Article

Not peer-reviewed version

# Enhancing Intrusion Detection in 5G and IoT Environments: A Comprehensive Machine Learning Approach Leveraging AWID3 Dataset

Zaher Salah * and Esraa Abu Elsoud

Posted Date: 24 July 2023

doi: 10.20944/preprints202307.1565.v1

Keywords: Network security; 5G; Intrusion detection; Machine learning; Feature selection; Wi-Fi networks

*Article*

# Enhancing Intrusion Detection in 5G and IoT Environments: A Comprehensive Machine Learning Approach Leveraging AWID3 Dataset

**Zaher Salah [1,*] and Esraa Abu Elsoud [2]**

[1] Department of IT, The Hashemite University, Jordan
[2] Department of CIS, The Hashemite University, Jordan; 2070606@hu.edu.jo
* Correspondence: zaher@hu.edu.jo

**Abstract:** Internet users have significantly increased as a result of the spread of Internet of Things (IoT) technologies and 5G networks. But these developments also make people more susceptible to cybercrime. Intrusion detection systems (IDSs), which protect against cyber threats and facilitate early response, have emerged as crucial security measures to handle this expanding risk. This study intends to present a comprehensive review of IDS, how it interacts with machine learning (ML), and develop a suitable approach for attack detection in 5G and IoT environments. To accomplish this, we leverage the AWID dataset, which is the first wireless traffic dataset specifically designed for security purposes, focusing on the IEEE 802.11 standard and developed to the AWID3 dataset. In this research, we suggest a powerful machine-learning framework for wireless system intrusion detection. We perform evaluations in three stages, covering scenarios for multiple nominal classes, multiple numeric classes, and binary classes. In order to improve the performance of the intrusion detection model, we also use feature selection approaches. Additionally, we offer a model that incorporates the outcomes of three feature selection techniques, highlighting how crucial it is to comprehend the features present in wireless datasets. Our experiments demonstrate how a machine learning-based approach can detect attacks with a high level of accuracy. In particular, the boosted decision tree performs best when overlapping feature selection procedures, whereas the Logistic Regression approach obtains the maximum accuracy of 99% in the first two phases. By providing a comprehensive framework for identifying attacks in 5G and IoT contexts using machine learning approaches, this research makes a contribution to the field of intrusion detection. The results underline how important it is to comprehend wireless dataset characteristics and highlight the possibility of ML-based methods for attaining highly accurate intrusion detection.

**Keywords:** Network security; 5G; Intrusion detection (IDS); Machine learning (ML); Feature selection; Wi-Fi networks

## 1. Introduction

The next generation of wireless networks will require a unified platform to support the huge numbers of devices, users, and services with different data rates and latency requirements. Current wireless technologies like 3G and 4G-LTE have several limitations that restrict any possible enhancement of the systems to meet these demands. Accordingly, researchers have been developing an advanced wireless communication technology called (5G) to satisfy the aforementioned requirements. After several scientific research, it was found that the fifth-generation technology has limitations too, it can't be used for long-distance communication or low-power wide-area technology. This means that the available deployed communication technologies will not be able to completely and efficiently keep up with future requirements. Furthermore, by 2030, it is predicted that a more advanced digital society supported by limitless wireless connectivity will have emerged [1]. 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. In order to create comprehensive IoT networks, these devices will be endogenously fully equipped with IoT modules that enable D2D communication with one another [2]. Furthermore, RAT will be supported by 5G to connect these devices. 5G networks will give the opportunity to introduce new radio technologies, like NOMA, mmWave, massive MIMO, and other several IoT communication technologies.

A comprehensive security policy is the starting point for managing the countermeasures need to secure wireless networks. Technical countermeasures which help in wireless security environments include hardware and software. Hardware counter- measure like Smart cards, VPNs, and biometrics are hardware solutions. While proper AP configuration, software patches, authentication, intrusion detection systems (IDS), and encryption are all examples of software countermeasures [3].

The intrusion detection system (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. The machine learning field has enabled different

Towards Effective Machine Learning Framework for Intrusion Detection in Wireless Networks paths that are effective in handling network intruders. Therefore, the employment of ML tools in 5G systems has attracted much interest from international projects and research. Intrusion Detection Systems (IDS) and the use of machine learning algorithms to detect illegal network access have become crucial components of cybersecurity solutions. By examining behavior patterns, these systems continuously improve their accuracy and performance over time by training themselves using labeled datasets [4].In this situation, it is possible to identify zero-day attacks by effectively identifying malicious traffic patterns, making early attack detection possible. Additionally, by utilizing adaptive learning capabilities, the implementation of machine learning techniques offers significant potential for identifying new assaults and enabling intelligent handling of emerging threats. This capacity plays a crucial role in developing strong security protocols for wireless networks and Internet of Things (IoT) devices, ensuring improved protection and resilience against changing cybersecurity risks [5].

The development of an effective wireless intrusion detection system has some major difficulties, including:

Handling High-Dimensional Data: Wireless IDS frequently runs into problems processing high-dimensional data, when the dataset contains a lot of features. Techniques for dimensionality reduction are used to get around this problem. By reducing the number of input variables in the training data, these techniques can present the information in a lower-dimensional subspace that captures the most important details while reducing noise and redundancy [6].

Dataset Balancing for Better Detection Performance: It's important to balance the dataset to avoid over-fitting, which occurs when the IDS becomes excessively specialized to the properties of the training data and has poor generalization on unseen data. The IDS can perform better in detecting intrusions across distinct classes by balancing the dataset, either by undersampling the majority class or oversampling the minority class.

Real-Time Abnormal Traffic Detection: Using the created model, the ability to identify abnormal traffic in real-time is a crucial aspect of wireless IDS. By allowing for prompt reactions to possible threats, real-time detection helps to lessen the effect of attacks. 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 respond to these questions, we implemented the experiment in three phases; using multi-nominal class, multi-numeric class, and binary class. In addition to using feature selection techniques to enhance the model's performance.

The remaining part of the research is divided into the following sections: Literature Review in Section II The existing literature on intrusion detection systems (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 III. 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 IV, 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 V summarizes the main conclusions and contributions of the study before drawing a conclusion. 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.

## 2. Literature Review

The revolution of communication networks led to an exponentially increasing of IoT devices connected to Wi-Fi networks, these devices create a massive scale of data traffic, which is not all benign traffic where intruders may exploit the massiveness of data to send their malicious data to users' networks, this will create a challenge in detection of such attacks. Feature selection is used to reduce the amount of data for intrusion detection model classifiers by removing noisy information and choosing the best features in the data. Which participates in improving the IDS performance and solving these challenges. We will focus in this research on the AWID dataset which is a Wi-Fi network intrusion benchmark dataset was introduced due to the lack of the dataset in wireless intrusion detection systems (WIDSs) where the oldest datasets were about IDSs in general

Chatzoglou et al. [7] the authors, 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, SSDP amplification, and website spoofing using the AWID3 benchmark dataset. They removed several aspects from the initial feature set that they felt were ineffective in identifying application layer threats in order 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. Botnets, 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, Chatzoglou et al. were able to reach an accuracy rate of 98.71%. They also looked into 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%.

In [8] authors intended to determine the bare minimum set of classifier features that could be used with any 802.11 imple- mentation 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. Chatzoglou et al. divided instances in the AWID3 dataset into three categories: Normal, Flood, and Impersonation. attacks like Deauth, Disas, Assoc, and Kr00k were featured in the Flood category, whilst RogueAP, EvilTwin, and Krack assaults were included in the Impersonation category. Actually, the authors' combined average accuracy for deep learning and machine learning (ML) methods was 99.96%. Their research proved the usefulness of the chosen features as well as the capability of ML and deep learning approaches for reliably identifying and categorizing network threats. The research results support the improvement of intrusion detection systems in 802.11-based networks.

Saini et.al [9] focused on the WPA3 protocol, which is essential for assuring Wi-Fi security. They presented a two-stage technique in their research to handle the problem of intrusion detection within a network. They were successful in detecting network threats with a remarkable accuracy rate of over 99%. By utilizing the strength of machine learning (ML) techniques, which were essential in accurately identifying and classifying invasive actions within the network. The results of their research reaffirm the importance of ML in boosting network security within the WPA3 protocol framework and advancing intrusion detection approaches.

C Zhang et al. investigated how to map machine learning algorithms to programmable network devices. Furthermore, state- of-the-art and newly proposed in-network ML algorithms are evaluated and compared in terms of functionality, resources, scalability, and throughput. They used six datasets, including KDD99 and AWID3, for Intrusion Detection purposes. Their accuracy ranged from 97.47% for decision trees to 49.37% for KNN [10].

Moving toward some popular research that detects network intrusion using various machine learning algorithms,

S.C. Sethuraman 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 [11]. In [12] authors proposed a three-layer IDS that employs a supervised methodology to identify a variety of well-known network-based cyberattacks on IoT 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 achieved an accuracy of 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, the authors in [13] proposed the Modified Binary Grey Wolf Optimization feature selection algorithm (MBGWO). In order to enhance IDS performance, the suggested algorithm is based on binary Greywolf optimization. After applying the SVM algorithm to the NSL- KDD dataset, they obtained a 99.22% accuracy result. Kasongo and Sun [?] suggested a wireless IDS system utilizing feature extraction and a feed-forward deep neural network. They decided to use the UNSW-NB15 and AWID datasets. Their results have been contrasted with those of common machine learning algorithms, including Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), and k-Nearest Neighbor (kNN). The experimental studies are divided into four categories: binary and multiclass attacks, full features, and selected features. The AWID dataset's feature set was condensed to 26 using the Extra Trees (ET) technique. Both datasets made up 20% of the AWID-CLS dataset: the training dataset included 359,115 instances, while the test dataset had 115,128 instances. The suggested model's accuracy for binary classification was 98.6% on the validation data and 98.69% on the test data. The best accuracy in the case of multiclass classification was 98.47% on the validation data and 98.59% on the test data. These findings demonstrate how well the model performs when correctly categorizing cases into the appropriate category. It's interesting to note that the research showed that accuracy increased when the number of attributes was decreased. The suggested model's accuracy for binary classification on the validation data and test data was an excellent 99.67% and 99.66%, respectively. In the same way, the model's multiclass classification accuracy was 99.78% on the validation data and 99.77% on the test data. These results highlight how feature reduction can improve the model's accuracy and performance.

In the area of wireless Intrusion Detection Systems (IDS), the AWID2 dataset is an essential resource. It runs within a WEP-based architecture and consists of a sizable number of packets. With more than 150 different features, this dataset has established itself as a key resource in the literature on wireless IDS. The AWID dataset has undergone enhancements and revisions to further increase its capabilities, leading to the creation of AWID3. This advancement was made possible by carefully gathering and examining the remnants of cyberattacks targeted at the IEEE 802.1X Extensible

Authentication Protocol (EAP) environment. AWID3 provides a more specialized and contextually relevant dataset for researching and developing wireless IDS techniques by concentrating on this specific environment.

## 3. Methodology

This section outlines the process for implementing the Framework for Intrusion Detection in Wireless Networks using machine learning techniques.

*Prepossessing*

Preprocessing is a very helpful and essential step for obtaining the correct data required to build a classifier, as shown in several types of research such as [14–16]. Data preprocessing, which aims to convert the raw data into a format that is simpler and more efficient to use for subsequent processing steps, is a crucial step in the knowledge discovery process because quality decisions must be based on quality data. Thus, the preprocessing procedures were carried out on the AWID3 dataset. The AWID3 consist of 13 CSV files with (36,913,503) instances, (30,387,099) normal traffic, and (6,526,404) malicious ones. That was explored and well understood. According to that, the procedure was followed as below:

1- Choosing a sample that includes multi-attacks with 120,000 instances and 254 features. 2- Removing empty features and features with a constant value.

Randomize: 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.

Replaces all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data.

Conversion of data type: convert string attributes to nominal. Figure 1 shows the preprocessing steps followed in AWID3 dataset.

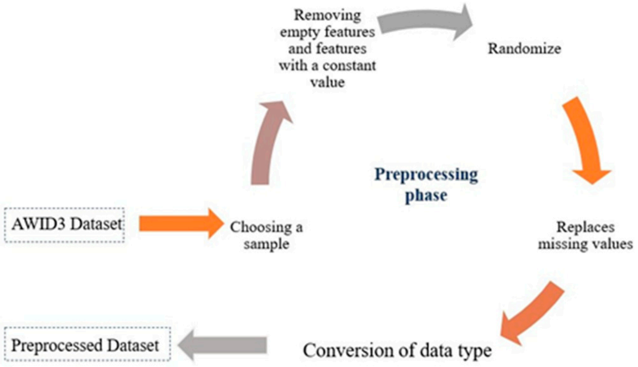


**Figure 1.** Preprocessing steps followed in AWID3.

*Structure of AWID3 Dataset*

Wireless technologies have increased rapidly in recent years. While serious efforts have been made to secure these tech- nologies, most security measures have proven inadequate in practice. The AWID project aims to provide a solid basis for researchers to develop robust security mechanisms for current and future generations of wireless networks by providing tools, methodologies, and datasets, as the previous datasets weren't specific to wireless networks.

WiFi (IEEE 802.11) has been replaced as the standardized technology for linking digital devices in wireless LAN due to the popularity of smart devices such as smartphones, smart watches, tablets, and Internet of Things devices. WiFi is frequently used in critical locations as well as in homes, businesses, and organizations. Unsurprisingly, extensive academic research has focused on 802.11 protocol security as well as WiFi network security. With frequent modifications and corrective actions, vulnerabilities have been found in even the most recent versions of the software although these vulnerabilities have been existing for more than 20 years. Security in wireless technology is a major issue that has long gone unresolved. External security measures should therefore be used as crucial elements of 802.11 wireless networks for preventing known or unknown attacks [17].

AWID dataset was extracted in 2016 then it has been developed into a new version in 2021 called AWID3. The IEEE 802.1X Extensible Authentication Protocol (EAP) environment uses a number of different attacks, and the publicly available, free dataset known as AWID3 collects and analyzes the traces of these attacks. It provides the first analysis of the IEEE 802.11w standard, which is required for hardware that has received WPA3 certification. AWID3 is anticipated to be crucial for designing and evaluating intrusion detection systems. The AWID dataset has 254 features in CSV format, including 253 generic features and one additional feature for labeling. The MAC and application layers both contain the extracted features. The AWID3 dataset was captured and collected using 16 different physical and virtual machines. The dataset consists of (36,913,503) instances of both (6,526,404) malicious and (30,387,099) normal traffic. 13 different attack types are present in the malicious traffic. The dataset is thoroughly described in the following subsections.

### *AWID3 feature description*

This type of study helps in identifying the best algorithms which can effectively work in detecting cyberattacks on wireless systems. The AWID3 dataset is used to display an in-depth analysis of machine learning classifier experiments. We imple- mented 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, 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 Feature Description** | |
|---|---|---|
| radiotap.dbm-antsignal | wlan-radio.signal-dbm | tcp.checksum |
| tcp.payload | wlan.duration | frame.time-delta-displayed |
| frame.time-delta | frame.time | tcp.time-relative |
| radiotap.channel.freq | wlan.fc.moredata | wlan-radio.frequency |
| wlan-radio.channel | wlan.fc.ds | wlan.fc.type |
| wlan.fc.protected | radiotap.channel.flags. cck | wlan.fc.subtype |
| wlan.fc.pwrmgt | wlan-radio.phy | radiotap.channel.flags. ofdm |
| radiotap.present.tsft | wlan.ra | radiotap.length |
| wlan.fc.retry | wlan.ta | wlan.bssid |
| wlan.sa | llc | ip.version |
| ip.proto | tcp.checksum.status | ip.ttl |
| ip.src | tcp.flags.reset | tcp.flags.syn |
| tcp.flags.fin | tcp.flags.ack | tcp.flags.push |
| frame.number | frame.len | frame.time-relative |
| wlan.sa | tcp.ack | tcp.analysis |
| tcp.seq | tcp.seq-raw | tcp.time-delta |

doi:10.20944/preprints202307.1565.v1

*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 [18], while AWID3 divides the attacks into four categories [17]:

1) 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 aren't being stopped. It can be divided into two categories: key re-installation and denial of service (DoS). Denial of service makes an effort to interfere with the connection between the key units of an 802.11 network, including Station (STA) and Access Point (AP), by attacking particular devices or putting more emphasis on the resources of the network and the connected devices on this network. The AWID3 dataset contains almost all of 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 system. There are six types of 802.11-specific attack categories: Deauthentication, Disassociation, Re-association, Rogue AP, Krack, and Kr00k.

2) 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.

3) Attacks against external nodes.

Attacks of this type typically involve a small number of actions that are started by corrupted 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.

4) 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.

**Table 2.** Attacks 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 |

Deauthentication Attack: To establish a connection between the client and the AP the client has to associate with the AP and must finish the authentication process before exchanging data. If the client wants to disconnect, he has to submit a disassociation frame to the AP. Alternatively, in case of 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 [19].

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 [20]. 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 [21]).

Re-association Attack: Reassociation occurs if a wireless client suddenly switches to another access point. While roaming is involved, the new and old access points connect to each other across the wired network to transfer wireless client data. Otherwise, the association and re-association procedures are the same. When a wireless client roams to a different access point, it uses the re-association process to inform the 802.11 networks that it has moved to a new location. The wireless client transmits to the new access point a re-association packet that identifies the old access point. To find out if the wireless client was previously connected, the new access point uses a wired connection to communicate with the old access point. If the wireless client was previously associated, the new access point sends a re-association response frame; if not, it sends a disassociation response frame. After sending the re-association response, the new access point makes contact with the old access point over the wired channel to complete the re-association procedure. The old access point's buffered frames are transmitted to the new access point. The new access point starts processing wireless client frames after the re-association procedure is complete [22].

Rogue AP Attack: WiFi 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 [23].

Krack Attack: The Krack attack has been noted as a potential security risk to the current encryption techniques used to preserve and protect Wi-Fi networks for the past 15 years. Publicly available information on the Krack attack includes information about the attack itself. There is no guarantee that every device will have a patch and be protected from these attacks coming from any networked point [24].

Kr00k Attack: Some WiFi traffic that has been encrypted with WPA2 can be decrypted by a vulnerability called Kr00k. Security company ESET discovered the vulnerability in 2019. According to ESET, this loophole affects more than a billion devices. Devices with Wi-Fi chips that have not yet received a patch from Broadcom or Cypress are vulnerable to Kr00k. The majority of modern Wi-Fi-enabled devices, including smartphones, tablets, laptops, and Internet of Things (IoT) devices, use these Wi-Fi chips [25].

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. In order 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. [26].

Machine learning techniques have been used to detect automated network-level brute force attacks using flow data [27].

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. [28].

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 malware [29].

SSDP Amplification : SSDP is a component of the Universal Plug and Play Protocol standard. Through the use of 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 in order 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.

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 (Halfond et al., 2006). 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, and they may use this information to harm their targets [31].

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 are capable of causing the WLAN to break due to the severe effect on Internet services [32].

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. In order 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 [33].

A classifier in machine learning is an algorithm that automatically assigns and categorizes data points to one of several categories or classes. There are two types of classifier models: supervised and unsupervised. Classifiers in the supervised model are trained to distinguish between labeled and unlabeled data. This training enables them to recognize patterns and operate autonomously without the use of labels. Unsupervised algorithms use pattern recognition to classify unlabeled datasets. 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.
(e) NaiveBayes.

## 4. 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 it 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).

*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. Figure 2 shows the proposed framework for processing and classification method that was followed in this phase.

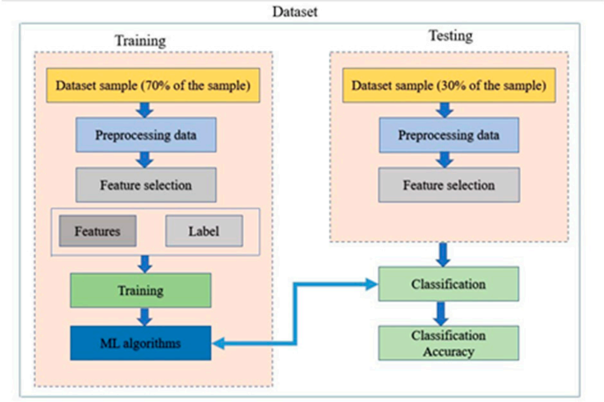


**Figure 2.** Proposed framework for processing and classification model.

**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 |

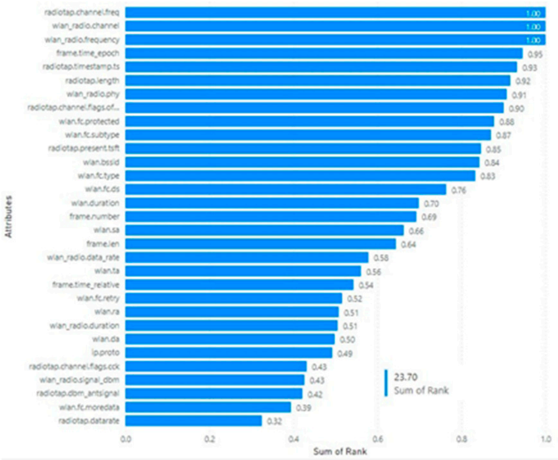
The procedure that was followed to implement this phase of evaluation and experiment was as follows:

1. Preprocessing step in several stages:

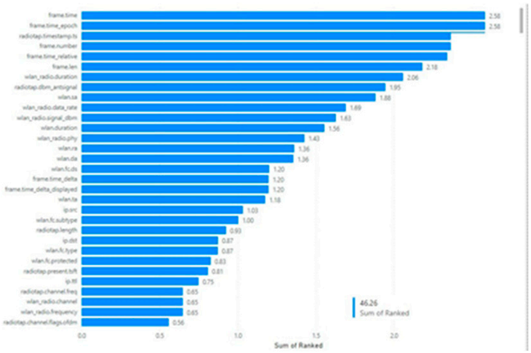
- At first, the data was cleaned by removing empty features and features with a constant value, the remaining features were 49.
- Randomize the data: Randomly shuffle the order of instances passed through it. The random number generator is reset with the seed value whenever a new set of instances is passed in, we chose the seed value to be 33.
- Remove Percentage: To fit the allocated memory for WEKA. The new sample consists of 7499 instances.
- Remove attributes that have over 50% of missing value, the remaining attributes are 44.

2. 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 IV-A and IV-A 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 Eval and Info Gain Attribute Eval. 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’s, so we have to remove them.



Gain Ratio Attribute Evaluation.



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 antsignal, wlan.fc.moredata, radiotap.datarate, ip.ttl, ip.src, frame.time delta, frame.time delta displayed, ip.dst, wlan.fc.pwrmgt, wlan.fc.frag, frame.time, wlan.seq, ip.version, llc, wlan.fc.order).

After preprocessing the data and applying feature selection algorithms, we applied different ML algorithms. Table 4 shows the performance of the learning algorithms using WEKA, while Table 5 shows the performance of the learning algorithms using AZURE.

**Table 4.** The performance of the learning algorithms on gain ratio and info gain feature selection.

| | | | | **Gain Ratio—Nomnal** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Splitting data 70% train and 30% test** | | | | 10-fold cross-validation | | | |
| **Algorithm** | **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 |
| NaiveBayes | 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 |
| | | | | Info Gain - Nominal | | | | |
| | Splitting data 70% train and 30% test | | | | 10-fold cross-validation | | | |
| Algorithm | Accuracy | Precision | Recall | F-Measure | Accuracy | Precision | Recall | F-Measure |
| treesJ48 | 99.67% | 1 | 1 | 1 | 99.69% | 1 | 1 | 1 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| NaiveBayes | 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 performance of the learning algorithms on gain ratio and info gain feature selection.

| **Gain Ratio - Nominal** | | | | |
|---|---|---|---|---|
| **Algorithm** | **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 |
| Info Gain - Nominal | | | | |
| 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 |

*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:

1. 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 7500 instances and 49 features.
- Remove attributes that have over 50% of missing value, the remaining attributes are 44.

**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 |

2. 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.pwrmgt, 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.pwrmgt, wlan.fc.type, 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 db wlan radio.phy, llc, ip.dst, ip.proto, ip.src, ip.ttl, ip.version, Label).

After preprocessing the data and applying feature selection algorithms, we applied different ML algorithms. Table 7 shows the performance of the learning algorithms using WEKA, while Table 8 shows the performance of the learning algorithms using AZURE.

**Table 7.** The performance of the learning algorithms on gain ratio and info gain feature selection (numeric).

| **Gain Ratio- Numerical** | | | | |
|---|---|---|---|---|
| **Splitting data 70% train and 30% test** | | | | |
| **Algorithm** | **Correlation coefficient** | **Mean absolute error** | Relative absolute error | Root relative squared error |
| DecisionStump | 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 |
| DecisionStump | 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 |
| DecisionStump | 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 |
| DecisionStump | 0.6079 | 1.0661 | 71.19% | 79.40% |
| Random Tree | 0.9958 | 0.0169 | 1.13% | 9.14% |

**Table 8.** The performance of the learning algorithms on gain ratio and info gain feature selection (numerical).

| **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 |

*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 (4000 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:

1. 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 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 40000 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 approach to feature selection that is notably sensitive to feature interactions. Figure 3 shows attributes distributed according to their sensitivity to the label.

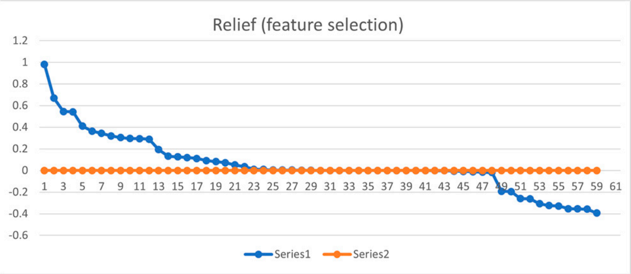


**Figure 3.** Relief feature selection.

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 40000 instances. Table 9 shows the performance of the learning algorithms using AZURE.

**Table 9.** The performance of the learning algorithms on relief feature selection.

| **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 |

After applying several experiments, we conclude that the AWID3 dataset contains many features that match the label class and just copying it, which leads to the emergence of these high accuracy results, and since there are many filters for features selection (where we already applied some of them previously ), in this experiment we will overlapping three different feature selection algorithms; (chi-squared feature selection, ANOVA feature selection, and Relief feature selection), then we will exclude the features with high ranked in the three algorithms.

Figure 4 will explain the idea to be implemented better.

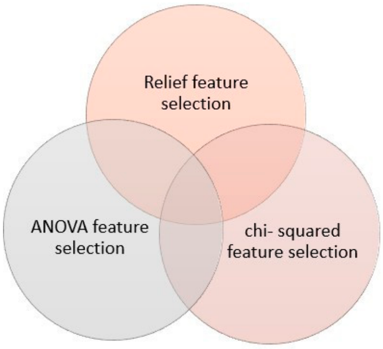


**Figure 4.** Overlapping between features selection algorithms.

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.pwrmgt, 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 40000 instances, the performance of the learning algorithms is presented in Table 10 for WEKA, and Table 11 for MATLAB.

**Table 10.** The performance of the learning algorithms when overlapping between features selection.

| Algorithm | Correlation coefficient | Mean absolute error | Relative absolute error | Root relative squared error |
|---|---|---|---|---|
| DecisionStump | 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 of the learning algorithms when overlapping between features selection algorithms-2.

| Algorithm | Accuracy | True Positive Rate (TPR) forclass 1 | False Negative | True Positive Rate (TPR) forclass | False Negative |
|---|---|---|---|---|---|

| | | | Rate (FNR) forclass 1 | | Rate (FNR) forclass 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% |
| Na¨ive Bayes | 95.3% | 98.4% | 1.6% | 92.2% | 7.8% |

*Analysis 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 5, For more detail, Figure 6, presented the result for this phase when we used training and validation split percentage.

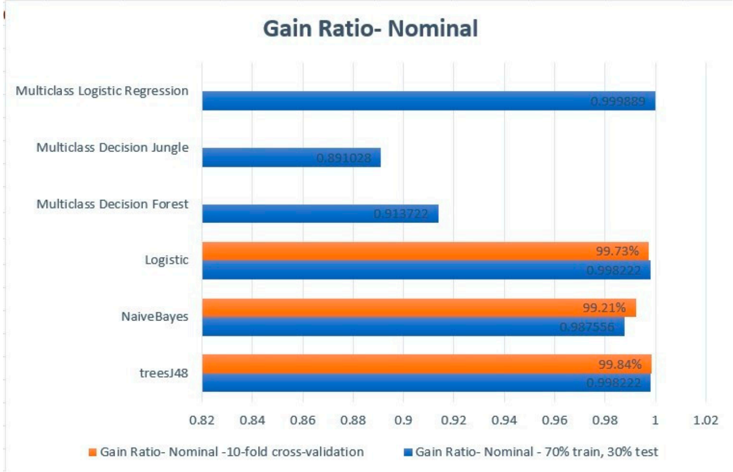


**Figure 5.** The performance of the proposed model built Gain Ratio Attributes Evaluator- Nominal class.

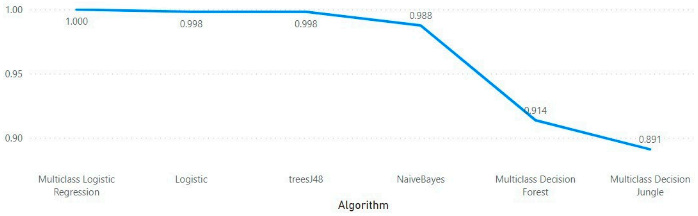


**Figure 6.** Accuracy results for ML algorithms using training and validation split percentage- Gain Ratio- Nominal.

When comparing this feature selection algorithm when the class is numeric with nominal class, the results show decreasing in performance as shown in Figures 7 and 8, where the highest accuracy that we get when the class is numeric was for logistic regression too with 99%, and the lowest was for DecisionStump with 82%.

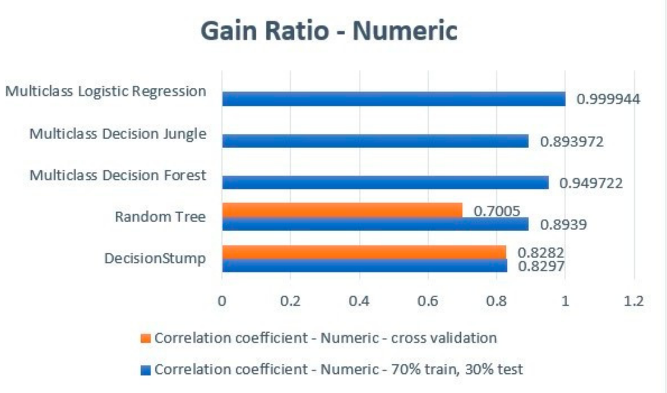


**Figure 7.** The performance of the proposed model built Gain Ratio Attributes Evaluator-Numeric class.

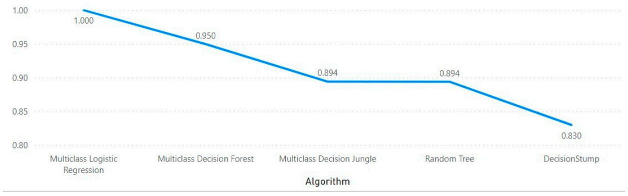
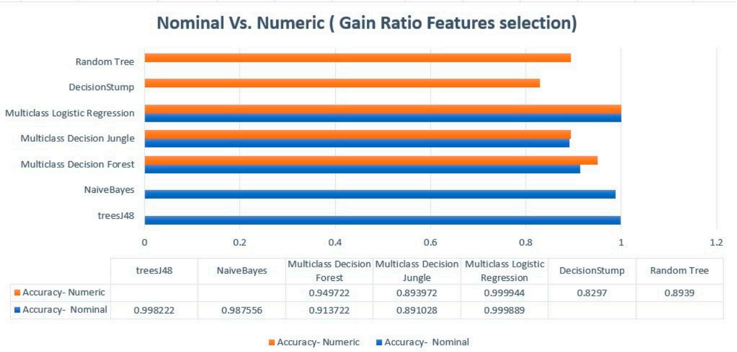


**Figure 8.** Accuracy results for ML algorithms using training and validation split percentage- Gain Ratio- Numeric.

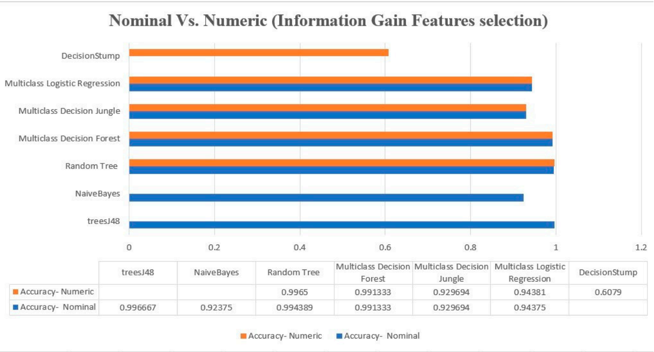
The results show that the performance of the suggested model was affected in some ML algorithms when the class changed to a numeric label, and some were not affected as shown in Figure 9.



| | treesJ48 | NaiveBayes | Multiclass Decision Forest | Multiclass Decision Jungle | Multiclass Logistic Regression | DecisionStump | Random Tree |
|---|---|---|---|---|---|---|---|
| Accuracy- Numeric | | | 0.949722 | 0.893972 | 0.999944 | 0.8297 | 0.8939 |
| Accuracy- Nominal | 0.998222 | 0.987556 | 0.913722 | 0.891028 | 0.999889 | | |

**Figure 9.** Comparison the performance when the class is Nominal and Numeric on Gain Ratio Attribute Evaluator.

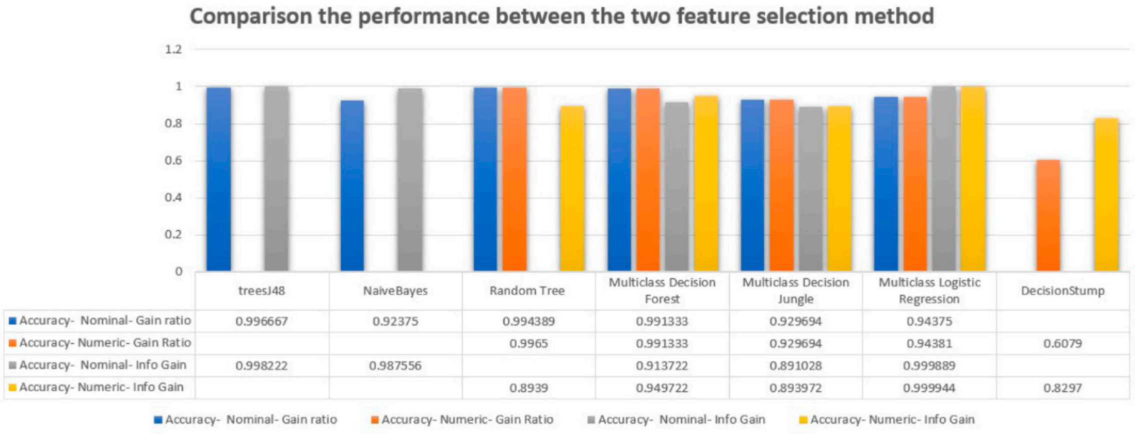
Referring to Figure 9, Logistic Regression still gives the highest accuracy with 99%, while Decision Stump and Random tree have the lowest accuracy with 82% and 89% consecutively. If we want to compare the accuracy of feature extraction methods that we have used, we have to look at Information gain attributes 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 10.



| | treesJ48 | NaiveBayes | Random Tree | Multiclass Decision Forest | Multiclass Decision Jungle | Multiclass Logistic Regression | DecisionStump |
|---|---|---|---|---|---|---|---|
| Accuracy- Numeric | | | 0.9965 | 0.991333 | 0.929694 | 0.94381 | 0.6079 |
| Accuracy- Nominal | 0.996667 | 0.92375 | 0.994389 | 0.991333 | 0.929694 | 0.94375 | |

**Figure 10.** Comparison the performance when the class is Nominal and Numeric on Information Gain Attribute Evaluator.

Figure 11 shows a comparison between the two feature selection method where the accuracy show stability and high results for ML algorithms that were used; except Decision-Stump which gave the lowest performance in both feature selection methods.



| | treesJ48 | NaiveBayes | Random Tree | Multiclass Decision Forest | Multiclass Decision Jungle | Multiclass Logistic Regression | DecisionStump |
|---|---|---|---|---|---|---|---|
| Accuracy- Nominal- Gain ratio | 0.996667 | 0.92375 | 0.994389 | 0.991333 | 0.929694 | 0.94375 | |
| Accuracy- Numeric- Gain Ratio | | | 0.9965 | 0.991333 | 0.929694 | 0.94381 | 0.6079 |
| Accuracy- Nominal- Info Gain | 0.998222 | 0.987556 | | 0.913722 | 0.891028 | 0.999889 | |
| Accuracy- Numeric- Info Gain | | | 0.8939 | 0.949722 | 0.893972 | 0.999944 | 0.8297 |

**Figure 11.** Comparison the performance between the two feature selection methods.

*Analysis 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 12 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.

*Comparison of our findings with other studies*

Table 12 summarized the studies mentioned in section II.

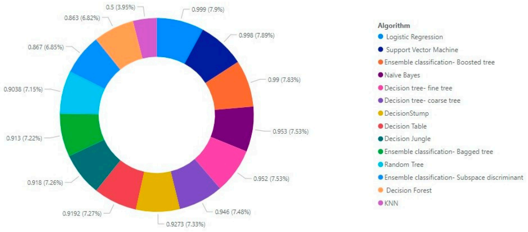


**Figure 12.** The performance of the proposed model for binary class.

**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%<br>F.S: 99% |
| [8] | Flood category contains Deauth, Disas, Assoc, and Kr00k attacks. Impersonation contains: RogueAP, Evil$_T$ *win, andKrack* | 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 | Multi class: 99.9% Bi-nary: 99% |

## 5. 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. Our 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 we dealt with throughout our research. We were able to overcome these difficulties and greatly raise the performance of our intrusion detection model by utilizing feature selection approaches. Three unique phases define our suggested Wireless Intrusion Detection System (WIDS): multi-nominal class, multi-numeric class, and binary class. We especially examined the performance of our model against the following 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, our experiments repeatedly showed great accuracy in identifying these assaults. Particularly, we were able to obtain an accuracy of 99% at its highest 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. Through our work, we hope to advance the state-of-the-art in wireless network security within the evolving 5G technological environment.

## References

1. Y. Zhou, Y. Gao, J. Chen, D. Li, Z. Liu, Y. Wei, and Z. Ma, “Blockchain for 5g advanced wireless networks,” in 2022 International Wireless Communications and Mobile Computing (IWCMC). IEEE, 2022, pp. 1306–1310.
2. 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, 2019.

3. K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, “Cyber threats to industrial iot: a survey on attacks and countermeasures,” IoT, vol. 2, no. 1, pp. 163–186, 2021.
4. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019.
5. 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, p. 102630, 2020.
6. J. Brownlee, “Introduction to dimensionality reduction for machine learning,” Machine Learning Mastery: Vermont, Australia, 2020.
7. E. Chatzoglou, G. Kambourakis, C. Smiliotopoulos, and C. Kolias, “Best of both worlds: Detecting application layer attacks through 802.11 and non-802.11 features,” Sensors, vol. 22, no. 15, p. 5633, 2022.
8. E. Chatzoglou, G. Kambourakis, C. Kolias, and C. Smiliotopoulos, “Pick quality over quantity: Expert feature selection and data preprocessing for 802.11 intrusion detection systems,” IEEE Access, vol. 10, pp. 64 761–64 784, 2022.
9. R. Saini, D. Halder, and A. M. Baswade, “Rids: Real-time intrusion detection system for wpa3 enabled enterprise networks,” arXiv preprint arXiv:2207.02489, 2022.
10. C. Zheng, M. Zang, X. Hong, R. Bensoussane, S. Vargaftik, Y. Ben-Itzhak, and N. Zilberman, “Automating in-network machine learning,” arXiv preprint arXiv:2205.08824, 2022.
11. S. C. Sethuraman, S. Dhamodaran, and V. Vijayakumar, “Intrusion detection system for detecting wireless attacks in ieee 802.11 networks,” IET networks, vol. 8, no. 4, pp. 219–232, 2019.
12. E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, “A supervised intrusion detection system for smart home iot devices. ieee internet things j. 6, 9042–9053 (2019),” 2019.
13. Q. M. Alzubi, M. Anbar, Z. N. 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, 2020.
14. 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, 2019.
15. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” CS224N project report, Stanford, vol. 1, no. 12, p. 2009, 2009.
16. 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, 2022.
17. E. Chatzoglou, G. Kambourakis, and C. Kolias, “Empirical evaluation of attacks against ieee 802.11 enterprise networks: The awid3 dataset,” IEEE Access, vol. 9, pp. 34 188–34 205, 2021.
18. C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, “Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset,” IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 184–208, 2015.
19. Arora, “Preventing wireless deauthentication attacks over 802.11 networks,” arXiv preprint arXiv:1901.07301, 2018.
20. M. A. C. Aung and K. P. Thant, “Ieee 802.11 attacks and defenses,” Ph.D. dissertation, MERAL Portal, 2019.
21. 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, 2011.
22. B. Lee, “Stateless re-association in wpa3 using paired token. electronics 2021, 10, 215,” 2021.
23. T. Zhou, Z. Cai, B. Xiao, Y. Chen, and M. Xu, “Detecting rogue ap with the crowd wisdom,” in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017, pp. 2327–2332.
24. C. P. Kohlios and T. Hayajneh, “A comprehensive attack flow model and security analysis for wi-fi and wpa3,” Electronics, vol. 7, no. 11, p. 284, 2018.
25. M. Cˇ erma´k, S. Svorencˇ´ık, and R. Lipovsky`, “Kr00k-cve-2019-15126–serious vulnerability deep inside your wi-fi encryption,” ESET Research White Paper, 2020.
26. S. K. Wanjau, G. M. Wambugu, and G. N. Kamau, “Ssh-brute force attack detection model based on deep learning,” 2021.
27. 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. IEEE, 2014, pp. 379–385.
28. 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, pp. 1–10, 2020.
29. O¨ . A. Aslan and R. Samet, “A comprehensive review on malware detection approaches,” IEEE Access, vol. 8, pp. 6249–6271, 2020.

doi:10.20944/preprints202307.1565.v1

30. X. Liu, L. Zheng, S. Cao, S. Helal, J. Zhou, H. Jia, and W. Zhang, “A multi-location defence scheme against ssdp reflection attacks in the internet of things,” in Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health. Springer, 2019, pp. 187–198.
31. W. G. Halfond, J. Viegas, A. Orso et al., “A classification of sql-injection attacks and countermeasures,” in Proceedings of the IEEE international symposium on secure software engineering, vol. 1. IEEE, 2006, pp. 13–15.
32. 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, 2020.
33. 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), 2016, pp. 1–6.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.