/Max/Avg: 0,0k/145,1k/108,2k
0,0k
12:55
Ethernet/Ethernet bytes (IPv4 and IPv6): 0 bps
1200
900
600
300
0
Mbps
0 bps for IPv4 & IPv6
Min/Max/Avg: 0,0M/985,7M/956,6M
0,0M
12:55
Ethernet/Non IP packets: 0/s
200k
160k
120k
80k
40k
0
Packets Drop
Min/Max/Avg: 0,0k/151,6k/131,2k
0,0k
12:55
Ethernet/Unicast packets: 0/s
280k
210k
140k
70k
0
Packets Drop
Min/Max/Avg: 0,0k/252,6k/229,7k
0,0k
12:55

FIGURE 35 The packet drop

Furthermore, the increase in unclassified traffic can potentially be addressed through the refinement of the implemented rules or the exploration of alternative methods. The research findings indicate a decrease in unknown traffic classification and an increase in unclassified traffic after the implementation of the UGRansome properties-based rule. This unexpected result highlights the need for further investigation into the relationship between unknown and unclassified traffic from a DPI perspective.

4.9 | Discussion and Comparative Experimentation

The Decision Tree (DT) model exhibits a commendable efficacy in accurately identifying Signature attacks (S) and Synthetic Signature (SS) attacks, achieving recall rates of 98% and 100%, respectively. This high recall signifies that the DT model successfully captures a vast majority of actual instances of these attack types. The Gradient Boosting (GB) model, although proficient, displays relatively lower precision and recall values for the Anomaly (A) class at 63%-66%.

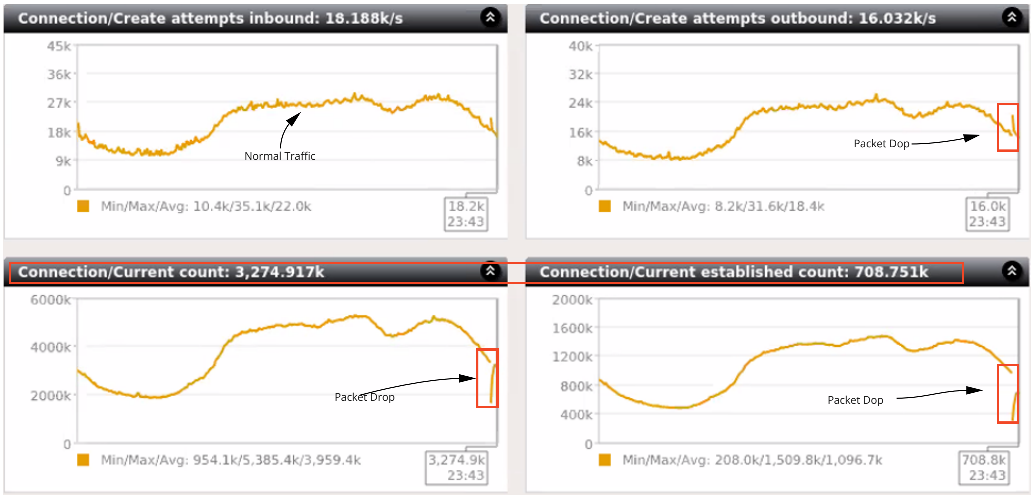
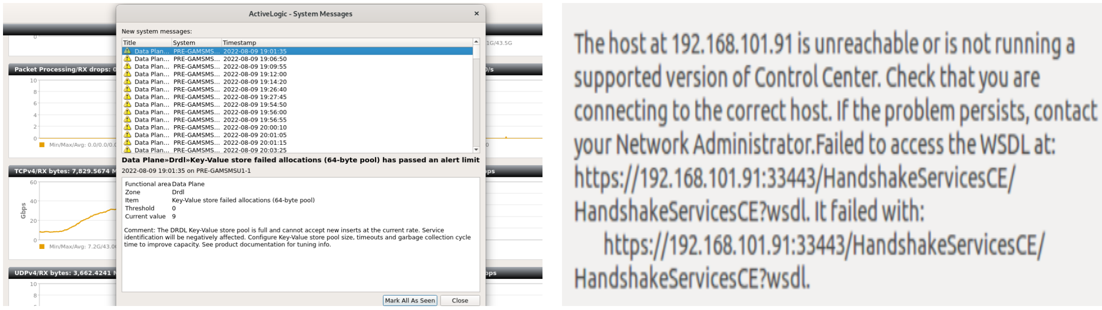


FIGURE 36 The packet drop and normal network traffic


Packet Processing/RX drops: 0
TCPv4/RX bytes: 7,829.5674 M
UDPv4/RX bytes: 3,662.4241 M
ActiveLogic - System Messages
Mark All As Seen
Close

New system messages:

Title	System	Timestamp
Data Plan...	PRE-GAMSMS...	2022-08-09 19:01:35
Data Plan...	PRE-GAMSMS...	2022-08-09 19:06:50
Data Plan...	PRE-GAMSMS...	2022-08-09 19:09:55
Data Plan...	PRE-GAMSMS...	2022-08-09 19:12:00
Data Plan...	PRE-GAMSMS...	2022-08-09 19:14:20
Data Plan...	PRE-GAMSMS...	2022-08-09 19:26:40
Data Plan...	PRE-GAMSMS...	2022-08-09 19:27:45
Data Plan...	PRE-GAMSMS...	2022-08-09 19:54:50
Data Plan...	PRE-GAMSMS...	2022-08-09 19:56:00
Data Plan...	PRE-GAMSMS...	2022-08-09 19:56:55
Data Plan...	PRE-GAMSMS...	2022-08-09 20:00:10
Data Plan...	PRE-GAMSMS...	2022-08-09 20:01:05
Data Plan...	PRE-GAMSMS...	2022-08-09 20:01:15
Data Plan...	PRE-GAMSMS...	2022-08-09 20:03:25

Data Plane»Drdl»Key-Value store failed allocations (64-byte pool) has passed an alert limit

2022-08-09 19:01:35 on PRE-GAMSMSU1-1

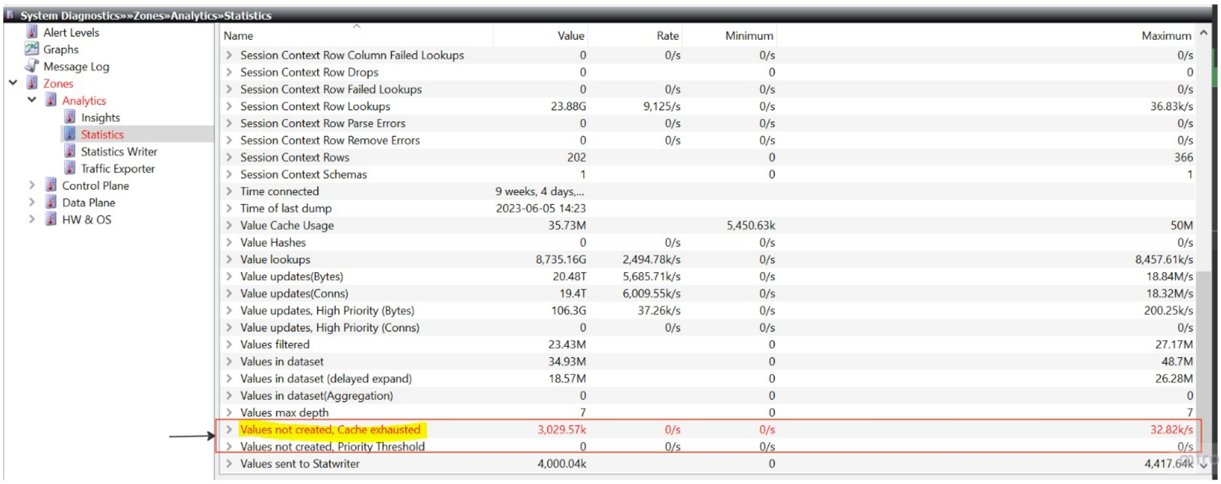
Functional area	Data Plane
Zone	Drdl
Item	Key-Value store failed allocations (64-byte pool)
Threshold	0
Current value	9

Comment: The DRDL Key-Value store pool is full and cannot accept new inserts at the current rate. Service identification will be negatively affected. Configure Key-Value store pool size, timeouts and garbage collection cycle time to improve capacity. See product documentation for tuning info.

The host at 192.168.101.91 is unreachable or is not running a supported version of Control Center. Check that you are connecting to the correct host. If the problem persists, contact your Network Administrator.Failed to access the WSDL at: https://192.168.101.91:33443/HandshakeServicesCE/HandshakeServicesCE?wsdl. It failed with:

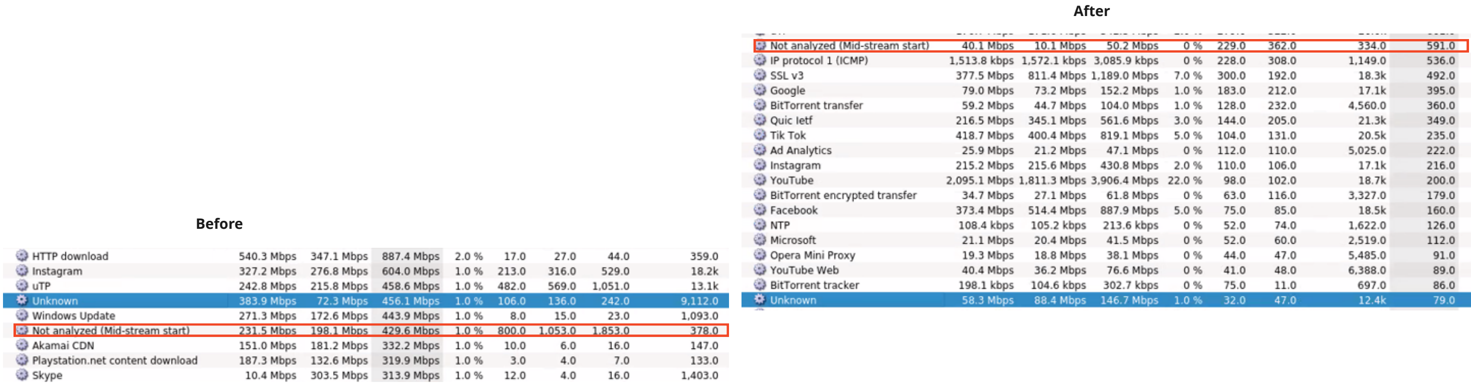
https://192.168.101.91:33443/HandshakeServicesCE/HandshakeServicesCE?wsdl.

FIGURE 37 Error caused by the UGRansome dataset



Name	Value	Rate	Minimum	Maximum
Session Context Row Column Failed Lookups	0	0/s	0/s	0/s
Session Context Row Drops	0		0	0
Session Context Row Failed Lookups	0	0/s	0/s	0/s
Session Context Row Lookups	23.88G	9,125/s	0/s	36.83k/s
Session Context Row Parse Errors	0	0/s	0/s	0/s
Session Context Row Remove Errors	0	0/s	0/s	0/s
Session Context Rows	202		0	366
Session Context Schemas	1		0	1
Time connected	9 weeks, 4 days,...			
Time of last dump	2023-06-05 14:23			
Value Cache Usage	35.73M		5,450.63k	50M
Value Hashes	0	0/s	0/s	0/s
Value lookups	8,735.16G	2,494.78k/s	0/s	8,457.61k/s
Value updates(Bytes)	20.48T	5,685.71k/s	0/s	18.84M/s
Value updates(Conns)	19.4T	6,009.55k/s	0/s	18.32M/s
Value updates, High Priority (Bytes)	106.3G	37.26k/s	0/s	200.25k/s
Value updates, High Priority (Conns)	0	0/s	0/s	0/s
Values filtered	23.43M		0	27.17M
Values in dataset	34.93M		0	48.7M
Values in dataset (delayed expand)	18.57M		0	26.28M
Values in dataset(Aggregation)	0		0	0
Values max depth	7		0	7
Values not created, Cache exhausted	3,029.57k	0/s	0/s	32.82k/s
Values not created, Priority Threshold	0	0/s	0/s	0/s
Values sent to Statwriter	4,000.04k		0	4,417.64k

FIGURE 38 The UGRansome's system diagnosis



Before

HTTP download	540.3 Mbps	347.1 Mbps	887.4 Mbps	2.0 %	17.0	27.0	44.0	359.0
Instagram	327.2 Mbps	276.8 Mbps	604.0 Mbps	1.0 %	213.0	316.0	529.0	18.2k
uTP	242.8 Mbps	215.8 Mbps	458.6 Mbps	1.0 %	482.0	569.0	1,051.0	13.1k
Unknown	383.9 Mbps	72.3 Mbps	456.1 Mbps	1.0 %	106.0	136.0	242.0	9,112.0
Windows Update	271.3 Mbps	172.6 Mbps	443.9 Mbps	1.0 %	8.0	15.0	23.0	1,093.0
Not analyzed (Mid-stream start)	231.5 Mbps	198.1 Mbps	429.6 Mbps	1.0 %	800.0	1,053.0	1,853.0	378.0
Akamai CDN	151.0 Mbps	181.2 Mbps	332.2 Mbps	1.0 %	10.0	6.0	16.0	147.0
Playstation.net content download	187.3 Mbps	132.6 Mbps	319.9 Mbps	1.0 %	3.0	4.0	7.0	133.0
Skype	10.4 Mbps	303.5 Mbps	313.9 Mbps	1.0 %	12.0	4.0	16.0	1,403.0

After

Not analyzed (Mid-stream start)	40.1 Mbps	10.1 Mbps	50.2 Mbps	0 %	229.0	362.0	334.0	591.0
IP protocol 1 (ICMP)	1,513.8 kbps	1,572.1 kbps	3,085.9 kbps	0 %	228.0	308.0	1,149.0	536.0
SSL v3	377.5 Mbps	811.4 Mbps	1,189.0 Mbps	7.0 %	300.0	192.0	18.3k	492.0
Google	79.0 Mbps	73.2 Mbps	152.2 Mbps	1.0 %	183.0	212.0	17.1k	395.0
BitTorrent transfer	59.2 Mbps	44.7 Mbps	104.0 Mbps	1.0 %	128.0	232.0	4,560.0	360.0
Quic Ietf	216.5 Mbps	345.1 Mbps	561.6 Mbps	3.0 %	144.0	205.0	21.3k	349.0
Tik Tok	418.7 Mbps	400.4 Mbps	819.1 Mbps	5.0 %	104.0	131.0	20.5k	235.0
Ad Analytics	25.9 Mbps	21.2 Mbps	47.1 Mbps	0 %	112.0	110.0	5,025.0	222.0
Instagram	215.2 Mbps	215.6 Mbps	430.8 Mbps	2.0 %	110.0	106.0	17.1k	216.0
YouTube	2,095.1 Mbps	1,811.3 Mbps	3,906.4 Mbps	22.0 %	98.0	102.0	18.7k	200.0
BitTorrent encrypted transfer	34.7 Mbps	27.1 Mbps	61.8 Mbps	0 %	63.0	116.0	3,327.0	179.0
Facebook	373.4 Mbps	514.4 Mbps	887.9 Mbps	5.0 %	75.0	85.0	18.5k	160.0
NTP	108.4 kbps	105.2 kbps	213.6 kbps	0 %	52.0	74.0	1,622.0	126.0
Microsoft	21.1 Mbps	20.4 Mbps	41.5 Mbps	0 %	52.0	60.0	2,519.0	112.0
Opera Mini Proxy	19.3 Mbps	18.8 Mbps	38.1 Mbps	0 %	44.0	47.0	5,485.0	91.0
YouTube Web	40.4 Mbps	36.2 Mbps	76.6 Mbps	0 %	41.0	48.0	6,388.0	89.0
BitTorrent tracker	198.1 kbps	104.6 kbps	302.7 kbps	0 %	75.0	11.0	697.0	86.0
Unknown	58.3 Mbps	88.4 Mbps	146.7 Mbps	1.0 %	32.0	47.0	12.4k	79.0

FIGURE 39 The reduction of unknown traffic classification

This lower precision indicates a higher occurrence of false negatives and fewer instances of true positives within this class[77]. Surprisingly, the Naive Bayes (NB) model surpasses both DT, Random Forest (RF), and GB models across all classes, exhibiting superior accuracy, precision, and recall. With an accuracy rate of 96% and F1 scores reaching 97%, the NB model demonstrates superior overall classification performance compared to its counterparts. Arguably, the NB model emerges as the most effective algorithm for effectively classifying zero-day vulnerabilities due to its exceptional performance across multiple metrics. Moreover, an ensemble model, combining the strengths of various models, showcases the highest accuracy at 97%. This ensemble model presents competitive precision, recall, and F1 scores, positioning it as the top-performing technique among the individual models (DT, RF, GB, and DT). The research findings unveil a noticeable trend following the implementation of the UGRansome properties-based rules. There is a discernible decrease in the classification of unknown traffic alongside a concurrent increase in unclassified traffic. This shift highlights the impact of the rule, potentially affecting the classification behaviour of the models and suggesting the need for further analysis to comprehend its implications fully. In essence, while individual models demonstrate distinct strengths and weaknesses, the NB model excels in classifying zero-day vulnerabilities, while the ensemble model emerges as the most accurate among the techniques explored in this study. The observed changes post-implementation of the UGRansome properties-based rule offer insights into the complexities of traffic classification, warranting continued investigation into its effects on model performance. A comparative analysis with existing studies is presented in Table 12 . Comparing our results to existing studies in Table 12 , it is evident that our NB model's performance exceeds the reported metrics in most studies. Similarly, our naive ensemble model outperforms a majority of existing techniques, indicating promising advancements in intrusion detection methodologies.

TABLE 12 Comparison of the proposed framework with existing studies

Studies	Dataset	Classifier	Metric	%
H. Zhang et al. (2018)[52]	UNSW-NB	Auto-encoder	Precision	95%
H. Liu and Lang (2019)[78]	KDD99	Particle Swarm Optimisation (PSO)	Precision	78%
Vinayakumar et al. (2019)[79]	WSN-DS	SVM	Recall	91%
Louati and Ktata (2020)[80]	KDDCup-99	SVM	Accuracy	99%
M. Nkongolo and M. Tokmak (2023)[18 6]	UGRansome	RF	ROC	100%
M. Tokmak (2022)[22]	UGRansome	Deep Forest	F1 score	88%
Xu, Xiong, Zhou, and Chen (2022)[81]	CAIDA	NB	Accuracy	80%
Ahmad et al. (2022)[82]	CICIDS2017	GB	Precision	75%
Le et al. (2022)[33]	NF-BoT-IoT-v2	Genetic Algorithm	Recall	87%

Continued on next page

TABLE 12 – continued from previous page

	Dataset	Classifier	Metric	%
Alzaqebah et al. (2022)[83]	UNSWNB-15	K-Nearest Neighbors (KNN)	Sensitivity	92%
G. Liu et al. (2022)[84]	WSN-DS	PSO	Accuracy	85%
M. Nkongolo and M. Tokmak (2023)[18]	UGRansome	Fuzzy Logic	Accuracy	98%
This work	UGRansome	NB Ensemble	F1 score	97%

Future research in IDSs, while poised for advancements, faces certain limitations. The implementation of advanced ensemble techniques and adaptive learning models may encounter challenges in computational complexity and resource demands, potentially limiting their scalability in real-time applications. Additionally, achieving explainable artificial intelligence in IDSs may pose hurdles in balancing interpretability with model complexity. Furthermore, the dynamic nature of cyber threats might pose difficulties in creating comprehensive threat models, potentially leading to gaps in threat coverage. Addressing these limitations will be critical in ensuring the practical applicability and efficacy of future IDS research. In our investigation, we found that Secure Shell (SSH) vulnerabilities stood out as the most impactful zero-day attacks. These zero-day vulnerabilities showed a penetration rate exceeding 500 Mbps specifically within the Anomaly (A) vulnerabilities, as depicted in Figure 40 .

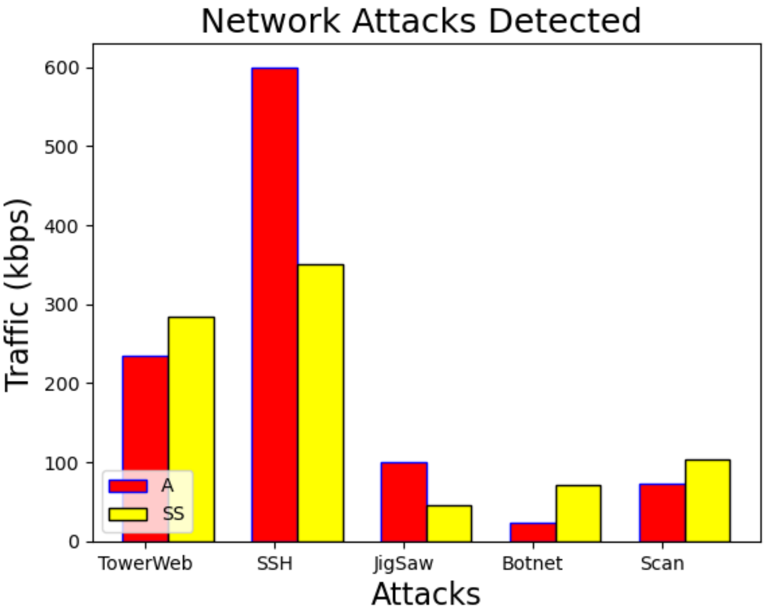


FIGURE 40 Attacks detected by the IDS using the UGRansome data

5 | CONCLUSION

The IDS field faces challenges in detecting unknown threats and minimising false alarms while combatting novel network attacks. Despite the rising prevalence of advanced threats and zero-day attacks, existing IDSs lack effectiveness due to inadequately designed datasets and the absence of innovative network shaping and filtering rules. Network shaping controls data flow speed or bandwidth, while filtering regulates data packet passage based on specific criteria such as source/destination IP addresses or port numbers for security purposes. An optimal IDS should incorporate diverse shaping and filtering rules to promptly block abnormal network issues. This article aims to develop an IDS empowered by innovative networking policies derived from the

UGRansome dataset to accurately identify and prevent unknown and new network threats in real time. Based on the findings, the study demonstrates varied performances among the models evaluated for intrusion detection. The Decision Tree (DT) model displays effectiveness in identifying Signature (S) and Synthetic Signature (SS) attacks but exhibits limitations in precision and recall for the Anomaly (A) class. The Naïve Bayes (NB) model surpasses other methods, showcasing superior accuracy, precision, and recall across all classes, particularly excelling in classifying zero-day vulnerabilities. The ensemble model emerges as the top-performing technique, showcasing high accuracy competitive precision, and recall values among all models assessed. Lastly, this research reveals noteworthy changes in traffic classification post-implementation of the UGRansome properties-based rule, emphasising potential enhancements in identifying network anomalies. These findings underscore the potential of the NB model and ensemble techniques in significantly improving intrusion recognition systems' efficiency, paving the way for more robust and effective strategies in combating emerging zero-day vulnerabilities.

6 | DATA AVAILABILITY

The data supporting the current study are available from https://doi.org/10.13140/RG.2.2.23570.07363/1.

7 | CONFLICTS OF INTEREST

The author declares that there are no conflicts of interest.

8 | ACKNOWLEDGMENTS

Open access funding enabled and organized by SANLiC Gold.

References

1. Waheed N, He X, Usman M. Security & Privacy in IoT Using Machine Learning and Blockchain: Threats & Countermeasures. *arXiv preprint arXiv:2002.03488* 2020.

2. Seniaray S, Jindal R. Machine Learning-Based Network Intrusion Detection System. In: Springer. 2022 (pp. 175–187).

3. Benavides E, Fuertes W, Sanchez S, Sanchez M. Classification of Phishing Attack Solutions by Employing Deep Learning Techniques: A Systematic Literature Review. In: Springer. 2020 (pp. 51–64).

4. Nkongolo M, Van Deventer JP, Kasongo SM, Van Der Walt W, Kalonji R, Pungwe M. Network Policy Enforcement: An Intrusion Prevention Approach for Critical Infrastructures. In: IEEE; 2022: 686–692.

5. Nisioti A, Mylonas A, Yoo PD, Katos V. From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Communications Surveys and Tutorials* 2018; 20(4): 3369–3388.

6. Nkongolo M, Deventer vJP, Kasongo SM. UGRansome1819: A Novel Dataset for Anomaly Detection and Zero-Day Threats. *Information* 2021; 12(10).

7. Jo W, Kim S, Lee C, Shon T. Packet Preprocessing in CNN-based network intrusion detection system. *Electronics* 2020; 9(7): 1151.

8. Otoum Y, Nayak A. As-ids: Anomaly and signature based ids for the internet of things. *Journal of Network and Systems Management* 2021; 29(3): 1–26.

9. Guo N, Tian Y, Li F, Yang H. Attention-based deep learning for network intrusion detection. In: . 11584. SPIE; 2020: 354–359.

10. Ferdiana R, others . A systematic literature review of intrusion detection system for network security: research trends, datasets, and methods. In: IEEE; 2020: 1–6.

11. Kumar I, Mohd N, Bhatt C, Sharma SK. Development of IDS using supervised machine learning. In: Springer. 2020 (pp. 565–577).

12. Ayyagari MR, Kesswani N, Kumar M, Kumar K. Intrusion detection techniques in network environment: A systematic review. *Wireless Networks* 2021; 27(2): 1269–1285.

13. Thakkar A, Lohiya R. Attack classification using feature selection techniques: a comparative study. *Journal of Ambient Intelligence and Humanized Computing* 2021; 12(1): 1249–1266.

14. Dasari DB, Edamadaka G, Chowdary C, Sobhana M, others . Anomaly-based network intrusion detection with ensemble classifiers and meta-heuristic scale (ECMHS) in traffic flow streams. *Journal of Ambient Intelligence and Humanized Computing* 2021; 12(10): 9241–9268.

15. Ribeiro VHA, Reynoso-Meza G. Ensemble learning by means of a multi-objective optimization design approach for dealing with imbalanced data sets. *Expert Systems with Applications* 2020; 147: 113232.

16. Mehboob U, Qadir J, Ali S, Vasilakos A. Genetic algorithms in wireless networking: techniques, applications, and issues. *Soft Computing* 2016; 20(6): 2467–2501.

17. Kornyo O, Asante M, Opoku R, et al. Botnet Attacks Classification in AMI Networks with Recursive Feature Elimination (RFE) and Machine Learning Algorithms. *Computers & Security* 2023: 103456.

18. Nkongolo M, Tokmak M. Zero-Day Threats Detection for Critical Infrastructures. In: Gerber A, Coetzee M., eds. *South African Institute of Computer Scientists and Information Technologists*Springer Nature Switzerland; 2023; Cham: 32–47.

19. Dang QV, Vo TH. Studying the Reinforcement Learning Techniques for the Problem of Intrusion Detection. In: IEEE; 2021: 87–91.

20. Abdelrahman AM, Rodrigues JJ, Mahmoud MM, et al. Software-defined networking security for private data center networks and clouds: Vulnerabilities, attacks, countermeasures, and solutions. *International Journal of Communication Systems* 2021; 34(4): e4706.

21. Azeez NA, Bada TM, Misra S, Adewumi A, Vyver V. dC, Ahuja R. Intrusion Detection and Prevention Systems: An Updated Review. In: Springer. 2020 (pp. 685–696).

22. Tokmak M. Deep forest approach for zero-day attacks detection. *Innovations and Technologies in Engineering*. 2022(ISBN: 978-625-6382-83-1): pp. 45-56.

23. Quintero-Bonilla S, Rey M. dA. A new proposal on the advanced persistent threat: A survey. *Applied Sciences* 2020; 10(11): 3874.

24. Banisar D. National Comprehensive Data Protection/Privacy Laws and Bills.. *SSRN* 2023.

25. Jordan Z. The Effect of the European Union (EU) General Data Protection Regulation (GDPR) on the Gaming Industry. *UNLV Gaming LJ* 2020; 10: 259.

26. Rege A, Bleiman R. A Free and Community-Driven Critical Infrastructure Ransomware Dataset. In: Onwubiko C, Rosati P, Rege A, et al., eds. *Proceedings of the International Conference on Cybersecurity, Situational Awareness, and Social Media*Springer; 2023: 25–37.

27. Citron DK. Privacy Injunctions. *Emory LJ* 2021; 71: 955.

28. Pradhan M, Nayak CK, Pradhan SK. Intrusion Detection System (IDS) and Their Types. In: IGI Global. 2020 (pp. 481–497).

29. Camacho J, Therón R, García-Giménez JM, Maciá-Fernández G, García-Teodoro P. Group-Wise Principal Component Analysis for Exploratory Intrusion Detection. *IEEE Access* 2019; 7: 113081–113093.

30. Liu H, Chen J, Ding Z, Ni G, He C, Jin R. Direction Finding with Cyclostationarity Analysis Against Frequency Interference. *Journal of Communications and Information Networks* 2022; 7(1): 88–95.

31. Birch K, Cochrane D, Ward C. Data as asset? The measurement, governance, and valuation of digital personal data by Big Tech. *Big Data and Society* 2021; 8(1): 20539517211017308.

32. Balon-Perin A. Ensemble-based methods for intrusion detection. Master's thesis. Institutt for Datateknikk og Informasjonsvitenskap. 2012.

33. Le TTH, Kim H, Kang H, Kim H. Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method. *Sensors* 2022; 22(3): 1154.

34. Akhi AB, Kanon EJ, Kabir A, Banu A. Network Intrusion Classification Employing Machine Learning: A Survey. 2019.

35. Ring M, Wunderlich S, Grüdl D, Landes D, Hotho A. Creation of flow-based data sets for intrusion detection. *Journal of Information Warfare* 2017; 16(4): 41–54.

36. Patil S. *Network Intrusion Detection System using Ensemble Learning*. PhD thesis. Dublin, National College of Ireland, 2020.

37. Paquet-Clouston M, Haslhofer B, Dupont B. Ransomware payments in the bitcoin ecosystem. *Journal of Cybersecurity* 2019; 5(1): tyz003.

38. Hindy H, Brosset D, Bayne E, et al. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE Access* 2020; 8: 104650–104675.

39. Deka RK, Bhattacharyya DK, Kalita JK. Active learning to detect DDoS attack using ranked features. *Computer Communications* 2019; 145: 203–222.

40. Kasongo SM, Sun Y. A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE access* 2019; 7: 38597–38607.

41. Siddique K, Akhtar Z, Khan FA, Kim Y. Kdd cup 99 data sets: A perspective on the role of data sets in network intrusion detection research. *Computer* 2019; 52(2): 41–51.

42. Shankar D, George GVS, S JNJNS, Madhuri PS. Deep Analysis of Risks and Recent Trends Towards Network Intrusion Detection System. *International Journal of Advanced Computer Science and Applications* 2023; 14(1).

43. Ahsan M, Rifat N, Chowdhury M, Gomes R. Detecting Cyber Attacks: A Reinforcement Learning Based Intrusion Detection System. In: IEEE; 2022: 461–466.

44. Gaurav A, Gupta BB, Panigrahi PK. A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system. *Enterprise Information Systems* 2022: 1–25.

45. Chiche A, Meshesha M. Towards a Scalable and Adaptive Learning Approach for Network Intrusion Detection. *Journal of Computer Networks and Communications* 2021; vol. 2021: pp. 9.

46. Krishna KVSSR, Bhanu B. Energy Efficient Intrusion Detection Using Deep Reinforcement Learning Approach. *Journal of Green Engineering* 2021; 11: 625–641.

47. Gowri P, Sivapriya G, Kamaleshwar N, Kesavaraj N, others . Real Time Signature Forgery Detection Using Machine Learning. In: IEEE; 2022: 1–5.

48. Maciá-Fernández G, Camacho J, Magán-Carrión R, García-Teodoro P, Therón R. UGR '16: A new dataset for the evaluation of cyclostationarity-based network IDSs. *Computers & Security* 2018; 73: 411–424.

49. Ashoor AS, Gore S. Importance of intrusion detection system (IDS). *International Journal of Scientific and Engineering Research* 2011; 2(1): 1–4.

50. Jeong YS, Hwan HJ, Nam KD. Performance Evaluation Metric for DPI Devices. In: Korea Information Processing Society; 2013: 318–319.

51. Zhang H, Jiang L, Yu L. Attribute and instance weighted naive Bayes. *Pattern Recognition* 2021; 111: 107674.

52. Zhang H, Wu CQ, Gao S, Wang Z, Xu Y, Liu Y. An effective deep learning based scheme for network intrusion detection. In: IEEE; 2018: 682–687.

53. Wang H, Muñoz-González L, Eklund D, Raza S. Non-iid data re-balancing at iot edge with peer-to-peer federated learning for anomaly detection. In: ; 2021: 153–163.

54. Stiawan D, Idris MYB, Bamhdi AM, Budiarto R, others . CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access* 2020; 8: 132911–132921.

55. Suthar F, Patel N, Khanna S. A signature-based botnet (emotet) detection mechanism. *International Journal of Engineering Trends and Technology* 2022; 70(5): 185–193.

56. Maglaras L, Janicke H, Ferrag MA. Combining Security and Reliability of Critical Infrastructures: The Concept of Securability. *Applied Sciences* 2022; 12(20).

57. Komisarek M, Pawlicki M, Simic T, Kavcnik D, Kozik R, Choraś M. Modern NetFlow network dataset with labeled attacks and detection methods. In: ; 2023: 1–8.

58. Gil Bravo A. Special Issue: Feature Papers in Eng 2022. *Eng* 2023; 4(2): 1156–1166.

59. Ramahlosi M, Akanbi Y. A Blockchain-based Model for Securing Data Pipeline in a Heterogeneous Information System. *Published Online by the SAICSIT 2023 Organising Committee Potchefstroom: South African Institute of Computer Scientists In-formation Technologists* 2023: 167.

60. Zhang N, Zhang X, Shang P, et al. Detection of Cotton Verticillium Wilt Disease Severity Based on Hyperspectrum and GWO-SVM. *Remote Sensing* 2023; 15(13).

61. Singh A, Mushtaq Z, Abosaq HA, Mursal SNF, Irfan M, Nowakowski G. Enhancing Ransomware Attack Detection Using Transfer Learning and Deep Learning Ensemble Models on Cloud-Encrypted Data. *Electronics* 2023; 12(18): 3899.

62. Singh A, Mushtaq Z, Abosaq HA, Mursal SNF, Irfan M, Nowakowski G. Enhancing Ransomware Attack Detection Using Transfer Learning and Deep Learning Ensemble Models on Cloud-Encrypted Data. *Electronics* 2023; 12(18).

63. Okafor CM, Kolade A, Onunka T, et al. Mitigating Cybersecurity Risks in the US Healthcare Sector. *International Journal of Research and Scientific Innovation (IJRSI)* 2023; 10(9): 177–193.

64. Tokmak M, Nkongolo M. Stacking an autoencoder for feature selection of zero-day threats. *arXiv e-prints* 2023: arXiv:2311.00304.

65. Shahzad U, Ahmad I, García-Luengo AV, Zaman T, Al-Noor NH, Kumar A. Estimation of coefficient of variation using calibrated estimators in double stratified random sampling. *Mathematics* 2023; 11(1): 252.

66. Nkongolo M, Van Deventer JP, Kasongo SM, Zahra SR, Kipongo J. A Cloud Based Optimization Method for Zero-Day Threats Detection Using Genetic Algorithm and Ensemble Learning. *Electronics* 2022; 11(11). doi: 10.3390/electronics11111749

67. Nkongolo M, Deventer vJP, Kasongo SM, Walt v. dW. Classifying Social Media Using Deep Packet Inspection Data. In: Ranganathan G, Fernando X, Rocha Á., eds. *Inventive Communication and Computational Technologies*Springer; 2023: 543–557.

68. Younas F, Usman M, Yan WQ. A deep ensemble learning method for colorectal polyp classification with optimized network parameters. *Applied Intelligence* 2023; 53(2): 2410–2433.

69. Xu Q, Wang J, Jiang B, Luo B. Fine-grained visual classification via internal ensemble learning transformer. *IEEE Transactions on Multimedia* 2023.

70. Raza A, Munir K, Almutairi MS, Sehar R. Novel Class Probability Features for Optimizing Network Attack Detection with Machine Learning. *IEEE Access* 2023.

71. Nkongolo M, Deventer vJP, Kasongo SM. The Application of Cyclostationary Malware Detection Using Boruta and PCA. In: Smys S, Lafata P, Palanisamy R, Kamel KA., eds. *Computer Networks and Inventive Communication Technologies*Springer; 2023: 547–562.

72. Alsaif SA, others . Machine Learning-Based Ransomware Classification of Bitcoin Transactions. *Applied Computational Intelligence and Soft Computing* 2023; 2023.

73. Alqahtani A, Sheldon FT. A survey of crypto ransomware attack detection methodologies: an evolving outlook. *Sensors* 2022; 22(5): 1837.

74. Li Z, Rios ALG, Xu G, Trajković L. Machine learning techniques for classifying network anomalies and intrusions. In: IEEE; 2019: 1–5.

75. Pérez-Bueno F, García L, Maciá-Fernández G, Molina R. Leveraging a Probabilistic PCA Model to Understand the Multivariate Statistical Network Monitoring Framework for Network Security Anomaly Detection. *IEEE/ACM Transactions on Networking* 2022.

76. Lara A, Ramamurthy B. OpenSec: Policy-based security using software-defined networking. *IEEE transactions on network and service management* 2016; 13(1): 30–42.

77. Nkongolo M, Tokmak M. Zero-day threats detection for critical infrastructures. *arXiv preprint arXiv:2306.06366* 2023.

78. Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences* 2019; 9(20): 4396.

79. Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* 2019; 7: 41525–41550.

80. Louati F, Ktata FB. A deep learning-based multi-agent system for intrusion detection. *SN Applied Sciences* 2020; 2(4): 675.

81. Xu L, Xiong W, Zhou M, Chen L. A Continuous Terminal Sliding-Mode Observer-Based Anomaly Detection Approach for Industrial Communication Networks. *Symmetry* 2022; 14(1): 124.

82. Ahmad T, Truscan D, Vain J, Porres I. Early Detection of Network Attacks Using Deep Learning. In: IEEE; 2022: 30–39.

83. Alzaqebah A, Aljarah I, Al-Kadi O, Damaševičius R. A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System. *Mathematics* 2022; 10(6): 999.

84. Liu G, Zhao H, Fan F, Liu G, Xu Q, Nazir S. An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. *Sensors* 2022; 22(4): 1407.