

Optimizing intrusion detection in 5G networks using dimensionality reduction techniques

Zaher Salah¹, Esraa Elsou², Waleed Al-Sit^{3,4}, Esraa Alhenawi², Fuad Alshraideh⁵, Nawaf Alshdaifat⁶

¹Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University, Zarqa, Jordan

²Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

³Department of Computer Engineering, Faculty of Engineering, Mu'tah University, Al-Karak, Jordan

⁴Department of Information Technology, Faculty of Computer Information Science, Higher Colleges of Technology, Dubai, United Arab Emirates

⁵Department of Software Engineering Faculty of Information Technology, Zarqa University, Zarqa, Jordan

⁶Faculty of Information Technology, Applied Science Private University, Amman, Jordan

Article Info

Article history:

Received May 9, 2024

Revised Jul 1, 2024

Accepted Jul 7, 2024

Keywords:

5G networks

Feature selection

Intrusion detection

Machine learning

Network security

Wi-Fi networks

ABSTRACT

The proliferation of internet of things (IoT) technologies has expanded the user base of the internet, but it has also exposed users to increased cyber threats. Intrusion detection systems (IDSs) play a vital role in safeguarding against cybercrimes by enabling early threat response. This research uniquely centers on the critical dimensionality aspects of wireless datasets. This study focuses on the intricate interplay between feature dimensionality and intrusion detection systems. We rely on the renowned IEEE 802.11 security-oriented AWID3 dataset to implement our experiments since AWID was the first dataset created from wireless network traffic and has been developed into AWID3 by capturing and studying traces of a wide variety of attacks sent into the IEEE 802.1X extensible authentication protocol (EAP) environment. This research unfolds in three distinct phases, each strategically designed to enhance the efficacy of our framework, using multinomial class, multi-numeric class, and binary class. The best accuracy achieved was 99% in the three phases, while the lowest accuracy was 89.1%, 60%, and 86.7% for the three phases consecutively. These results offer a comprehensive understanding of the intricate relationship between wireless dataset dimensionality and intrusion detection effectiveness.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Zaher Salah

Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University

Zarqa, Jordan

Email: zaher@hu.edu.jo

1. INTRODUCTION

The future of wireless networks requires a unified architecture to support a wide range of devices, users, and services with different latency requirements and data speeds. Current wireless technologies, such as third generation (3G), and fourth generation-long term evolution (4G-LTE), have a lot of limitations that prevent future system improvements to meet these needs. To meet these expectations, researchers have worked to develop fifth generation (5G), a cutting-edge wireless communication technology. After conducting multiple scientific studies, it was determined that the fifth-generation technology also has drawbacks that make it inappropriate for low-power wide-area, or long-distance communication [1]. This means that the available deployed communication technologies will not be able to keep up with future

requirements completely and efficiently. Furthermore, by 2030, it is predicted that a more advanced digital society supported by limitless wireless connectivity will have emerged [2]. The rise of 5G has made the ground-breaking concept of the internet of things (IoT) viable, and it is revolutionizing how platforms, software, people, and devices are connected. 5G provides seamless connectivity and makes it simpler for many firms to be linked to the vast internet network by utilizing cutting-edge technologies and innovative concepts. These devices are expected to be pre-installed with IoT modules that allow for direct device-to-device communication, which is necessary for building large-scale IoT networks [3]. Additionally, 5G will make it easier for radio access technology (RAT) to link those devices. The introduction of new radio technologies including massive multiple-input and multiple-output (MIMO), mmWave, non-orthogonal multiple access (NOMA), and other IoT communication technologies will be made possible by the rollout of 5G networks.

Developing a comprehensive security protocol is the first step toward coordinating the actions required to protect wireless networks. Technical controls that are intended to improve wireless security comprise both software and hardware elements. Hardware countermeasures include things like virtual private networks (VPNs), smart cards, and biometrics. On the other hand, software countermeasures consist of the right setups for access points, software updates, intrusion detection systems (IDS), authentication protocols, and encryption methods [4]. The IDS has been essential in preventing and spotting attacks in the early stages. Authentication, firewalls, data encryption, and other traditional methods of network protection are the most frequently used in securing networks, but they are unable to balance resource usages such as energy, bandwidth, and intrusion detection effectiveness. Intelligent IDS is an emerging solution for network protection and security against attacks. Machine learning has made it possible to handle network intrusions in a variety of ways. Consequently, there has been a lot of interest from worldwide programs and research regarding the use of machine learning (ML) technologies in 5G networks.

Cybersecurity solutions need to have network access. They continuously improve their accuracy and performance by learning from behavior patterns using labelled datasets overtime [5]. This situation can identify zero-day attacks, through identifying malicious traffic patterns effectively leading to early attack detection. Furthermore, machine learning techniques enable the use of adaptive learning capabilities that can lead to significant new assault identification and intelligent handling of emerging threats. For instance, this ability plays a pivotal role in enhancing strong security protocols for wireless networks and internet of things (IoT) devices that will enhance improved protection and resilience against evolving cyber threats [6].

The development of an effective wireless intrusion detection system presents several major challenges, including: high-dimensional data handling: In some situations when there are large numbers of features in the dataset, wireless IDS often comes across difficulties in processing high-dimensional data. Such problems are actually solved with dimensionality reduction techniques that help get rid of them. By reducing training data's input variables count, these techniques may present information on a lower-dimensional subspace capturing those details that are most important in order to continuously monitor network traffic and spot variations from the norm that could indicate security breaches, techniques including anomaly detection, pattern recognition, and machine learning algorithms are used. To address these inquiries, we conducted a comprehensive experiment encompassing three distinct phases: multi-nominal class analysis, multi-numeric class investigation, and binary class assessment. Additionally, we enhance the model's performance through the strategic utilization of feature selection techniques. These contributions collectively underscore the robustness and applicability of our framework in tackling intrusion detection challenges within 5G and IoT environments [7].

The remaining part of the research is divided into the following sections: literature review in section 2 The existing literature on IDS that use the AWID3 dataset is thoroughly reviewed in this section. It reviews earlier the research, methodology, and strategies applied in the field, highlighting significant discoveries, constraints, and knowledge gaps. The procedure used to reprocess the dataset is thoroughly documented in section 3. This covers both the implementation and initial setup procedures of the intrusion detection system as well as the actions taken to preprocess the AWID3 dataset. In section 4, the AWID3 dataset is thoroughly evaluated throughout all of its stages. Utilizing the proper metrics and procedures, the effectiveness and performance of the intrusion detection system are evaluated. Section 5 summarizes the main conclusions and contributions of the study before concluding. The importance of the research findings and their consequences for the discipline of intrusion detection in wireless networks are highlighted in this section. It also talks about future directions for the field's progress and study.

The number of IoT devices linked to Wi-Fi networks has significantly increased as a result of communication network development. These electronic devices produce a lot of data traffic, some of which could be harmful since hackers can utilize the flood of data they can access to break into customers' networks. This makes it very difficult to recognize and stop these kinds of attacks. To mitigate this difficulty, feature selection is essential since it reduces the volume of data that intrusion detection model classifiers must process. The effectiveness of these problems can be effectively addressed by improving the

performance of IDS by removing unnecessary information and choosing the most useful elements from the data. The focus of this study will be on the AWID dataset, a benchmark dataset for Wi-Fi network intrusions.

Since the previous datasets mostly addressed IDSs in a broader sense, this dataset was created in response to the lack of thorough datasets dedicated to wireless intrusion detection systems (WIDSs). Chatzoglou *et al.* [8] particularly focused on attacks leveraging 802.11 and non-802.11 network protocol characteristics that target the application layer. Such as botnet, malware, SSH, SQL injection, website spoofing and simple service discovery protocol (SSDP) amplification using the AWID3 benchmark dataset. They removed several aspects from the initial feature set that they felt were ineffective in identifying application layer threats to improve the effectiveness of their research. They, therefore, conducted their trials using 16 and 17 features. The classes of the dataset were divided into normal, flood, and other categories. The six application layer assaults in the dataset were mapped to the flood and other classes. Bot-nets, malware, and SQL injection were all included in the other class, however, SSDP amplification, website spoofing, and SSH were expressly included in the Flooding class. Similar to our strategy, the authors used a k-fold validation procedure with k set to 10. In the experiments they conducted, three machine learning (ML) models: decision trees (DT), lightGBM, and bagging, were used. Using the aforementioned ML models, the Authors were able to reach an accuracy rate of 98.71%. They also investigated deep learning methods, achieving a maximum accuracy of 97.86%. Additionally, they also used feature set conflation, a strategy we used for binary classification that increased accuracy to 99%.

Chatzoglou *et al.* [9] intended to determine the bare minimum set of classifier features that could be used with any 802.11 implementation version. Their research also looked at how well machine learning algorithms performed in detecting different network assaults using datasets from the AWID family. The authors chose 16 features for their studies that were largely applicable to all frame types and sub-types of 802.11, guaranteeing their direct applicability to a variety of network configurations. They specifically avoided characteristics that displayed recurrent patterns since they could generate biases and result in overfitting. This set of features was chosen since it was predicted that they would be constant throughout all frames, preventing analysis bias. Saini *et al.* [10] concerned on studying the WPA3 protocol, where they presented a two-stage technique in their research to handle the problem of intrusion detection within a network. They achieved 99% accuracy in detecting network threats. The results of their research demonstrated the importance of ML in boosting network security within the WPA3 protocol, and advancing intrusion detection frameworks.

A study on the process of transferring machine learning algorithms to programmable network devices was carried out by Zheng *et al.* [11]. Furthermore, an evaluation and comparison of recently proposed and state-of-the-art in-network machine learning algorithms were conducted concerning their functionality, throughput, scalability, and resource utilization. Six different datasets, including KDD99 and AWID3, were used by the researchers for intrusion detection. These algorithms' accuracy ranged from 49.37% for k-nearest neighbor (kNN) to 97.47% for decision trees [11]. Moving toward some popular research that detects network intrusion using various machine learning algorithms, Sethuraman [12] proposed a wireless intrusion detection system focused on the passive mode for the access point since the wireless attacks are somehow tricky to spoof users by introducing a fake access point claiming to be a legitimate one. The proposed method achieved accuracy results of 98% using the AWID dataset. Anthi *et al.* [13] proposed a three-layer IDS that employs a supervised methodology to identify a variety of well-known network-based cyberattacks on internet of things networks, including denial of service (DoS), man-in-the-middle, spoofing, reconnaissance, and replay attacks. They have applied the neural network algorithm to the NSL-KDD dataset and the achieved accuracy was 96%. Due to the non-linear nature of the intrusion attempt and a large number of features, network traffic performance is unpredictable. These present a challenge for intrusion detection systems researchers. As a result, Alzubi *et al.* [14] proposed the modified binary grey wolf optimization feature selection algorithm (MBGWO). To enhance IDS performance, the suggested algorithm is based on binary Greywolf optimization. After applying the machine learning algorithm [15] to the NSL- KDD dataset, they obtained a 99.22% accuracy result. A feed-forward deep neural network and feature extraction are used in Kasongo and Sun [16] proposed wireless intrusion detection system. The datasets that were chosen were the AWID and UNSW-NB15. A study was carried out to compare their outcomes with those that were obtained through popular machine learning algorithms, including k-nearest neighbor (kNN), random forest (RF), support vector machine (SVM), naive Bayes (NB), and decision tree (DT). Four categories were created out of the experimental research: full features, chosen features, and binary and multiclass attacks. The AWID dataset's feature set was whittled down to 26 using the extra trees (ET) approach. With 359,115 cases in the training dataset and 115,128 instances in the test dataset, together they accounted for 20% of the total AWID-CLS dataset.

About binary classification, the wireless IDS system performed about 98.6% accuracy on validation data and 98.69% accuracy on test data. The maximum accuracy for multi-class classification was 98.59% on

the test data and 98.47% on the training data. These outcomes demonstrate how well the system classifies situations with accuracy. Unexpectedly the study found that accuracy increased as the number of features decreased. On the test and validation sets of data, the model showed exceptional accuracy rates of 99.66% and 99.67% for binary classification, respectively. These results show how the accuracy of the model is improved by feature reduction. In this context, it is worth noting that feature selection reduces errors in predications from 99.78% to 99.77% and therefore very small changes in the input data can lead to big changes in the output results of a model. The AWID2 dataset is important for wireless IDS. It runs on top of WEP-based architecture and has a large number of packets. This dataset has become an essential reference point for wireless IDS literature with over 150 different features. The AWID dataset was improved and further developed into AWID3 to achieve better performance. To make this change possible, remnants from cyber-attacks against IEEE 802.1X extensible authentication protocol (EAP) were collected and analyzed thoroughly. By focusing on this particular environment, AWID3 provides a more specialized as well as contextually relevant dataset for re-searching as well as developing wireless IDS techniques.

2. METHOD

In this section, there is an introduction to dataset properties, attacks and structure as well as a description of our machine learning-based implementation of the framework for intrusion detection in wireless networks. However, it should be noted that an elaborate explanation of our contributions is found in Section 4 where we delve into details about our research approach and results. The main objective of this paper is to address the intricate complexities associated with intrusion detection within 5G and IoT environments. To accomplish this, we carried out a comprehensive experiment involving three stages; multi-nominal class analysis, multi-numeric class investigation, and binary class assessment. Through these phases we aimed at assessing our method's effectiveness under different scenarios while having an insight on its adaptability. Moreover, besides these three phases we have singled out one special component of our study: a new approach towards overlapping feature selection procedures. Our contribution has been an innovative way of implementing the technique used for selecting relevant features which are important for building intrusion detection system. In contrast to traditional feature selection methods that operate on their own, our methodology explores aggregate effects of various feature selection techniques used at once together. Combining the results of Relief, analysis of variance (ANOVA), chi-squared feature selection, and gain ratio and information gain attribute evaluations, we present a fresh perspective.

2.1. Preprocessing

As several research publications [15], [17] have shown, the initial preprocessing step is an important and helpful procedure that is essential for acquiring exact data needed for developing a classifier. An essential step in the knowledge extraction process is data pretreatment. Its purpose is to convert raw data into a more efficient and effective format for further processing. This phase is vital since making accurate judgements depends on having high-quality data. Therefore, the preprocessing techniques were applied to the AWID3 dataset. The AWID3 dataset consists of 13 CSV files, with a total of 36,913,503 occurrences. This includes 30,387,099 instances of conventional network traffic and 6,526,404 cases of malicious activity. Using this dataset, the following method was carried out. The second step is choosing a sample that includes multi-attacks with 120,000 instances and 254 features and the third step is removing empty features and features with a constant value. The fourth step is randomly shuffling the order of instances passed through it. The random number generator is reset with the seed value whenever a new set of instances is passed in, we chose the seed value to be 33. The fifth step is replacing all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data and the final step is conversion of data type: convert string attributes to nominal. Figure 1 shows the preprocessing steps followed in AWID3 dataset, while Figure 2 shows the proposed framework for processing and classification method that was followed in our experiment.

2.2. Structure of AWID3 dataset

The proliferation of wireless technology has experienced significant growth in recent years. Despite significant attempts to safeguard these technologies, the majority of security measures have been found to be insufficient in practice. The objective of the AWID project is to establish a strong foundation for researchers to create reliable security mechanisms for present and future wireless networks. This will be achieved by offering tools, methodologies, and datasets that are specifically tailored for wireless networks, as the existing datasets were not designed for this purpose.

Due to the widespread use of smart devices like smartphones, smart watches, tablets, and internet of Things devices, Wi-Fi (IEEE 802.11) has been supplanted as the standardized method for connecting digital devices in wireless local area network (LAN). Wi-Fi is commonly utilized in both essential areas and

residential, commercial, and institutional settings. Not surprisingly, there has been a significant amount of academic research dedicated to studying the security of the 802.11 protocol and Wi-Fi networks. Despite many updates and corrective efforts, vulnerabilities have been discovered in even the latest iterations of the software, persisting for almost two decades. The issue of security in wireless technology has remained unresolved for a significant period of time. External security measures are essential components of 802.11 wireless networks to prevent both known and unexpected assaults [18].

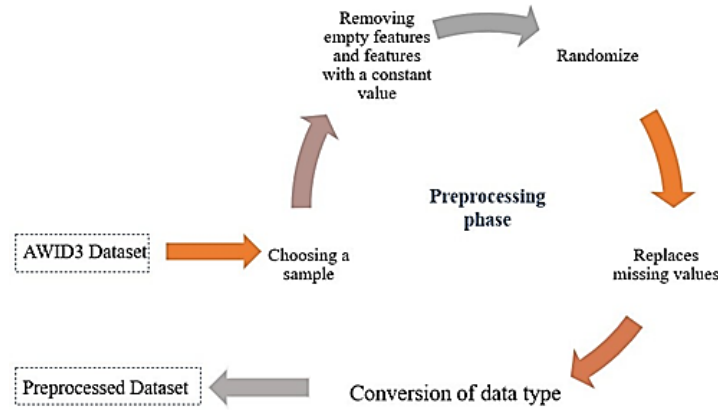


Figure 1. Preprocessing steps followed in AWID3

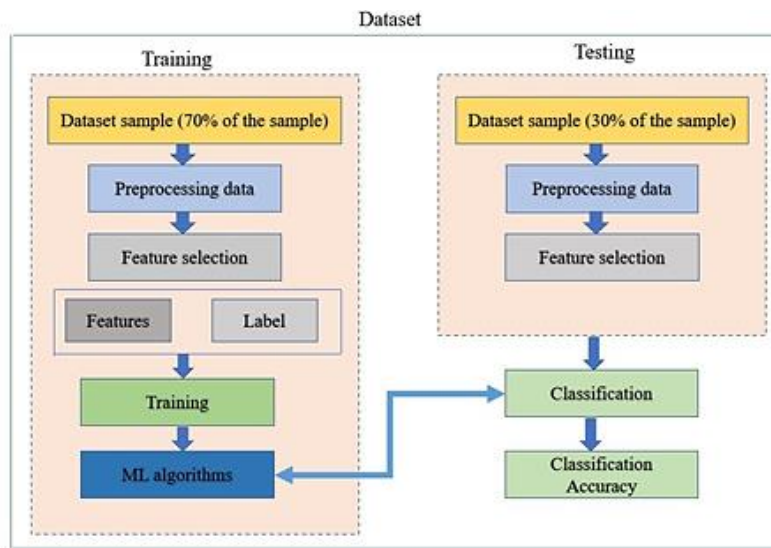


Figure 2. Proposed framework for processing and classification model

The AWID dataset was extracted in 2016 and subsequently evolved into a new version called AWID3 in 2021. The IEEE 802.1X extensible authentication protocol (EAP) environment is susceptible to many assaults. The AWID3 dataset, which is freely accessible, is specifically designed to gather and examine the traces of these attacks. This analysis is the initial examination of the IEEE 802.11w standard, which is mandatory for hardware that has obtained WPA3 certification. AWID3 is expected to play a critical role in the design and evaluation of intrusion detection systems. The AWID dataset is composed of 254 features in CSV format, consisting of 253 generic features and one extra feature specifically for labelling purposes. Both the MAC and application layers encompass the extracted features. The AWID3 dataset was obtained and gathered utilizing a total of 16 distinct real and virtual machines. The dataset comprises 36,913,503 occurrences, including 6,526,404 instances of malicious traffic and 30,387,099 cases of normal traffic. There are a total of 13 distinct attack types seen in malicious network traffic. The dataset is comprehensively detailed in the subsequent subsections.

2.3. AWID3 features description

The AWID3 dataset is used to display an in-depth analysis of machine learning classifier experiments. We implemented our experiments on the chosen sample when the label class is nominal, numerical, and binary (Normal=0, Attack=1). Additionally, since feature selection is one of the best ways to improve training model performance [9], we decided to use feature selection techniques in our experiments. Finding the minimum number of features required to guarantee that the probability distribution of the resulting data classes closely resembles the original distribution when all features are used is the goal of feature selection. Using fewer attributes in patterns makes patterns easier to understand, which is one of the additional benefits of classification without using all features. It also reduces learning runtime and improves classification accuracy. The feature selection techniques that we have used are the gain ratio attribute evaluation and information gain attribute evaluation, Relief, ANOVA, and Chi-squared feature selection. The attributes which are considered for the evaluation from previous feature selection techniques are listed in Table 1.

Table 1. The AWID3 dataset features description

Network and Protocol Information	Network Protocols
<i>tcp.checksum</i> : TCP checksum value.	<i>wlan.sa</i> : Source MAC address in WLAN.
<i>tcp.payload</i> : Payload of TCP packets.	<i>llc</i> : Logical Link Control protocol.
<i>wlan.duration</i> : Duration of WLAN packets.	<i>ip.version</i> : IP protocol version.
<i>frame.time-delta-displayed</i> : Time difference between displayed frames.	<i>ip.proto</i> : IP protocol type.
<i>frame.time-delta</i> : Time difference between frames.	<i>tcp.checksum.status</i> : Status of TCP checksum.
<i>frame.time</i> : Time of frame capture.	<i>ip.ttl</i> : IP Time-to-Live.
<i>tcp.time-relative</i> : Relative time for TCP packets.	<i>ip.src</i> : Source IP address.
<i>radiotap.channel.freq</i> : Frequency of the radio channel.	<i>tcp.flags.reset</i> : TCP reset flags.
<i>wlan.fc.moredata</i> : WLAN frame control field indicating more data.	<i>tcp.flags.syn</i> : TCP synchronization flags.
<i>wlan-radio.frequency</i> : Frequency of the WLAN radio.	<i>tcp.flags.fin</i> : TCP finish flags.
<i>wlan-radio.channel</i> : Channel used by WLAN radio.	<i>tcp.flags.ack</i> : TCP acknowledgment flags.
<i>wlan.fc.ds</i> : WLAN frame control field indicating data-to-station.	<i>tcp.flags.push</i> : TCP push flags.
<i>wlan.fc.type</i> : WLAN frame control field indicating frame type.	<i>frame.number</i> : Frame number.
<i>wlan.fc.protected</i> : WLAN frame control field indicating frame protection.	<i>frame.len</i> : Frame length.
<i>radiotap.channel.flags.cck</i> : Radio channel flags for CCK modulation.	<i>frame.time-relative</i> : Relative frame time.
<i>wlan.fc.subtype</i> : WLAN frame control field indicating frame subtype.	<i>tcp.ack</i> : TCP acknowledgment.
<i>wlan.fc.pwrmtg</i> : WLAN frame control field indicating power management.	<i>tcp.analysis</i> : TCP analysis data.
<i>wlan-radio.phy</i> : PHY type used by the WLAN radio.	<i>tcp.seq</i> : TCP sequence number.
<i>radiotap.channel.flags.ofdm</i> : Radio channel flags for OFDM modulation.	<i>tcp.seq-raw</i> : Raw TCP sequence number.
<i>radiotap.present.tsft</i> : Presence of Timestamp field.	<i>tcp.time-delta</i> : Time difference for TCP packets.
<i>wlan.ra</i> : Receiver's MAC address.	Signal strength and quality
<i>radiotap.length</i> : Length of the radiotap header.	<i>radiotap.dbm-antenna</i> : Signal strength in dBm.
<i>wlan.fc.retry</i> : WLAN frame control field indicating frame retry.	<i>wlan-radio.signal-dbm</i> : Signal strength for WLAN radio in dBm.
<i>wlan.ta</i> : Transmitter's MAC address.	
<i>wlan.bssid</i> : BSSID of the WLAN.	

2.4. Attacks in AWID3

Unlike the original AWID dataset, the current frame includes a number of attacks that take advantage of loopholes in higher-layer protocols, in addition to some newer attacks. The various attacks were divided into three groups by AWID [19], while AWID3 divides the attacks into four categories [18]:

- 802.11 Specific attacks: These types of attacks only target the MAC layer of 802.11 systems and mostly target wireless networks by continuously presenting serious threats that are not being stopped. It can be divided into two categories: key re-installation and denial of service (DoS). Denial of service tries to interfere with the connection between the key units of an 802.11 network, including station (STA) and access point (AP), by attacking devices or putting more emphasis on the resources of the network and the connected devices on this network. The AWID3 dataset contains almost all these well-known attacks. A key re-installation attack aims to reinstall a pair-wise key or group key that was previously used in the framework. There are six types of 802.11-specific attack categories: Deauthentication, Disassociation, Re-association, Rogue AP, Krack, and Kr00k.
- Attacks against the local nodes: Well-known attacks only require a few steps. They are launched at benign nodes in the local network by a malicious wireless network or a hacked node. They primarily affect higher layers, like the application layer. There are three types of attacks in this category: SSH brute force, Botnet, and Malware.
- Attacks against external nodes: Attacks of this type typically involve a small number of actions that are started by corrupt or malicious local clients. The attack target in this case is located outside the internet. There are two types of attacks against external nodes: SSDP amplification and SQL injection attacks.

- Multi-layer attacks: Because the clients cannot place total trust in the architecture that links them to the internet, this class also includes multi-step attacks that utilize at least two different layers. Two types of attacks are considered in this category: evil twin and Website spoofing.

Table 2 presents the number of each type of these attacks in the dataset in detail. Now we will describe the attacks included in the AWID3 dataset successively.

- a. Deauthentication attack: To establish a connection between the client and the AP the client must associate with the AP and must finish the authentication process before exchanging data. If the client wants to disconnect, he must submit a disassociation frame to the AP. Alternatively in case a client suddenly and unexpectedly leaves an AP, he has to send a deauthentication frame. The deauthentication or disassociation frames are unencrypted and do not need authentication, according to the 802.11 network specifications. As a result, an attacker can quickly impersonate a client or access point's MAC address to send deauthentication requests on their behalf. Identifying legitimate deauthentication from fraudulent deauthentication could be investigated by verifying the source of deauthentication requests [20].
- b. Disassociation attack: After authentication, a client and the AP communicate via an association message to connect the client to the AP. Upon receiving a message, an attacker sends an access point a spoofed message; the AP then disconnects the client whose MAC address is mentioned in the message [21]. Thus, stopping the communication between the Mesh AP and the client, but the client was still authenticated to the previously associated network. The client can re-associate after the attack by only sending the re-association request. As the re-connection requires less time, in this case, this attack is, therefore, less dangerous than the deauthentication attack [22].
- c. Re-association attacks: a wireless client switches to a different access point during a reassociation attack. To transfer client data, the old and new access points establish a connection via the wired network during roaming. This method is like the common association process. When a client roams to a different access point, it signals its new position by initiating reassociation. The client notifies the new access point of the old one using a re-association packet. The new access point uses a wired channel to communicate with the old one to confirm the client's previous connection. If the client was previously connected, the new access point responds with a re-association frame; otherwise, it sends a disassociation frame. After the re-association response, the new access point contacts the old access point via the wired channel to complete the procedure. The new access point processes client frames after receiving buffered frames from the old one. Reassociation attacks exploit vulnerabilities in this system, potentially enabling unauthorized access or interference with wireless communication [23].
- d. Rogue AP attack: Wi-Fi hot-spot service is available in many public places. The public nature of these places and the poor security make these hot spots vulnerable to attacks, such as spoofing, fraud, and rogue AP. Rogue AP creates a fake AP using the same name (SSID), so it appears like a legitimate one [24].
- e. The Krack attack has been identified as a potential security threat to the existing encryption methods employed to safeguard and secure Wi-Fi networks over the last 15 years. There is no assurance that every device will receive a patch and be safeguarded against these assaults originating from any networked location [25]. Kr00k Attack refers to a vulnerability that allows for the decryption of certain Wi-Fi communication that has been encrypted using the WPA2 protocol. The vulnerability was found by the security company ESET in 2019. As per ESET, this vulnerability impacts around one billion devices. Devices equipped with Wi-Fi chips that have not yet been updated by Broadcom or Cypress are susceptible to the Kr00k vulnerability. These Wi-Fi chips are utilized by the majority of contemporary Wi-Fi-enabled devices, such as smartphones, tablets, laptops, and internet of things (IoT) devices [26].
- f. SSH brute force attack: a popular internet communication protocol used by programmers, webmasters, and system administrators is called secure shell (SSH). Attackers typically use scripts and applications as brute-force tools. To get around authentication procedures, these tools try numerous password combinations. The host becomes the target of continuing brute force attacks if it is directly connected to the internet or WAN and the SSH service is active [27]. Machine learning techniques have been used to detect automated network-level brute force attacks using flow data [28].
- g. Botnet attack: one of the most common security threats is an IoT-based botnet, which spreads more quickly and has a bigger impact than other attacks. Popular topics in cybersecurity literature include botnet detection [29].
- h. Malware: malware is defined as malicious software that is frequently used by criminals to launch cyberattacks against target computers. Any software that maliciously executes payloads on victims' computers (computers, smartphones, computer networks, and so on) is referred to as malware. Viruses, worms, Trojan horses, rootkits, and ransomware are just a few examples of the various types of malwares [30].
- i. SSDP amplification: SSDP is a component of the Universal Plug and Play Protocol standard. Using this protocol, devices connected to the internet can seamlessly discover services. Utilizing these protocols,

attackers launch DDoS attacks by boosting and reflecting network traffic at their targets [30]. For standard SSDP service, a request sender will get responses from service providers. However, a hacker must first build a botnet by gathering vulnerable hosts and devices from the internet to carry out an SSDP reflection attack. This gives the attacker control of the request's sender and enables him to spoof the victim's IP address in IP address request packets.

- j. SQL injection attack: SQL injection attack: SQL injection refers to a class of code injection attacks in which user-supplied data is included in an SQL query in such a way that some of the user's input is treated as SQL code. Attacks using SQL injection are extremely dangerous because once an attacker has gained access to the system database, they can change the data already there. Attackers may harm the owner of the injected website through improper data manipulation [31], and they may use this information to harm their targets [32].
- k. Evil twin: an attacker can impersonate a legitimate access point because spoofing the network name and MAC address of a legitimate access point is understandable. This fake AP claims to be a legitimate access point known as the evil twin. The hotspot and software capabilities on the client devices are enough to launch the evil twin attack. If a client connects to an evil twin, it can enter as a man-in-the-middle attack between legitimate access points and the client, so he can eavesdrop on or manipulate sensitive client data. In addition, many malicious twin attacks can cause the WLAN to break due to the severe effect on internet services [33].
- l. Website spoofing: The practice of "website spoofing" involves online criminals creating a website that closely matches a reputable brand and a domain, that is almost an exact copy of the web address of the legitimate brand. To obtain private information such as login credentials, Social Security numbers, credit card information, or bank account numbers, website spoofing tricks customers, partners, and employees of a brand into a fake website [34], [35].

Classifiers are algorithms that automatically assign and categorize data points into categories or classes. There are two main types of classifier models: supervised and unsupervised. In supervised learning, classifiers are trained using labeled data to distinguish between different categories. In contrast, unsupervised algorithms are concerned with pattern recognition to classify the unlabeled dataset. The classifier models that were used in the classification process are highlighted in this section: (a) decision tree, (b) decision forest, (c) decision jungle, (d) logistic regression, and (e) naïve Bayes.

Table 2. Attack types on AWID3 dataset

Attack	Normal traffic	Malicious traffic
Deauth	1,587,527	38,942
Disas	1,938,585	75,131
(Re)Assoc	1,838,430	5,502
Rogue AP	1,971,875	1,310
Krack	1,388,498	49,990
Kr00k	2,708,637	186,173
SSH	2,428,688	11,882
Botnet	3,169,167	56,891
Malware	2,181,148	131,611
SQL injection	2,595,727	2,629
SSDP	2,641,517	5,456,395
Evil twin	3,673,854	104,827
Website spoofing	2,263,446	405,121
Total	30,387,099	6,526,404

3. RESULTS AND DISCUSSION

This section provides a detailed description of the AWID3 dataset's implementation and evaluation, along with the associated findings. WEKA, AZURE, and MATLAB were employed, as was highlighted in the section before. Several machine learning algorithms were implemented. The following sections detailed the different phases of the evaluation process where the dataset is used. The dataset consists of (36,913,503) instances (30,387,099) of legitimate traffic and (6,526,404) instances of malicious traffic. In this research we will choose a sample from the dataset consisting of several attacks in addition to normal traffic, then we will apply ML algorithms after preprocessing the data as mentioned in the previous section in Figure 1. To get the best accuracy results and to get the most effective model for detection attacks in such an environment, we will implement our experiments on the chosen sample in three phases; when the label class is nominal, numerical, and binary (Normal=0, Attack=1). An evaluation of ML approaches using both nominal and numerical features is necessary for ensuring robustness, versatility, and applicability in real-life scenarios. As a result, a more comprehensive assessment of model performance and a deeper understanding of the problem can be achieved.

3.1. Phase I: multi attack (nominal class)

In this experiment, we will apply ML algorithms on a sample that includes multi-attacks with 120,000 instances and 254 features, to evaluate the IDS model in multi-class, the attacks included in the sample are presented in Table 3. The procedure that was followed to implement this phase of evaluation and experiment was as follows:

- a. Preprocessing step in several stages:
 - At first, the data was cleaned by removing empty features and features with a constant value, the remaining features were 49.
 - Randomize the data: Randomly shuffle the order of instances passed through it. The random number generator is reset with the seed value whenever a new set of instances is passed in, we chose the seed value to be 33.
 - Remove percentage: To fit the allocated memory for WEKA. The new sample consists of 7,499 instances.
 - Remove attributes that have over 50% of missing value, the remaining attributes are 44.
- b. Feature selection based on gain ratio evaluation and information gain evaluation.

The results of two different feature selection algorithms revealed that not all features have a considerable impact on identifying the label. Figures 3 and 4 show the features ranked by their importance; from the most important to the least important, in order of significance to the response variable. According to the gain ratio attribute evaluation and info gain attribute evaluation. The results show that not all of them are necessary to be used to build a successful ML model, on the other hand, some features copy the label as it is, so we have to remove them.

Table 3. Training and testing in AWID3 dataset

Attack type	Traffic in the sample
Krack	20,000
Kr00k	20,000
Disas	20,000
Malware	20,000
SSDP	20,000
Normal	20,000
Total	120,000

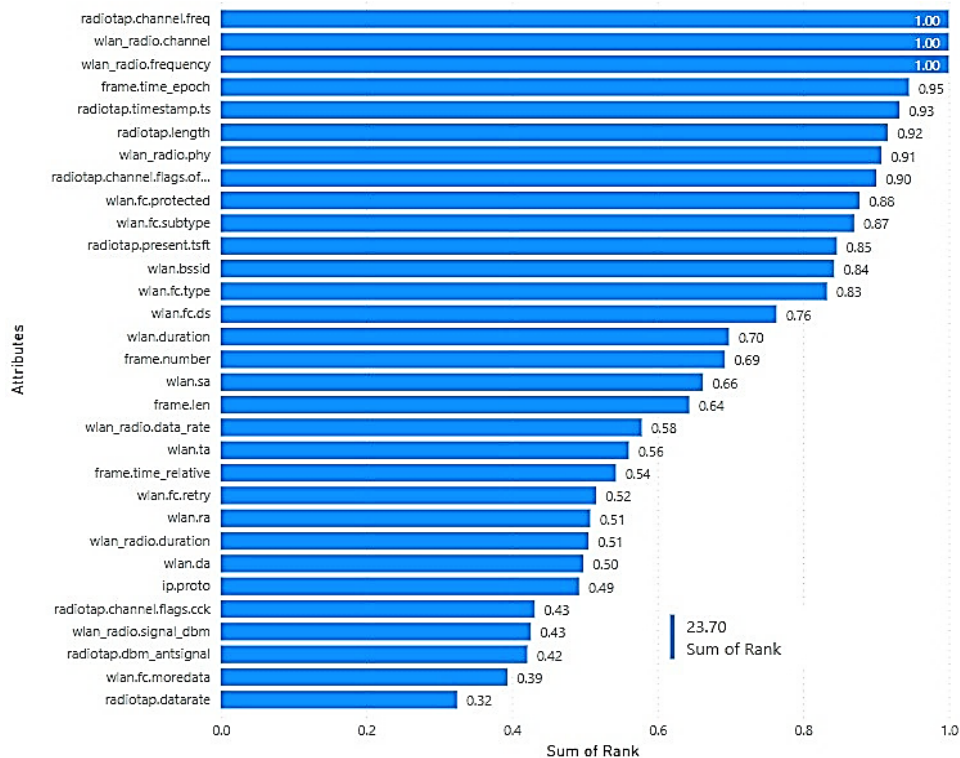


Figure 3. Gain ratio attribute evaluation

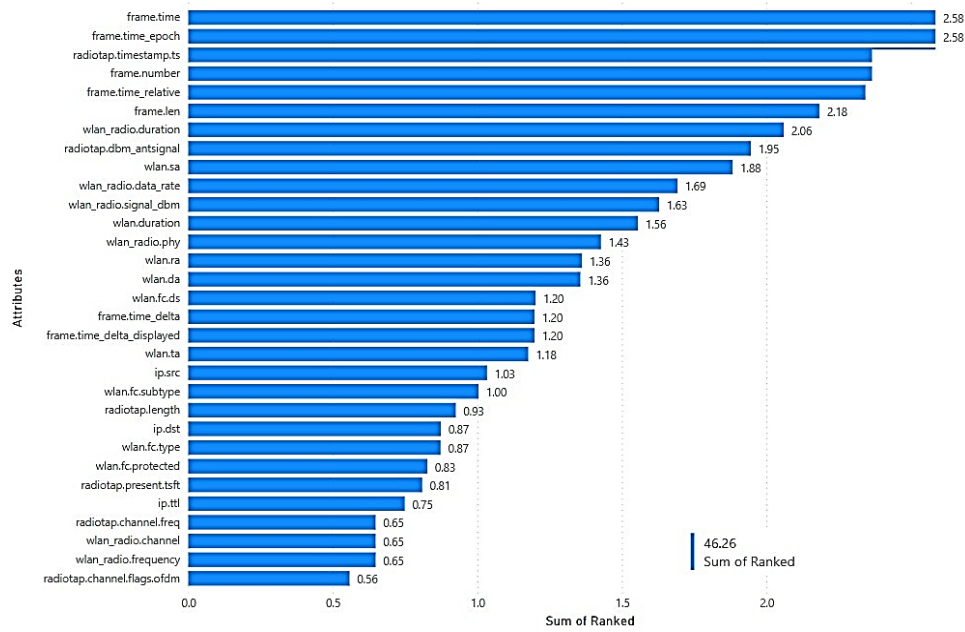


Figure 4. Information gain attribute evaluation

According to the gain ratio feature evaluator, these features (*wlan_radio.frequency*, *radiotap.channel.freq*, *wlan_radio.channel*, *radiotap.timestamp.ts*, *radiotap.length*, *wlan_radio.phy*, *radiotap.channel.flags.ofdm*) copy the label, so we removed them to get the accurate result for the ML model. The remaining 36 features are (*wlan.fc.protected*, *wlan.fc.subtype*, *radiotap.present.tsft*, *wlan.bssid*, *wlan.fc.type*, *wlan.fc.ds*, *wlan.duration*, *frame.number*, *wlan.sa*, *frame.len*, *wlan_radio.data_rate*, *wlan.ta*, *frame.time relative*, *wlan.fc.retry*, *wlan.ra*, *wlan_radio.duration*, *wlan.da*, *ip.proto*, *radiotap.channel.flags.cck*, *wlan_radio.signal_dbm*, *radiotap.dbm_antisignal*, *wlan.fc.moredata*, *radiotap.datarate*, *ip.ttl*, *ip.src*, *frame.time_delta*, *frame.time_delta_displayed*, *ip.dst*, *wlan.fc.pwrmtg*, *wlan.fc.frag*, *frame.time*, *wlan.seq*, *ip.version*, *llc*, *wlan.fc.order*). Following data preprocessing and the application of feature selection methods, we utilized various machine learning algorithms. Table 4 displays the learning algorithms performance utilizing WEKA, and Table 5 displays the learning algorithms performance utilizing AZURE.

Table 4. Performance of the learning algorithms on gain ratio and info gain using WEKA

Algorithm	Gain Ratio-Nominal				10-fold cross-validation			
	Accuracy	Precision	Recall	F-Measure	Accuracy	Precision	Recall	F-Measure
treesJ48	99.82%	0.997	0.997	0.977	99.84%	0.998	0.998	0.998
Naïve Bayes	98.76%	0.997	1	0.999	99.21%	0.998	0.996	0.997
Logistic	99.82%	1	1	1	99.73%	0.998	0.989	0.993

Algorithm	Info Gain-Nominal				10-fold cross-validation			
	Accuracy	Precision	Recall	F-Measure	Accuracy	Precision	Recall	F-Measure
treesJ48	99.67%	1	1	1	99.69%	1	1	1
Naïve Bayes	92.38%	0.995	0.998	0.996	92.39%	0.995	0.998	0.996
Random Tree	99.44%	1	0.997	0.998	99.49%	0.99	1	0.99

Table 5. The learning algorithms performance on gain ratio and info gain using AZURE

Algorithm	Gain Ratio Nominal			
	Overall Accuracy	Average Accuracy	Precision	Recall
Multiclass Decision Forest	0.91372	0.97124	0.9587	0.9709
Multiclass Decision Jungle	0.89103	0.96368	0.9155	0.8911
Multiclass Logistic Regression	0.99989	0.99996	0.9999	0.9999

Algorithm	Info Gain-Nominal			
	Overall Accuracy	Average Accuracy	Precision	Recall
Multiclass Decision Forest	0.99133	0.99711	0.9916	0.9913
Multiclass Decision Jungle	0.92969	0.97657	0.9393	0.9296
Multiclass Logistic Regression	0.94375	0.98125	0.9569	0.9436

3.2. Phase II: multi attack (numeric class)

In this Phase we applied ML algorithms in the same previous sample that includes multi attacks with 120,000 instances and 254 features, to evaluate the IDS model in multi-class, the attacks included in the sample are presented in the Table 6. The procedure that was followed to implement this phase of evaluation and experiment was as follows:

- a. Preprocessing step in several stages:
 - At first, the data was cleaned by removing empty features and features with a constant value, the remaining features were 49.
 - Randomize the data: Shuffle the order of instances passed through it randomly (seed value is 33).
 - Remove Percentage: To fit the allocated memory for WEKA. The new sample consists of 7,500 instances and 49 features.
 - Remove attributes that have over 50% of missing value, the remaining attributes were 44.
- b. Feature selection based in to gain ratio evaluation and information gain evaluation, as we had done in the previous phase when the class was nominal, in order to compare the accuracy results for the nominal and numeric classes.
 - The remaining 36 features based on gain ratio attributes evaluation were (*wlan.fc.protected*, *wlan.fc.subtype*, *radio-tap.present.tsft*, *wlan.bssid*, *wlan.fc.type*, *wlan.fc.ds*, *wlan.duration*, *frame.number*, *wlan.sa*, *frame.len*, *wlan.radio.data.rate*, *wlan.ta*, *frame.time.relative*, *wlan.fc.retry*, *wlan.ra*, *wlan.radio.duration*, *wlan.da*, *ip.proto*, *radiotap.channel.flags.cck*, *wlan_radio.signal_dbm*, *radiotap.dbm_antsignal*, *wlan.fc.moredata*, *radiotap.datarate*, *ip.ttl*, *ip.src*, *frame.time_delta*, *frame.time_delta_displayed*, *ip.dst*, *wlan.fc.pwrmtg*, *wlan.fc.frag*, *frame.time*, *wlan.seq*, *ip.version*, *llc*, *wlan.fc.order*).
 - According to information gain attributes evaluator these features (*radiotap.timestamp.ts*, *frame.time.epoch*, *frame.time*, *frame.number*, *frame.time.relative*, *frame.len*, *wlan.radio.duration*) are copying the label, so we removed them to get the accurate result for the ML model.
 - The remaining 37 features were (*frame.time-delta*, *frame.time-delta-displayed*, *radiotap.channel.flags.cck*, *radiotap.channel.flags.ofdm*, *radiotap.channel.freq*, *radiotap.datarate*, *radiotap.dbm_antsignal*, *radiotap.length*, *radiotap.present.tsft*, *wlan.duration*, *wlan.bssid*, *wlan.da*, *wlan.fc.ds*, *wlan.fc.frag*, *wlan.fc.order*, *wlan.fc.moredata*, *wlan.fc.protected*, *wlan.fc.pwrmtg*, *wlan.fc.retry*, *wlan.fc.subtype*, *wlan.ra*, *wlan.sa*, *wlan.seq*, *wlan.ta*, *wlan.radio.channel*, *wlan_radio.data.rate*, *wlan_radio.frequency*, *wlan_radio.signal_dbm*, *wlan_radio.phy*, *llc*, *ip.dst*, *ip.proto*, *ip.src*, *ip.ttl*, *ip.version*, *Label*).
- c. Following the preparation of the data and the implementation of feature selection methods, we utilized various machine learning algorithms. Table 7 displays the learning algorithms performance utilizing the WEKA software, while Table 8 exhibits the learning algorithms performance utilizing AZURE software.

Table 6. Training and testing in AWID3 dataset (numeric class)

Attack type	Class Value	Traffic in the sample
Normal	0	20,000
Krack	1	20,000
Disas	2	20,000
SSDP	4	20,000
Malware	5	20,000
Total		120,000

3.3. Phase III: binary classification

In this phase, the class was converted to binary, where the normal traffic is represented by 0, and attack traffic is represented by 1. The sample consists of 40,000 instances and 254 attributes, 20,000 instances are normal traffic, and the other 20,000 are malicious traffic (4,000 instances of each attack; Krack, Kr00k, Disas, Malware, and SSDP). The procedure that was followed to implement this phase of evaluation and experiment was as follows:

- Preprocessing step in several stages

First, the data was cleaned by removing empty features and features with a constant value, the remaining features were 69. Randomize the data: randomly shuffle the order of instances passed through it. The random number generator is reset with the seed value whenever a new set of instances is passed in, we chose the seed value to be 33.

- String to nominal, converts a range of string attributes (unspecified number of values) to nominal (set number of values).

When applying different ML algorithms using WEKA on the sample that consists of 59 attributes and 40,000 instances, the results show very high accuracy in most algorithms except random tree where the correlation coefficient was in random tree 0.8631. However, to avoid these fitting and high accuracies in the dataset, we applied Relief (feature selection) which is an algorithm developed by Kira and Rendell in 1992 that takes a filter-method framework to feature selection that is notably sensitive to feature interactions. Figure 5 shows attributes distributed according to their sensitivity to the label. To avoid the overfitting that we get in the previous experiment, we removed attributes with sensitivity higher than 0.1, which are copying the label, then different ML algorithms were applied using AZURE on the remaining 39 attributes and 40,000 instances. Table 9 shows the learning algorithms performance using AZURE.

Table 7. The learning algorithms performance on gain ratio and info gain (numeric) using WEKA

Gain Ratio-Numerical				
Splitting data 70% train and 30% test				
Algorithm	Correlation coefficient	Mean absolute error	Relative absolute error	Root relative squared error
Decision Stump	0.8297	0.7723	51.39%	55.83%
Random Tree	0.8939	0.4455	29.65%	45.31%
10-fold cross-validation				
Algorithm	Correlation coefficient	Mean absolute error	Relative absolute error	Root relative squared error
Decision Stump	0.8282	0.7732	51.84%	56.03%
Random Tree	0.7005	0.795	53.30%	71.36%
Info Gain-Numerical				
Splitting data 70% train and 30% test				
Algorithm	Correlation coefficient	Mean absolute error	Relative absolute error	Root relative squared error
Decision Stump	0.6079	1.0661	71.19%	79.40%
Random Tree	0.9965	0.0268	1.79%	8.48%
10-fold cross-validation				
Algorithm	Correlation coefficient	Mean absolute error	Relative absolute error	Root relative squared error
Decision Stump	0.6079	1.0661	71.19%	79.40%
Random Tree	0.9958	0.0169	1.13%	9.14%

Table 8. The learning algorithms performance on gain ratio and info gain (numerical) using AZURE

Gain Ratio-Numerical				
Algorithm	Overall Accuracy	Average Accuracy	Precision	Recall
Multiclass Decision Forest	0.94972	0.98324	0.94972	0.94972
Multiclass Decision Jungle	0.89397	0.96466	0.9031	0.894
Multiclass Logistic Regression	0.99994	0.99998	0.9999	0.9999
Info Gain-Numerical				
Algorithm	Overall Accuracy	Average Accuracy	Precision	Recall
Multiclass Decision Forest	0.99133	0.99711	0.99133	0.99133
Multiclass Decision Jungle	0.92969	0.97657	0.9393	0.9296
Multiclass Logistic Regression	0.94381	0.98127	0.9569	0.9437

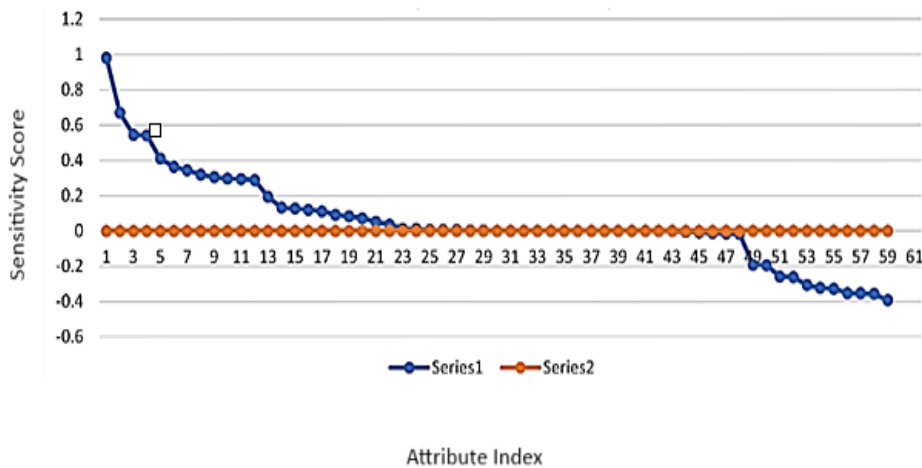


Figure 5. Relief feature selection

Table 9. The learning algorithms performance on Relief feature selection using AZURE

Algorithm		Accuracy	Precision	Recall	F1 Score
Two-Class	Logistic regression	0.994	0.998	0.927	0.961
Two-Class	Decision jungle	0.888	0.993	0.783	0.876
Two-Class	Decision forest	0.947	0.977	0.916	0.945
Two-Class	Boosted decision tree	0.968	1	0.614	0.76
Two-Class	Support vector machine	0.993	0.994	0.927	0.959
Two-Class	Locally deep support vector machine	0.995	1	0.938	0.968

Utilizing the AWID3 dataset, we conducted several experiments utilizing different machine-learning algorithms and feature selection techniques. The dataset has an abundance of features that closely connect with the label classes, as we discovered throughout the analysis of the results. In essence, certain features were very clear as to which class an instance belonged to. We used visualization tools, especially for decision trees, to confirm our findings and gain further understanding of the operation of our intrusion detection framework. These visualizations helped confirm the presence and influence of the highly correlated features on the accuracy of the model. Given the wide range of feature selection methods that are available (some of which we have already used), we decided to take a novel approach in this experiment. We used the Chi-squared, ANOVA, and Relief feature selection algorithms to combine three different feature selection methods into our research. Our goal was to leverage the strengths of each method and identify common ground among them. To this end, we focused on features that received high rankings across all three algorithms, considering them as especially valuable for our intrusion detection model. For a clearer visualization of our approach, please refer to Figure 6, where we elucidate the implementation of this concept.

The overlapping features are (*frame len, frame number, frame time epoch, frame time relative, ip proto, ip ttl, radio-tap channel flags cck, radiotap channel flags ofdm, radiotap channel freq, radiotap dbm antsignal, radiotap length, radio-tap present tsft, radiotap timestamp ts, tcp ack raw, tcp dstport, tcp flags push, tcp srcport, tcp time relative, wlan bssid, wlan da, wlan duration, wlan fc ds, wlan fc moredata, wlan protected, wlan fc retry, wlan fc subtype, wlan fc type, wlan ra, wlan radio channel, wlan radio data rate, wlan radio frequency, wlan radio phy, wlan radio signal dbm, wlan ta*). The remaining features are 25 (*frame.time, frame.time delta, wlan.fc.pwrmtg, wlan.fc.subtype, wlan.radio.duration, wlan.sa, wlan.seq, llc, ip.dst, ip.src, ip.version, tcp.ack, tcp.analysis, tcp.checksum, tcp.checksum.status, tcp.flags.syn, tcp.flags.ack, tcp.flags.fin, tcp.flags.reset, tcp.payload, tcp.seq, tcp.seq raw, tcp.time delta, Label*). Applying different ML and deep learning algorithms using WEKA and MATLAB on the remaining 25 attributes and 40,000 instances, the learning algorithms performance is presented in Table 10 for WEKA, and Table 11 for MATLAB.

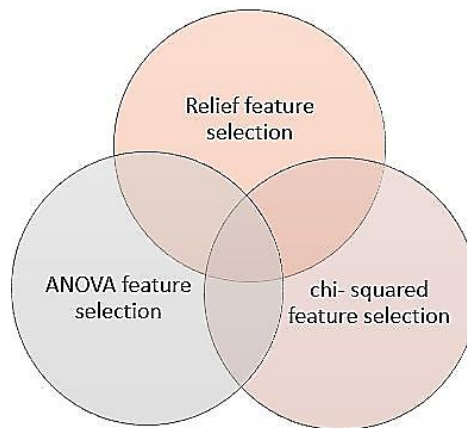


Figure 6. Overlapping between features selection algorithms

Table 10. The performance when overlapping between features selection using WEKA

Algorithm	Correlation coefficient	Mean absolute error	Relative absolute error	Root relative squared error
Decision Stump	0.9273	0.0762	15.2441 %	37.4972 %
Random Tree	0.9038	0.0784	15.6863%	42.9573%
Decision Table	0.9192	0.0771	15.4243 %	39.4434 %

Table 11. The performance when overlapping between features selection algorithms using MATLAB

Algorithm	Accuracy	True positive rate (TPR) for class 1	False negative rate (FNR) for class 1	True positive rate (TPR) for class 0	False negative rate (FNR) for class 0
Decision tree-fine tree	95.2%	92.2%	7.8%	98.2%	1.8%
Decision tree-medium tree	95.2%	92.2%	7.8%	98.2%	1.8%
Decision tree-coarse tree	94.6%	91.5%	8.5%	97.8%	2.2%
Ensemble classification-Boosted tree	99.0%	99.9%	.1%	98%	2%
Ensemble classification-Bagged tree	91.3%	84.2%	15.7%	98.3%	1.7%
Ensemble classification-subspace discriminant	86.7%	89.3%	10.7%	84.2%	15.8%
Naive Bayes	95.3%	98.4%	1.6%	92.2%	7.8%

3.4. Analysis of results for the three phases

In this section, we analyze the performance of the proposed classifier models, by comparing the results that we get in the previous section. The finding and results for multi-attack classifiers when the label class is nominal, show high accuracy, where the highest accuracy was for treesJ48 and logistic regression with 99%, and the lowest accuracy was for decision forest and decision jungle with 91% and 89% consecutively. Figure 7 and, for more detail, Figure 8, present the results for this phase when we used training and validation split percentage.

When comparing this feature selection algorithm when the class is numeric with nominal class, the results show decreasing in performance as shown in Figures 9 and 10, where the highest accuracy that we get when the class is numeric was for logistic regression too with 99%, and the lowest was for decision stump with 82%. The results show that the performance of the suggested model was affected in some ML algorithms when the class changed to a numeric label, and some were not affected as shown in Figure 11. Referring to Figure 11, logistic regression still gives the highest accuracy with 99%, while decision stump and random tree have the lowest accuracy with 82% and 89% consecutively. If we want to compare the accuracy of feature extraction methods that we have used, we have to look at Information gain attributes evaluation, which gives us good accuracy almost for all ML algorithms, and this performance did not affect when the label class is nominal or numeric, where random tree, decision forest, and treesJ48, gave the highest accuracy 99% while Decision-Stump gave the lowest accuracy 60%. As shown in Figure 12.

Figure 13 shows a comparison between the two feature selection methods. The accuracy results demonstrate stability and high performance for the ML algorithms used, indicating the effectiveness of both methods. However, the decision-stump algorithm consistently showed the lowest performance across both feature selection methods, highlighting its limitations compared to other algorithms.

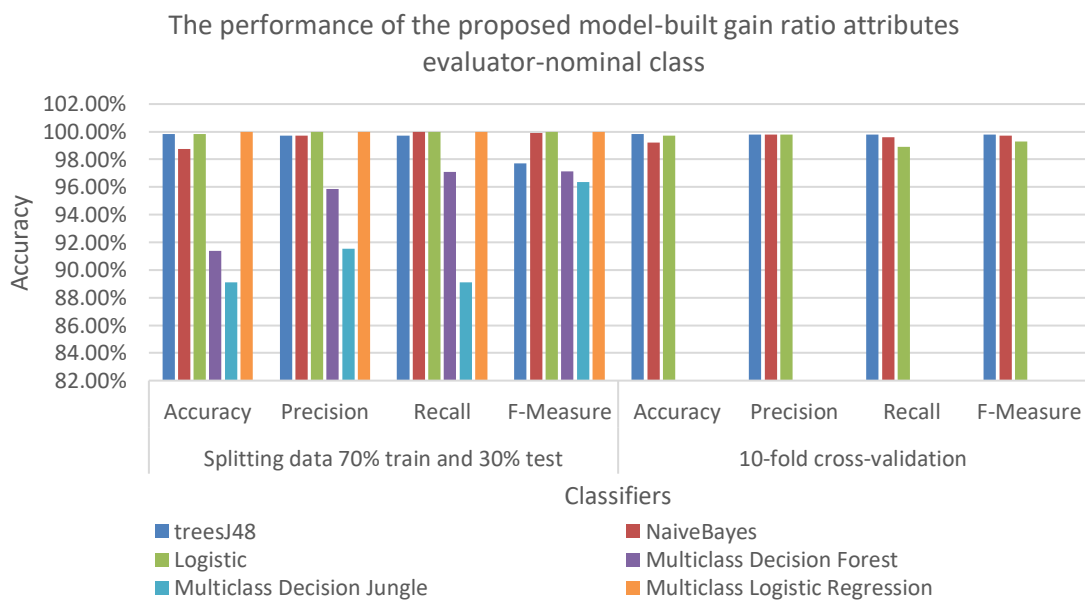


Figure 7. The performance of the proposed model-built gain ratio attributes evaluator-nominal class

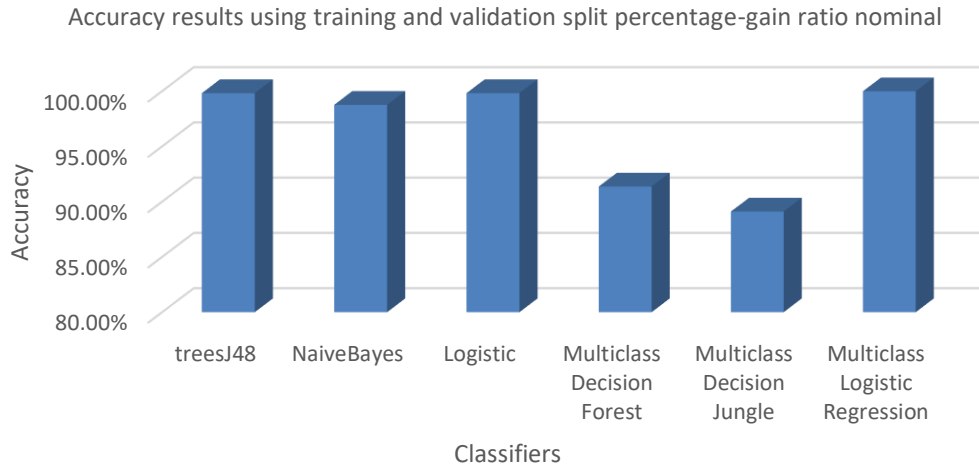


Figure 8. Accuracy results using training and validation split percentage-gain ratio- nominal

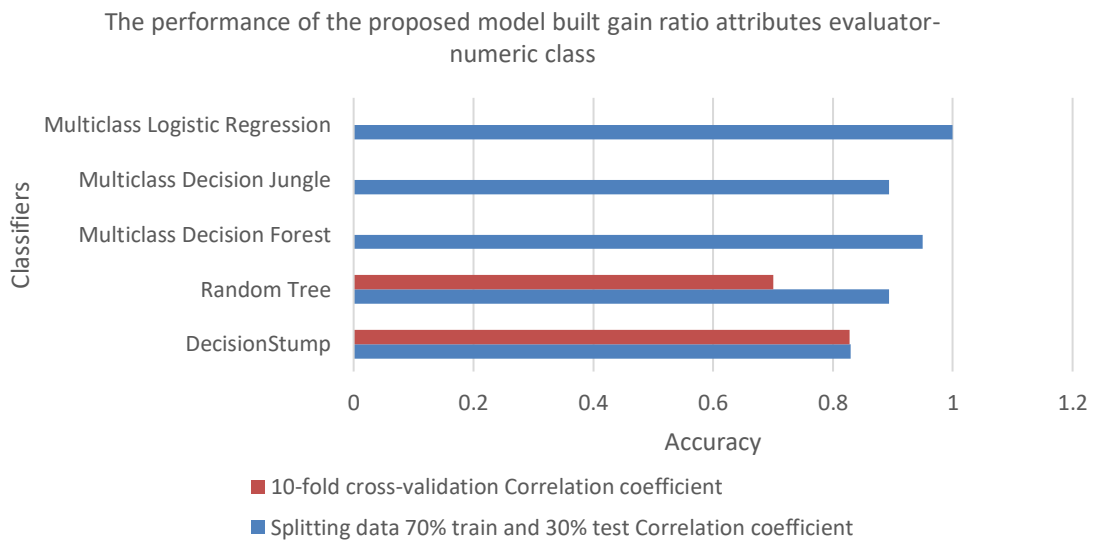


Figure 9. The performance of the proposed model-built gain ratio attributes evaluator-numeric class

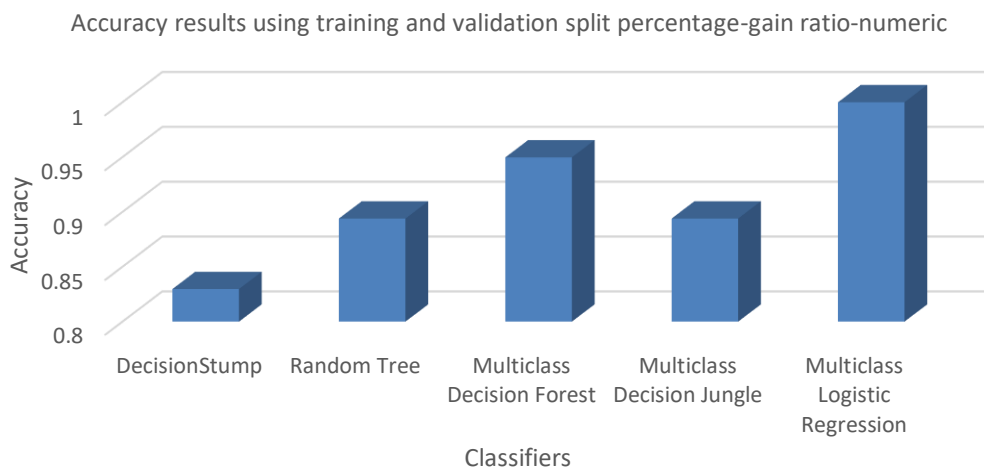


Figure 10. Accuracy results using training and validation split percentage-gain ratio-numeric

Comparison the performance when the class is nominal and numeric on gain ratio evaluator

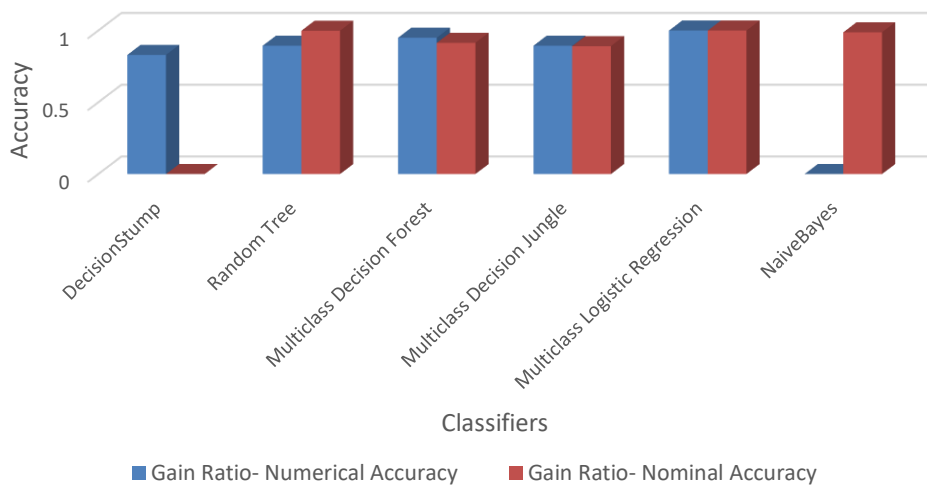


Figure 11. Comparison the performance when the class is nominal and numeric on gain ratio evaluator

Comparison the performance when the class is nominal and numeric on information gain

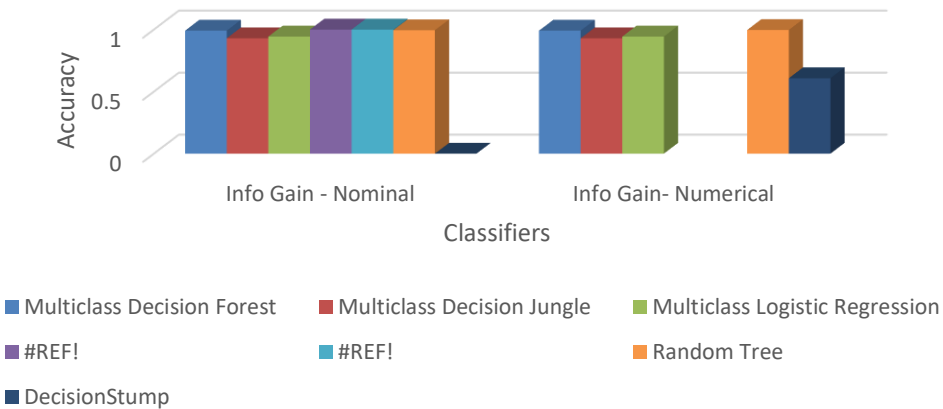


Figure 12. Comparison the performance when the class is nominal and numeric on information gain

3.5. Analysis of binary classification; phase III

In this phase, three different feature selection algorithms were used as we mentioned in the previous section; (chi-squared feature selection, ANOVA feature selection, and Relief feature selection), Figure 14 shows the performance of 14 different ML and deep learning algorithms that used in the proposed classifier. Logistic regression and boosted tree had the highest accuracy while KNN had the lowest accuracy.

3.6. Comparison of our findings with other studies

Table 12 summarizes the studies mentioned in Section 2, providing a comprehensive overview of the various attacks, feature selection methods, and approaches along with their respective accuracies. This table highlights the effectiveness of different machine learning (ML) and deep learning (DL) approaches in detecting various types of cyber-attacks. Notably, our work achieves the highest accuracy with a multi-class accuracy of 99.9% and a binary accuracy of 99%, demonstrating the efficacy of our feature selection and classification methods.

Comparison of the performance between the two feature selection methods

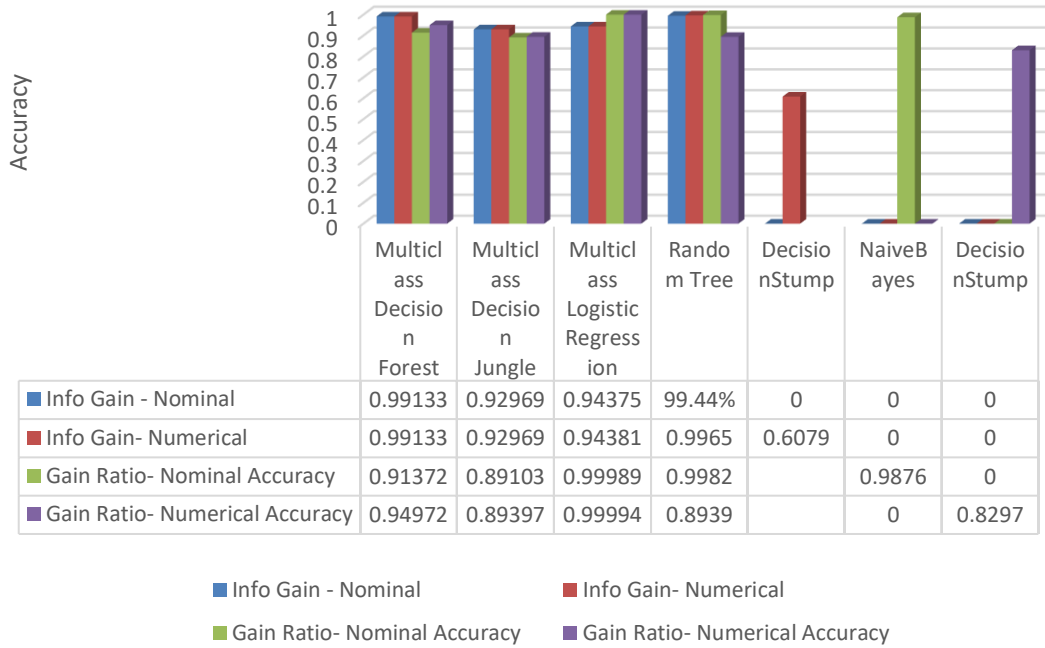


Figure 13. Comparison of the performance between the two feature selection methods

Table 12. Comparison of our findings with other recent studies that used the AWID3 dataset

Reference	Attack	Feature Selection	Approach and Accuracy
[7]	Attacks on application layer (Botnet, Malware, SSH, SQL Injection, SSDP amplification, and Web-site spoofing)	Yes	ML: 98.7% DL: 97.86% F.S: 99%
[8]	Flood category contains Deauth, Disas, Assoc, and Kr00k attacks. Impersonation contains: RogueAP, Evil twin, and Krack	Yes	ML and DNN: 99.96%
[9]	De-authentication, Rogue AP, Evil twin, Krack, and SSID	No	ML: 99.7%
[10]	All attacks	No	SVM: 79% DT: 99.8%
Our Work	Krack, Kr00k, Dis, Malware and SSDP	Yes	Multiclass: 99.9% Binary: 99%

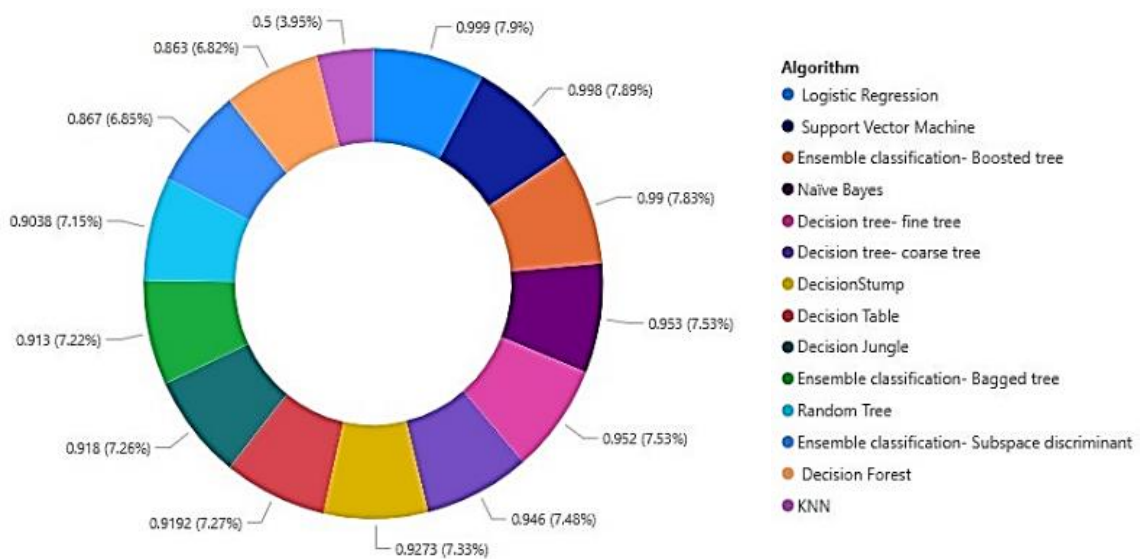


Figure 14. The proposed model performance for binary class

4. CONCLUSION

Due to improvements in network connectivity and the growing user base, 5G networks are becoming more and more popular, which has attracted a lot of attention. This research has concentrated on creating and assessing a novel model utilizing the newest wireless dataset, AWID3, as wireless network security is becoming a crucial concern. The complexity of high dimensionality and the natural imbalance between benign and malicious traffic are two significant issues that this research dealt with.

These difficulties were able to overcome by utilizing feature selection approaches. Three unique phases define our suggested WIDS: multi-nominal class, multi-numeric class, and binary class. The performance of the proposed model against five forms of attacks: Disassociation, Krack, Kr00k, Malware, and SSDP. Disassociation, Krack, and Kr00k assaults are 802.11 MAC layer attacks; malware operates at a higher layer; and SSDP uses rogue or compromised local clients to direct attacks toward external targets. Using machine learning-based methodologies, the experiments repeatedly showed great accuracy in identifying these attacks. Particularly, an accuracy of 99% was obtained throughout all three stages, with the lowest recorded accuracies for each phase being 89.1%, 60%, and 86.7%, respectively.

By conducting this research, we highlight the importance of understanding the inherent characteristics of wireless datasets and their significant influence on the effectiveness of intrusion detection models. The complexity of wireless datasets will likely be addressed in the next projects, which will improve and advance the capabilities of wireless IDS.




REFERENCES

- [1] R. Agrawal, "Comparison of different mobile wireless technology (from 0G to 6G)," *ECS Transactions*, vol. 107, no. 1, pp. 4799–4839, Apr. 2022, doi: 10.1149/10701.4799ecst.
- [2] Y. Zhou *et al.*, "Blockchain for 5G advanced wireless networks," in *2022 International Wireless Communications and Mobile Computing*, May 2022, pp. 1306–1310, doi: 10.1109/IWCMC55113.2022.9825182.
- [3] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, Jan. 2020, doi: 10.1109/JIOT.2019.2948888.
- [4] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, "Cyber threats to industrial IoT: a survey on attacks and countermeasures," *Internet of Things*, vol. 2, no. 1, pp. 163–186, Mar. 2021, doi: 10.3390/iot2010009.
- [5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Dec. 2019, doi: 10.1186/s42400-019-0038-7.
- [6] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of internet of things (IoT): A survey," *Journal of Network and Computer Applications*, vol. 161, Jul. 2020, doi: 10.1016/j.jnca.2020.102630.
- [7] J. Brownlee, "Introduction to dimensionality reduction for machine learning," *Machine Learning Mastery*, pp. 1–5, 2020.
- [8] E. Chatzoglou, G. Kambourakis, C. Smiliotopoulos, and C. Koliass, "Best of both worlds: detecting application layer attacks through 802.11 and non-802.11 features," *Sensors*, vol. 22, no. 15, Jul. 2022, doi: 10.3390/s22155633.
- [9] E. Chatzoglou, G. Kambourakis, C. Koliass, and C. Smiliotopoulos, "Pick quality over quantity: expert feature selection and data preprocessing for 802.11 intrusion detection systems," *IEEE Access*, vol. 10, pp. 64761–64784, 2022, doi: 10.1109/ACCESS.2022.3183597.
- [10] R. Saini, D. Halder, and A. M. Baswade, "RIDS: real-time intrusion detection system for WPA3 enabled enterprise networks," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Dec. 2022, pp. 43–48, doi: 10.1109/GLOBECOM48099.2022.10001501.
- [11] C. Zheng *et al.*, "Automating in-network machine learning," *arXiv:2205.08824*, May 2022.
- [12] S. C. Sethuraman, S. Dharamdara, and V. Vijayakumar, "Intrusion detection system for detecting wireless attacks in IEEE 802.11 networks," *IET Networks*, vol. 8, no. 4, pp. 219–232, Jul. 2019, doi: 10.1049/iet-net.2018.5050.
- [13] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019, doi: 10.1109/JIOT.2019.2926365.
- [14] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Computing and Applications*, vol. 32, no. 10, pp. 6125–6137, Feb. 2020, doi: 10.1007/s00521-019-04103-1.
- [15] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Computational and Mathematical Organization Theory*, vol. 25, no. 3, pp. 319–335, Mar. 2019, doi: 10.1007/s10588-018-9266-8.
- [16] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, no. 1, Nov. 2020, doi: 10.1186/s40537-020-00379-6.
- [17] C. Kubik, S. M. Knauer, and P. Groche, "Smart sheet metal forming: importance of data acquisition, preprocessing and transformation on the performance of a multiclass support vector machine for predicting wear states during blanking," *Journal of Intelligent Manufacturing*, vol. 33, no. 1, pp. 259–282, Jun. 2022, doi: 10.1007/s10845-021-01789-w.
- [18] E. Chatzoglou, G. Kambourakis, and C. Koliass, "Empirical evaluation of attacks against IEEE 802.11 enterprise networks: the AWID3 dataset," *IEEE Access*, vol. 9, pp. 34188–34205, 2021, doi: 10.1109/ACCESS.2021.3061609.
- [19] C. Koliass, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 184–208, 2016, doi: 10.1109/COMST.2015.2402161.
- [20] A. Arora, "Preventing wireless deauthentication attacks over 802.11 networks," *arXiv:1901.07301*, Nov. 2018.
- [21] M. A. C. Aung and K. P. Thant, "IEEE 802.11 attacks and defenses," *Proceedings of the 17th International Conference on Computer Application (ICCA)*, Yangon, Myanmar, 2019.
- [22] R. Cheema, D. Bansal, and S. Sofat, "Deauthentication/disassociation attack: implementation and security in wireless mesh networks," *International Journal of Computer Applications*, vol. 23, no. 7, pp. 7–15, Jun. 2011, doi: 10.5120/2901-3801.
- [23] B. Lee, "Stateless re-association in WPA3 using paired token," *Electronics*, vol. 10, no. 2, p. 215, Jan. 2021, doi: 10.3390/electronics10020215.
- [24] T. Zhou, Z. Cai, B. Xiao, Y. Chen, and M. Xu, "Detecting rogue AP with the crowd wisdom," in *Proceedings - International*




- Conference on Distributed Computing Systems*, Jun. 2017, pp. 2327–2332, doi: 10.1109/ICDCS.2017.31.
- [25] C. P. Kohlios and T. Hayajneh, “A comprehensive attack flow model and security analysis for Wi-Fi and WPA3,” *Electronics (Switzerland)*, vol. 7, no. 11, p. 284, Oct. 2018, doi: 10.3390/electronics7110284.
- [26] D. Schepers and A. Ranganathan, “Framing frames: bypassing Wi-Fi encryption by manipulating transmit queues,” in *32nd USENIX Security Symposium*, 2023, pp. 53–68.
- [27] S. K. Wanjau, G. M. Wambugu, and G. N. Kamau, “SSH-brute force attack detection model based on deep learning,” *International Journal of Computer Applications Technology and Research*, vol. 10, no. 1, pp. 42–50, Jan. 2021, doi: 10.7753/ijcatr1001.1008.
- [28] M. M. Najafabadi, T. M. Khoshgoftaar, C. Kemp, N. Seliya, and R. Zuech, “Machine learning for detecting brute force attacks at the network level,” in *2014 IEEE International Conference on Bioinformatics and Bioengineering*, Nov. 2014, pp. 379–385, doi: 10.1109/BIBE.2014.73.
- [29] A. A. Ahmed, W. A. Jabbar, A. S. Sadiq, and H. Patel, “Deep learning-based classification model for botnet attack detection,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 7, pp. 3457–3466, Mar. 2022, doi: 10.1007/s12652-020-01848-9.
- [30] O. Aslan and R. Samet, “A comprehensive review on malware detection approaches,” *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.
- [31] X. Liu *et al.*, “A multi-location defence scheme against SSDP reflection attacks in the internet of things,” in *Communications in Computer and Information Science*, vol. 1137, Springer Singapore, 2019, pp. 187–198.
- [32] W. G. J. Halfond, J. Viegas, and A. Orso, “A classification of SQL-injection attacks and countermeasures,” *International Symposium on Signals, Systems, and Electronics*, 2006, pp. 1–11.
- [33] P. Shrivastava, M. S. Jamal, and K. Kataoka, “EvilScout: detection and mitigation of evil twin attack in SDN enabled WiFi,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 89–102, Mar. 2020, doi: 10.1109/TNSM.2020.2972774.
- [34] A. A. Ahmed and N. A. Abdullah, “Real time detection of phishing websites,” in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct. 2016, pp. 1–6, doi: 10.1109/IEMCON.2016.7746247.
- [35] J. Susan and P. Subashini, “Deep learning inpainting model on digital and medical images-A review,” *International Arab Journal of Information Technology*, vol. 20, no. 6, pp. 919–936, 2023, doi: 10.34028/iajit/20/6/9.

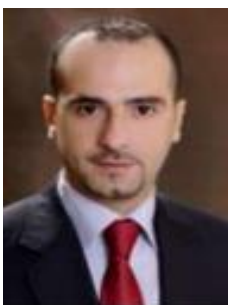
BIOGRAPHIES OF AUTHORS






Zaher Salah    received his Ph.D. degree in computer science from the University of Liverpool, UK, in 2014, his M.Sc. degree in computer science from Yarmouk University, Jordan, in 2004, and his B.Sc. degree in computer science from University of Jordan, Jordan, in 2001. He is currently an associate professor in the Information Technology Department of The Hashemite University, Zarqa, Jordan. His research interests include machine learning, cyber security, information retrieval, opinion mining, sentiment analysis, biometrics, digital image and analysis, and pattern recognition. He can be contacted at email: zaher@hu.edu.jo.






Esraa Elsoud    received the B.Sc. degree in electrical engineering from The Hashemite University, Jordan, in 2013 and M.Sc. in cyber security from The Hashemite University in 2023. Eng. Esraa’s current research interests include cyber security, machine learning, big data and mobile network. She is currently a lecturer in Zarqa University, Zarqa, Jordan. She can be contacted at email: eabuelsoud@zu.edu.jo.






Waleed Al-Sit    received his Ph.D. degree in electrical engineering and electronics from the University of Liverpool, UK, in 2015, his M.Sc. degree in electrical and computer engineering from New York Institute of Technology (NYIT), Jordan, in 2008, and his BSc degree in computer engineering from Mu’tah University, Jordan, in 2006. He is currently an associate professor in the Department of Computer Engineering, Mu’tah University, Al-Karak, Jordan; Higher Colleges of Technology, Dubai, UAE. His research interests include machine learning, cyber security, sentiment analysis, networking, and pattern recognition. He can be contacted at email: w_sitt@mutah.edu.jo.






Esraa Alhenawi    is a dedicated academic with a distinguished background in computer science. She earned her Ph.D. in computer science with honors in 2022, following her master's degree in the same field in 2015. She also holds a bachelor's degree in computer engineering from 2010, currently serving as an assistant professor at Zarqa University since September 2023. She can be contacted at email: ealhenawi@zu.edu.jo.



Fuad Alshraideh    assistant professor in the Faculty of Science and Information Technology at Zarqa University, Jordan. He earned his Ph.D. in computer science/software engineering in 2023, adding to his extensive academic qualifications that include a master's degree in computer science from 2013 and a bachelor's degree in computer science from 2006. He has been a faculty member at Zarqa University since 2023 and is actively involved in the field of information technology and software engineering. Dr. Al-Shraideh can be reached at falshraideh@zu.edu.jo.



Nawaf Alshdaifat    received the B.Sc. in computer science from Al-Albaysat University, Jordan, M.Sc. degrees in computer science from University of Jordan, Jordan, in 2002 and 2011, respectively and the Ph.D. degree in computer sciences from University Sains Malaysia, Malaysia, in 2023. His research interests include deep learning and tracking objects. He can be contacted by this email: n.alshdaifat@asu.edu.jo.