The object of research is IEEE 802.11 (Wi-Fi) networks, which are often the targets of a group of attacks called "evil twin". Research into this area is extremely important because Wi-Fi technology is a very common method of connecting to a network and is usually the first target of cybercriminals when they attack businesses. With the help of a systematic analysis of the literature focused on countering attacks of the "evil twin" type, this work identifies the main advantages of using artificial intelligence systems in the analysis of network data and identification of intrusions in Wi-Fi networks. To evaluate the effectiveness of intrusion detection and cybercrime analysis, a number of experiments as close as possible to real attacks on Wi-Fi networks were conducted.

As part of the research reported in this paper, a method is proposed for detecting cybercrimes in IEEE 802.11 (Wi-Fi) wireless networks using artificial intelligence, namely a model built on the basis of the k-nearest neighbors method. This method is based on the classification of previously collected data, namely the signal strength from the access point, and then continuous comparison of the newly collected data with the trained model.

A compact and energy-efficient prototype of a hardware and software system has been designed for the implementation of monitoring, analysis of ethernet network packets and data storage based on time series. In order to reduce the load on the computer network and taking into account the limited computing power of the system, a method of data aggregation was proposed, which ensures fast transfer of information.

The results, namely 100 % of test cases (more than 7 thousand), were classified correctly, which indicates that the chosen method of data analysis will significantly increase the security of information and communication systems at the state and private levels

Keywords: IEEE 802.11, Wi-Fi, evil twin, machine learning, classification, triangulation, cyber security

UDC 004.681
DOI: 10.15587/1729-4061.2023.282131

DEVISING A METHOD FOR DETECTING "EVIL TWIN" ATTACKS ON IEEE 802.11 NETWORKS (WI-FI) WITH KNN CLASSIFICATION MODEL

Roman Banakh
Corresponding author
Assistant
Department of Information Technology Security**
E-mail: banakh.ri@gmail.com
Andrian Piskozub
PhD*
Ivan Opirskyy
Doctor of Technical Sciences*
*Department of Information Security**
**Lviv Polytechnic National University
S. Bandery str., 12, Lviv, Ukraine, 79013

Received date 15.03.2023
Accepted date 08.06.2023
Published date 30.06.2023

How to Cite: *Banakh, R., Piskozub, A., Opirskyy, I. (2023). Devising a method for detecting "Evil Twin" attacks on IEEE 802.11 networks (Wi-FI) with knn classification model. Eastern-European Journal of Enterprise Technologies, 3 (9 (123)), 20–32. doi: https://doi.org/10.15587/1729-4061.2023.282131*

1. Introduction

Evil twin (ET) is a type of wireless attack where an attacker installs a fake Wireless Fidelity (Wi-Fi) access point with the same name as a legitimate access point to trick users into connecting to it. Once a user connects to a fake access point, an attacker can intercept and manipulate the user's network traffic, steal sensitive information, and conduct further attacks [1].

The new Wi-Fi Protected Access (WPA) version 3 security protocol, unlike WPA2, includes a feature called Simultaneous Authentication of Equals (SAE), also known as Dragonfly, which provides stronger protection against the ET attack. However, it is important to note that although WPA3 is more resistant to ET attacks, it does not completely protect against them. Since a very large number of client devices were manufactured before the invention of the WPA3 protocol, manufacturers of network devices laid a transitive mechanism that allows outdated user devices to work with new network equipment [2]. Subsequently, evidence was published in 2019 that some WPA3 routers were vulnerable to downgrade attacks [3]. Obviously, downgrading to WPA2 could potentially allow an attacker to bypass SAE protection and perform an ET attack, which indicates the relevance of this problem.

But as you know, every electronic device, even those manufactured at the same factory using identical technology, can have a unique digital footprint. If we talk about networks of the IEEE 802.11 standard, the signal strength can be the digital fingerprint of an access point at the channel level of the Open Systems Interconnection (OSI) model. In Wi-Fi networks, even the slightest deviation of the router's antenna can change the network coverage area. If we talk about an ET attack, it is usually carried out using equipment that is quite different from the one used to distribute Wi-Fi traffic. In addition, the location of the equipment plays a big role. If the Wi-Fi access point will be located in a certain room, and the attack will be carried out from outside its boundaries, then at the time of the attack, the signal level from the access point on a group of client devices will be completely different than in the absence of an attack.

The source of detection of attacks can be a device that is statically in the same place and periodically checks the signal level from a given access point. After that, it is possible to conclude about the presence of an attack based on the level of the

received signal. In combination with artificial intelligence algorithms, this approach will make it possible to distinguish even the smallest anomalies. In addition, the more devices that monitor the signal strength of a legitimate access point, the higher the probability of detecting an attack. It is obvious that the attacker has fewer and fewer chances to choose the correct configuration of the signal power on the device that acts as an ET.

As of 2021, the global economic value of Wi-Fi was estimated at $3.3 trillion and is expected to grow to $5 trillion by 2025 [4]. Since Wi-Fi technology is used in many areas of information activity, the issue of its security is also acute. So, we can consider the issue of Wi-Fi network security to be a very relevant topic.

2. Literature review and problem statement

Smart systems are no longer something that can only be used in expensive laboratories or hospitals. The rapid development of microchips has led to the development of a new direction in information technologies called the Internet of Things (IoT). They measure body temperature, pulse, oxygen saturation and other important indicators of the human body. In residential premises, such devices measure humidity, temperature, and air quality, and perform routine tasks instead of people. IoT devices are synchronized using cloud solutions and further enable process automation.

Any IoT device has at least one sensor, data from which is processed and transmitted for further display or analysis. Developers of such devices use a variety of information collection channels, often quite non-obvious, in order to measure the metrics they are interested in. Sometimes, Wi-Fi network cards can act as such sensors. For example, work [5] presents a solution that makes it possible to distinguish the activity of the human body using the analysis of the radio frequency spectrum from the Wi-Fi access point present in the same room. As you know, the human body can act as a natural obstacle for any radio signal. According to a certain human action during the experiment, the authors of [5] recorded changes in the signal strength from the access point. Some human actions are quite different by human nature, but the system identified them as the same. But subjectively, problems in distinguishing some actions occurred due to the fact that only one sensor appeared in the experiment. Therefore, in some cases, human actions did not affect the signal between the sensor and the access point.

Work [6] proposes an approach that allows unmasking the intruder. Metadata such as the SSID that their primary or secondary device is trying to connect to can be used to determine the name of an attacker's home access point. Obviously, the problem of determining the belonging of several devices can be quite a difficult task since different mobile devices have different amplification at the physical level. Objectively, if a directional antenna is used on the device with which the attack is carried out, it will be quite difficult to detect additional mobile devices of the attacker.

In work [7], an algorithm for identifying an ET attack on client devices is proposed, which is clearly important for clients, because it can prevent the theft of their personal data. In corporate networks, where there are dozens of such client devices, this approach can be used to create a powerful monitoring network. But subjectively, if one of the devices got into the attacker's network, it is impossible to guarantee that the information security team will be notified. Only the current user will be able to learn about such an event, and in this case, you will have to rely on his ability to respond to incidents of this nature. Objectively, in the case of public access points, this approach will not work since the clients are not permanent and the network owner has no control over their devices. A rather similar approach is proposed in [8], where the authors proposed a solution to detect an ET attack based on the differences in MAC addresses and external IP addresses of legitimate and illegitimate access points. But the issue of protection against the ET attack is still not resolved since this detection method is purely a solution for users of wireless computer networks and does not depend on the actions of the network administrator. This approach makes it possible to detect a duplicate access point and warn the client about a possible attack and offer to disconnect to protect against other types of attacks on user devices.

In order to identify the connection to an illegitimate access point, in [9] a mathematical model is presented to detect an ET attack by observing the time delay of termination of the TCP connection between the client and the server. But this method cannot be considered reliable enough since the issue of its use in extensive Wi-Fi networks operating in Mesh mode remains unresolved. The main problem is the increase in the number of nodes to the final destination, and the corresponding increase in response time can lead to false positives.

Another method of determining an ET attack is proposed in [10]. The attack detection algorithm is based on the intercepted deauthentication frame from the broadcast. If such a frame is detected, the intrusion detection system starts searching for access points with identical identifiers. But objectively, in the case of the "evil twin" attack, as well as in the case of interception of handshake packets to attack the WPA2 security protocol, attackers may not use any packet injection. The deauthentication package is only a tool to speed up the reconnection of client devices to the illegitimate access point. If the attackers are trying to remain undetected, they will not use this method but will just wait for a better moment when the signal of the illegitimate access point is stronger than the legitimate one for clients.

Wi-Fi networks are very popular due to the fact that they create comfortable conditions for Internet users. They are used both at home and in corporate settings. But the controlled data zone of this type of network is not clear, which often makes them a fairly easy target for cybercriminals. Some attacks have to be identified by methods that are not at all obvious, such as, for example, by collecting data on the signal strength from an access point and comparing it with a reference value. In [11], a method for identifying MAC address substitution in Wi-Fi networks is proposed by detecting a significant deviation in signal strength from the reference signal. However, objectively, the process of such a comparison is not reliable enough since the signal strength is affected by many physical factors, which can lead to a large number of false positives. An option to overcome the relevant difficulties may be the introduction of intelligent methods of data analysis, such as machine learning algorithms. This is the approach used in work [12]. The authors of work [12] propose an approach to detect an ET attack using a lightweight machine learning algorithm, namely Bayesian classification, based on a group of service data from access points, such as SSID, MAC address. If the probability threshold exceeds 75 %, the access point is marked as probably illegitimate. However, the investigation of networks takes place at the network level of the OSI model, and yet, subjectively, the analysis of data from the link level may give more accurate results.

Work [13] proposes an approach for determining the positioning of nodes in Wi-Fi networks based on the k-nearest neighbors (KNN) machine learning algorithm. Although the main goal of the considered work is not to improve the state of cyber security in wireless computer networks, nevertheless, the application of machine learning algorithms can provide advantages in solving some related problems.

Work [14] reports the results of research on detection of an ET attack using active statistical algorithms and algorithms for detecting anomalies. It is shown that the algorithms used by them (Trained Mean Matching and Hop Differentiating Technique) cope with the task with high efficiency. But a rather big problem is that the operation of such a system requires quite heavy calculations, which can sometimes be impractical for owners of small wireless networks. In addition, the authors state that the presence of network noise at a long distance of the computing server from the access point is a limitation. An option to overcome the relevant difficulties can be the use of sensors independent of the access point and cloud computing. This is the approach proposed in the current work.

So, a number of shortcomings can be singled out from the ones analyzed. The first is the operation of the intrusion detection system on client devices since this is not always appropriate due to the cost of calculations, which, among other things, affects the time of operation of client devices from autonomous power. In addition, the system itself should not have any communication with legitimate Wi-Fi infrastructure to avoid network interference. Also, the disadvantage is comparing the signal level using reference values since the signal strength of the access point can change in the presence or absence of certain physical obstacles. The consequence of such changes can be a large number of false positives of the intrusion detection system.

All this allows us to state that the detection of an ET attack is quite relevant, and it is appropriate to conduct a study aimed at devising a method of autonomous detection of intrusions using intelligent systems.

3. The aim and objectives of the study

The purpose of our study is to improve the methods of detecting an ET attack, in which attackers change legitimate systems to drag customers in order to take over their data. In the future, this will make it possible to significantly increase the resistance of wireless Wi-Fi networks to an ET attack.

To achieve the goal, the following tasks were set:

– to measure the signal strength from legitimate access points, simulate an ET attack, and measure signal strength from illegitimate access points;

– to analyze and prepare data for training a machine learning model;

– to train the machine learning model;

– to evaluate the effectiveness of the machine learning model.

4. The study materials and methods

The object of the research is IEEE 802.11 standard networks, which by their nature are vulnerable to the ET attack. The main hypothesis of this study assumes the possibility of identifying the appearance of an illegitimate Wi-Fi access point next to a legitimate one. A change in power from an access point with legitimate identifiers may indicate the appearance of an illegitimate access point in the controlled airwaves, i.e., a potential ET attack. During the development of the model of the intrusion detection system, it was assumed that the attack can be prevented thanks to machine learning algorithms with a trainer, which will allow more accurate identification of intrusions and avoid false positives.

The basic device used in the study is a Raspberry Pi 4 Model B single-board computer (UK) with optional Alfa Network AWUS036NHA (Taiwan) network adapters, which have the ability to work in monitoring mode. A Linux operating system based on the Debian distribution was installed on a single-board computer. A monitored device is any Wi-Fi router. Xiaomi Mi 4A (People's Republic of China) was used in this study.

The Wi-Fi over-the-air packet capture software was developed using the Python programming language. The functionality of intercepting packets from the ether was implemented using the Scapy library [15]. Hereafter, such devices will be called sensors.

A database based on time series InfluxDB [16] was installed on the main single-board computer to record data from sensors.

In the scheme with one sensor, software is also installed on it, which intercepts packets and the InfluxDB database. In a scheme with two or more sensors, InfluxDB is installed on only one of the sensors, and a Wi-Fi access point is deployed on it, which is used to deliver data to the database and other service communication.

Scapy is a powerful Python library for processing and analyzing network packets, namely creating and sending network packets, inspecting and analyzing network traffic, testing and analyzing network security. In this study, our program intercepted packets that contained a beacon (the Dot11Beacon layer, in Scapy). A Wi-Fi beacon is a type of control frame sent periodically by an access point (AP) to announce its presence and provide basic network information to clients and is therefore the best type of packet to monitor. The purpose of beacons is to allow customers to discover available Wi-Fi networks and provide basic information about the network, such as the network name (SSID), supported data rates, security modes, and the signal strength of the access point in decibels.

Beacons are used at the initial stage of a Wi-Fi connection and are necessary for the proper functioning of Wi-Fi networks. They provide a way for customers to discover available networks and make informed decisions about which network to connect to. As such, beacons are an important component of Wi-Fi networks, providing information about the network and allowing clients to discover and connect to the network.

During the experiment, Beacon packets were collected from the ether continuously (up to 10 packets per second). Signal strength is measured in decibels.

As already mentioned above, the sensors work on the basis of single-board computers, and therefore the computing power is quite limited. Also, communication between sensors may not be stable in some cases due to a number of reasons.

The main problem is that every second the service network may receive a request to write to the database. When several small packets are transmitted separately, each packet needs its own header and other control information. These overheads can quickly accumulate and take up a significant portion of the available bandwidth, leaving

less room for actual data transfer. Data aggregation can help reduce this overhead by combining multiple packets into one larger packet that requires only one header and control information. By combining multiple packets into one larger packet, the time required for data transmission can be reduced, which can help reduce latency and improve the overall performance of the service network. There is also a greater risk of packet loss or damage due to factors such as noise or interference. Consolidating multiple packages into one larger package reduces the chance of package loss or damage, as a larger package is more stable and less susceptible to such factors.

In general, data aggregation can help optimize network performance in low-speed networks by reducing overhead, latency, and the risk of packet loss or corruption. This usually improves the performance of network clients and ensures more efficient use of available network resources.

So, in order to avoid overloading the InfluxDB server, which was running on a single-board computer, as well as to avoid the load on the service network, it was decided to send already aggregated data once per minute. The general scheme of data collection from the air and communication between sensors is shown in Fig. 1.

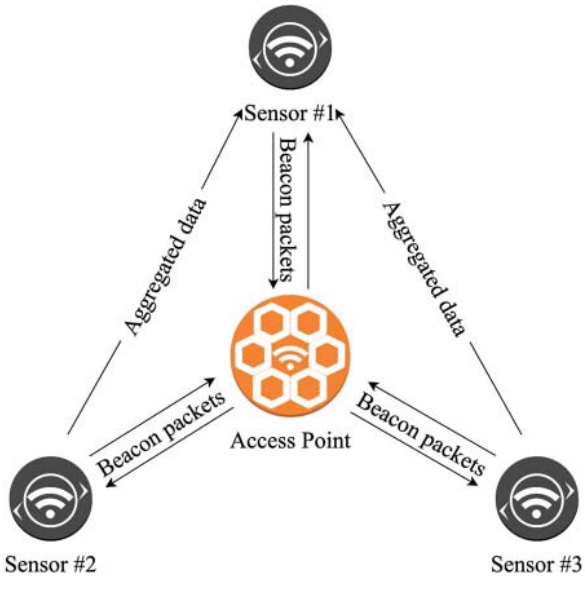

Fig. 1. General scheme of data collection from the ether and communication between sensors

A set of metrics in a one-minute interval is the number of intercepted beacons, the maximum signal strength, the minimum signal strength, the average signal strength, the signal strength that appeared most often on the air. The average capacity is calculated as the sum of all capacities divided by their number (1):

$$pwr_{avg} = \frac{sum(pwr)}{number\ of\ packets}. \quad (1)$$

The most frequently appeared signal strength is found using the mode's statistical function (2):

$$pwr_{most\ frequent} = mode(pwr). \quad (2)$$

After collecting the data, you can start training the machine learning model. For this, it is necessary to perform a number of actions, such as:

– selection of a machine learning algorithm based on a defined task;

– data preparation, which involves the collection, cleaning, and pre-processing of data. The data must be organized in a format suitable for the selected machine learning algorithm;

– data separation into training and test data. The training data is used to train the model and the test data is used to evaluate the performance of the model;

– based on the training data, the model is trained;

– test data is used to evaluate the machine learning model so that its model performance. This will help determine how well the model is able to generalize new data;

– if the model works poorly, you can try to improve it by adjusting the algorithm, changing the parameters of the model, or changing the data;

– if the model meets the requirements set before it, it can be deployed in a working environment to make predictions based on new data.

5. Results of research on detection of an "Evil Twin" attack on IEEE 802.11 standard networks using the k-nearest neighbors classification model

5. 1. Measurement of signal strength from legitimate and illegitimate access points. Simulating an "evil twin" attack

During the measurements, the access point and sensor(s) were located in a dedicated laboratory with no other electronic devices that could affect the signal level. However, the influence of the environment is not excluded since radio waves can be subject to various types of interference. Data was collected only in the Wi-Fi 2.4 GHz radio frequency range (2412–2472 MHz).

To reproduce the attack in close to real conditions, an access point with the same MAC address and network identifier (SSID) was created. Next, this access point was located in places outside the laboratory in which the experiment was conducted with a legitimate access point. The imitation of a legitimate access point was moved 15 times in the adjacent premises to the one in which the sensors and the legitimate access point were placed.

Two groups of experiments were conducted. In the first group, one sensor was used, which collected information about the signal strength from the access point (Fig. 2, *a*). In the second, there were three such sensors. The sensors were placed in a conditional triangle around the legitimate access point, that is, the so-called triangulation approach was applied (Fig. 2, *b*).

Fig. 3 shows the concept of the component scheme of the software-hardware system for detecting intrusions using machine learning algorithms.

The integrated system consists of three layers. The first is the information collection layer. This layer consists of three components – a data collector, a data aggregator, and a time series database. The task of this layer is exclusively data collection. The second is the data analysis layer. This layer consists of a data store prepared for training, a computing environment for training, and a continuous integration and delivery component of the machine learning model. The third is the layer of the application level. This layer consists of a web traffic load balancer, an application that operates on a trained model, and a database that stores data on detected attacks.

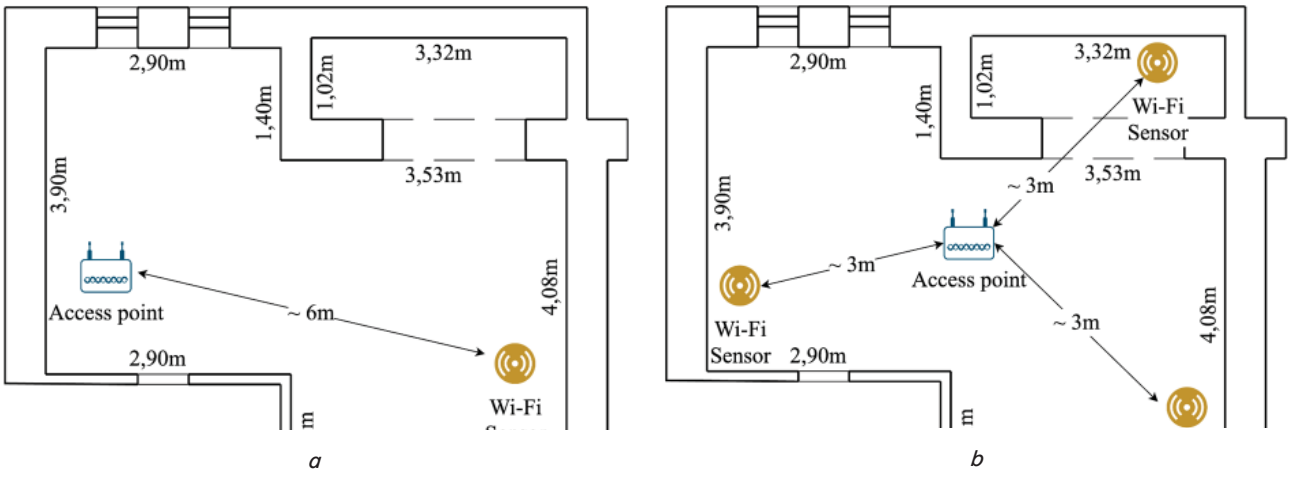

Fig. 2. The layout of equipment with: *a* – one sensor; *b* – three sensors

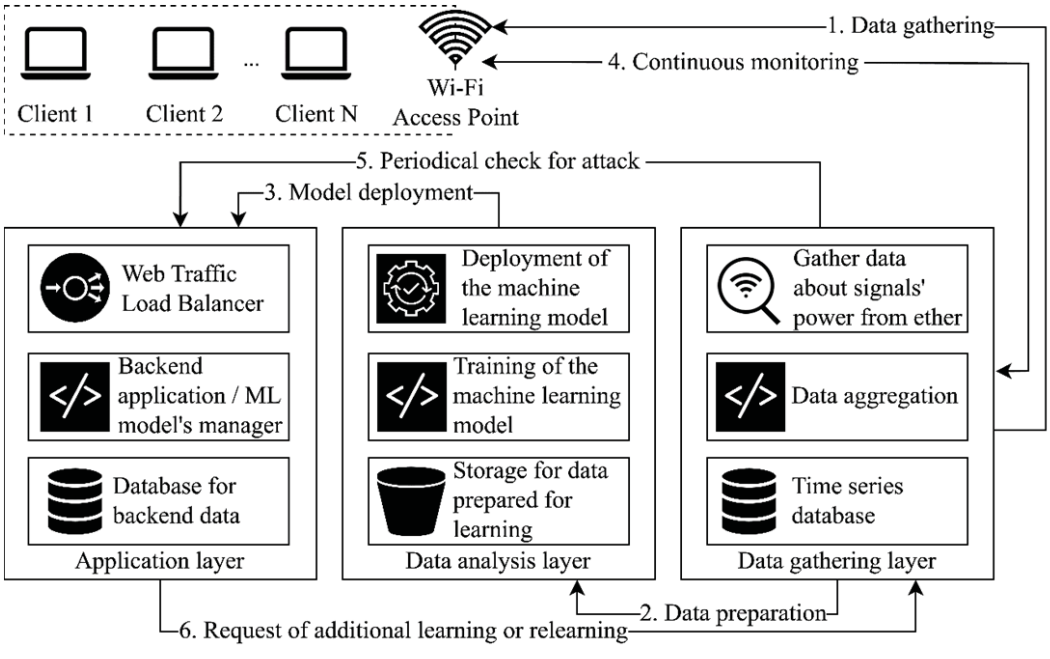

Fig. 3. Scheme of work and elements of the hardware and software system for detecting the "evil twin" attack

System operation: at the first stage, sensors collect data. After collection and aggregation, the data is recorded in a database based on time series. Once enough data is collected, it is sent to a data warehouse to train a machine learning model, after which the model is trained. Next, the trained model is delivered to the backend server.

After the process of collection, training. and delivery of artifacts, the components from the information collection layer are set to the watchdog mode of operation, the purpose of which is to monitor the signal level and compare it with the model. If necessary, you can call the mode of repeated collection and post-training or retraining of the model.

5. 2. Analyzing and preparing data for machine learning model training

A large amount of data is required to train a machine learning model. In order to understand whether the collected data is ready for processing, we visualize small time segments and compare metrics from legitimate and illegitimate access points (Fig. 4).

Fig. 4 shows the data collected by the same sensor, which displays the signal strength from legitimate and illegitimate access points for two hours. Fig. 4 shows that there is a rather large gap between the signal power of legitimate and illegitimate access points. This allows us to conclude that these data can be used for statistical analysis and training of a machine learning model.

The Python3 programming language with the numpy library [17] was used for data analysis, for working with multidimensional arrays and matrices. The libraries pandas [18] were also used for data manipulation and their further analysis, matplotlib [19] – for visualization of two-dimensional graphs, and seaborn [20] as an extension to matplotlib.

The total number of data sets was 14,579 rows for the experiment with one sensor and 20,610 rows for the experiment with three sensors. To understand how much the collected data is ready for model training, we can call the correlation visualization function on the heat map using the seaborn library. The seaborn library will show how much the data from the dataset correlates with each other (Fig. 5, 6).

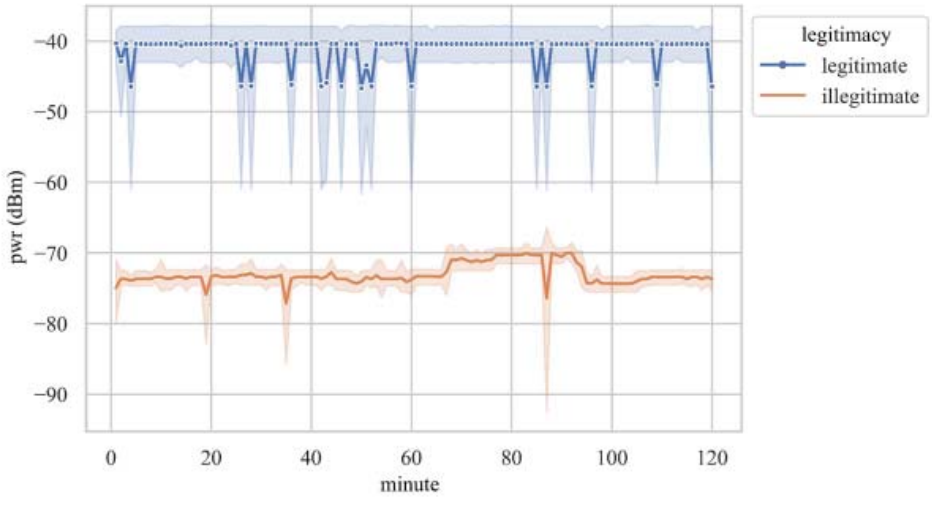

Fig. 4. Comparison of signal strength from legitimate and illegitimate access points

Fig. 5. Metric colleration heat map for a single sensor data set

In Fig. 5, min is the minimum signal level, max is the maximum signal level, *avg* is the average signal level, *mf* is the mode, or the one that occurs most often, *np* is the number of packets.

The first part of the metric *s*1–*sn* is responsible for the sensor number, and the second part, the name of the metric after the underscore, corresponds to the metrics mentioned in the explanation to Fig. 3.

As you know, the correlation coefficient measures the strength and direction of the linear relationship between two variables and ranges from −1 to +1, where a coefficient of +1 indicates a perfect positive linear relationship and a coefficient of −1 indicates a perfect negative linear relationship. A coefficient of 0 means no linear relationship.

The threshold for "low" correlation may depend on context and specific application. However, in general, a correlation coefficient less than 0.3 or greater than −0.3 is often considered a low correlation [21].

However, if there are many variables in the data set and the correlation between a particular variable and other variables is consistently weak, it may be appropriate to exclude that variable from the analysis or modeling process.

You can immediately see that the columns with the suffix *np*, which is an abbreviation for number of packets, are very poorly correlated with other parameters, in some cases the value is negative. This means that such parameters can be harmful to the prediction model. Therefore, they must be neglected in order to ensure the accuracy of the model. Fig. 7, 8 illustrate the heat map of the correlation of metrics but without the metric that is responsible for the number of intercepted beacons.

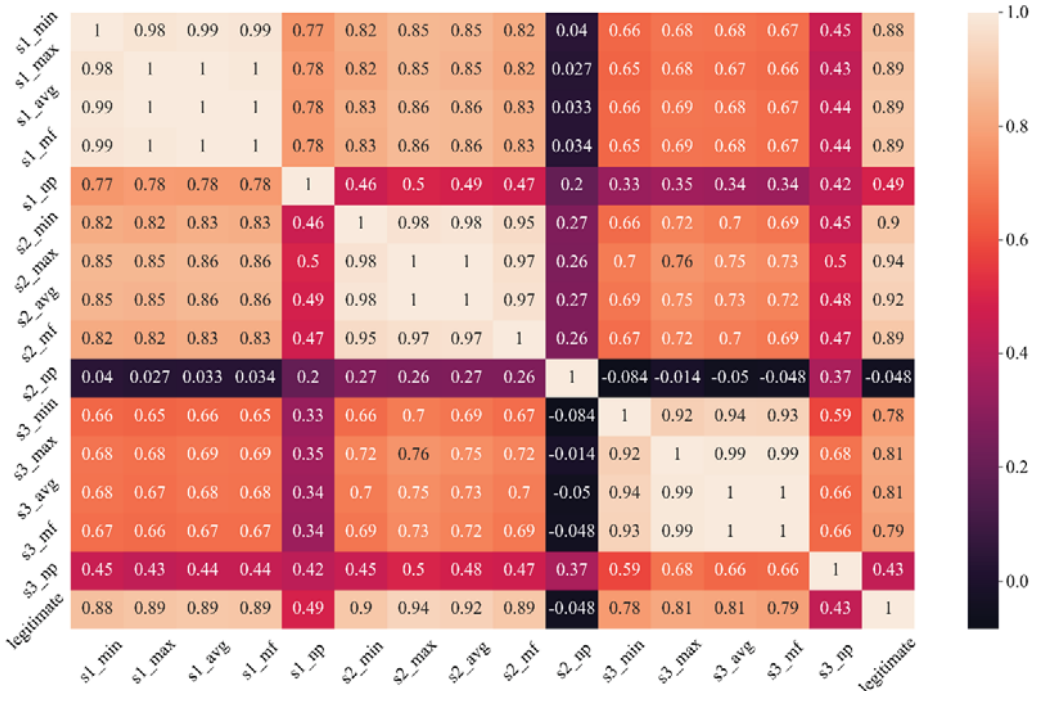

Fig. 6. Metric colleration heat map for a three-sensor data set

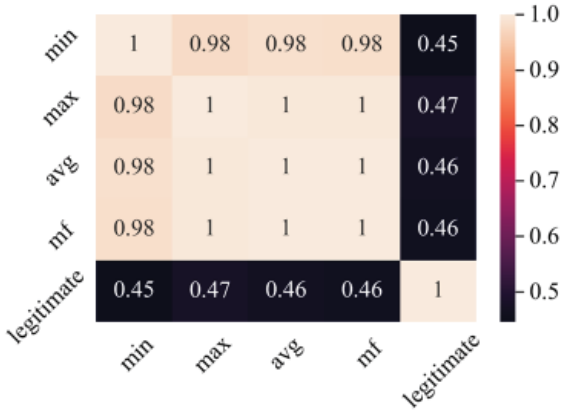

Fig. 7. Metric correlation heatmap for a single sensor data set with no metric of the number of Beacon packets intercepted

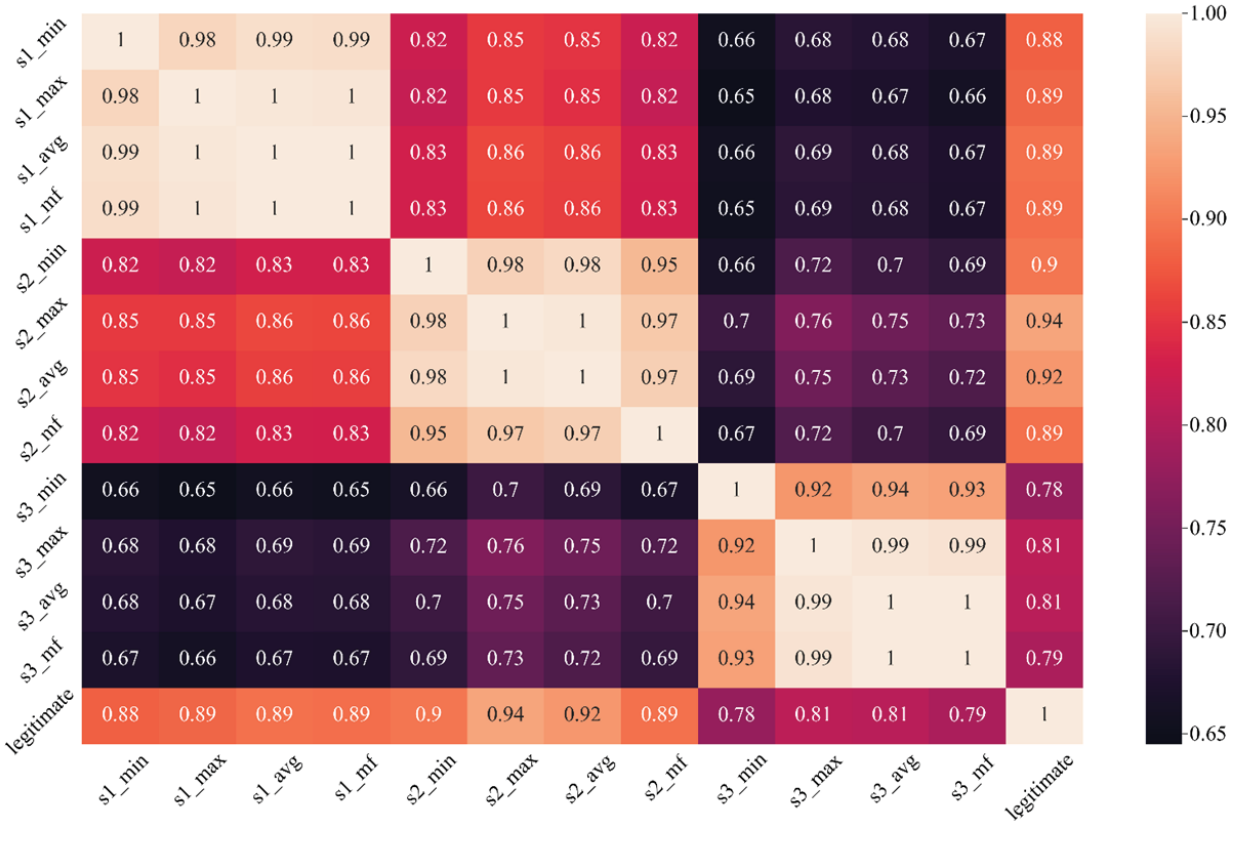

Fig. 8. Heat map of data correlation without metric of number of Beacon packets intercepted

As can be seen in Fig. 7, 8, the minimum correlation between the metrics is at least 0.45 and 0.65 for the data set with one and three sensors, respectively, which are quite good indicators.

Now, using the *plotpair* function of the *seaborn* library, the correlation of metrics for a group of experiments with one sensor (Fig. 9) and for each of the sensors for a group of experiments with three sensors (Fig. 10–12) is visualized. This comparison makes it possible to clearly see whether there is an intersection of signal strength data from legitimate and illegitimate access points.

Fig. 9 shows a fairly significant crossing of the minimum signal power from legitimate and illegitimate access points, which, of course, can become a problem when training a machine learning model. Nevertheless, such a case is quite likely under real conditions.

Fig. 10, 11, respectively, of sensor No. 1 and No. 2, demonstrate that the metrics of legitimate and illegitimate access points do not overlap, which is a pretty good sign for building a machine learning model. In the case of sensor No. 3 (Fig. 12), as in the case of the experiment with one sensor, slight intersections are visible when comparing the metrics of values from legitimate and illegitimate access points. But, despite such a distribution, this is a perfectly acceptable phenomenon because under real conditions, an attacker can place an ET in a place where the sensor will capture packets with a signal level similar to that of a legitimate access point.

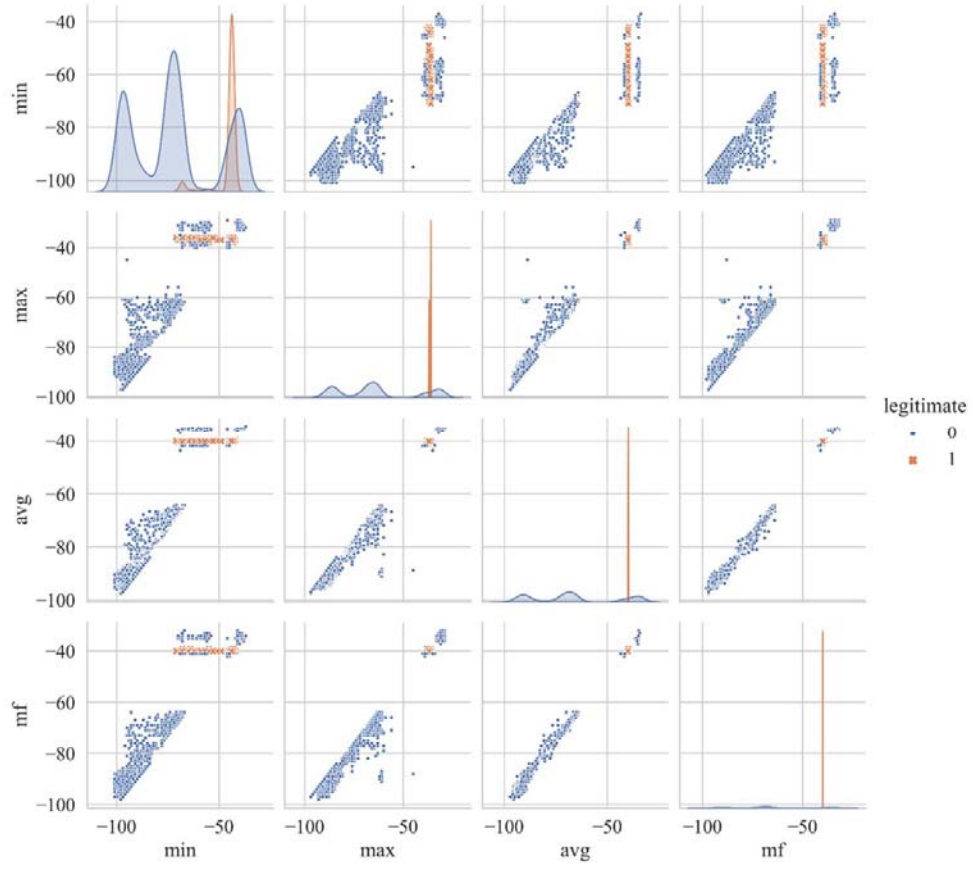

Fig. 9. Matching legitimate and illegitimate access point metrics for a group of experiments with a single sensor

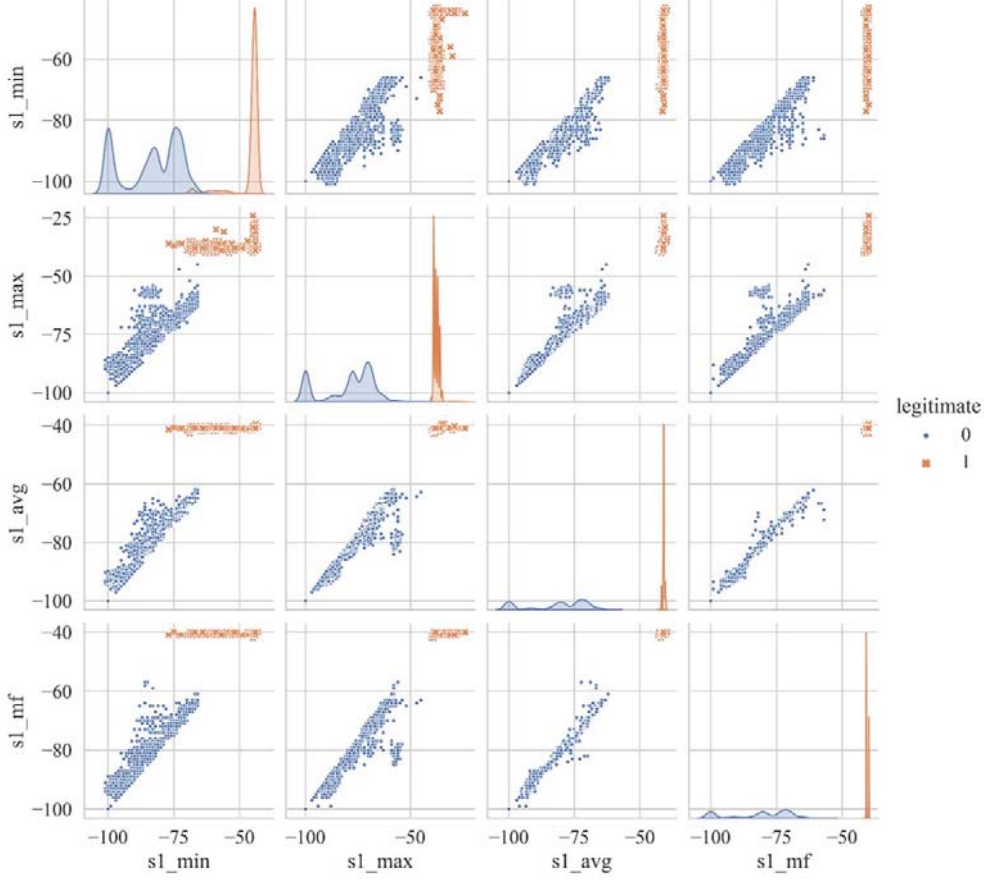

Fig. 10. Comparison of metrics of legitimate and illegitimate access points recorded by sensor No. 1

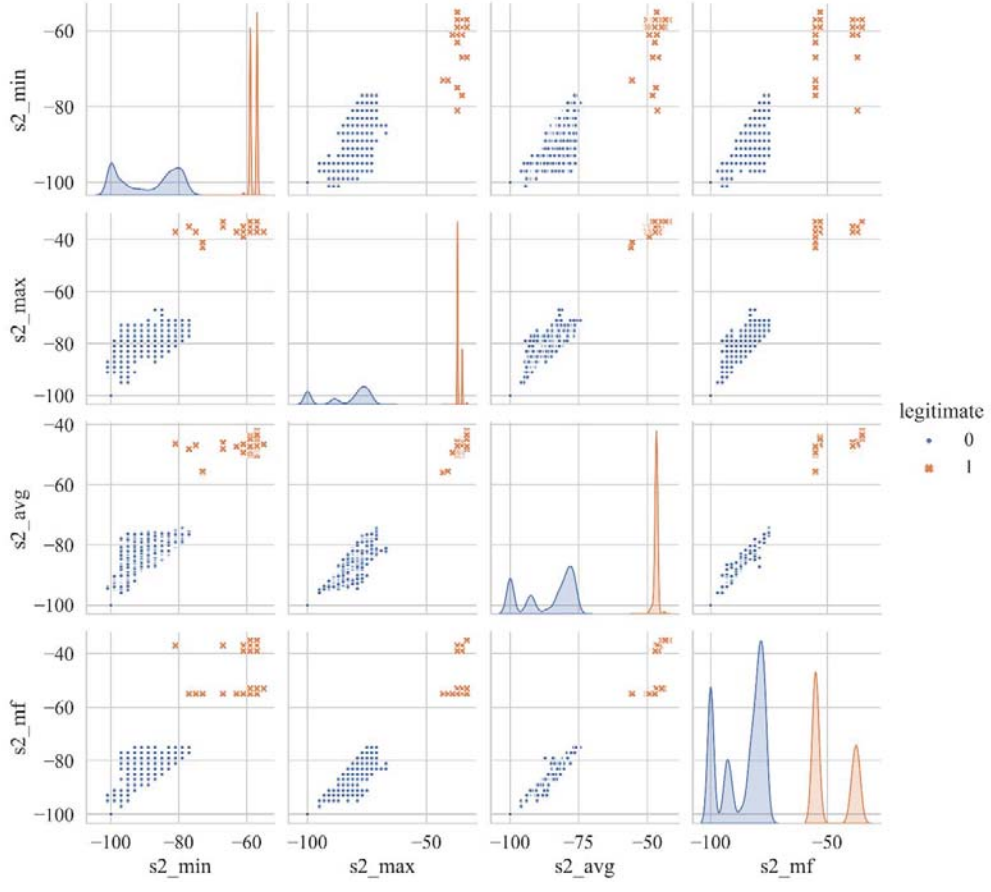

Fig. 11. Comparison of metrics of legitimate and illegitimate access points recorded by sensor No. 2

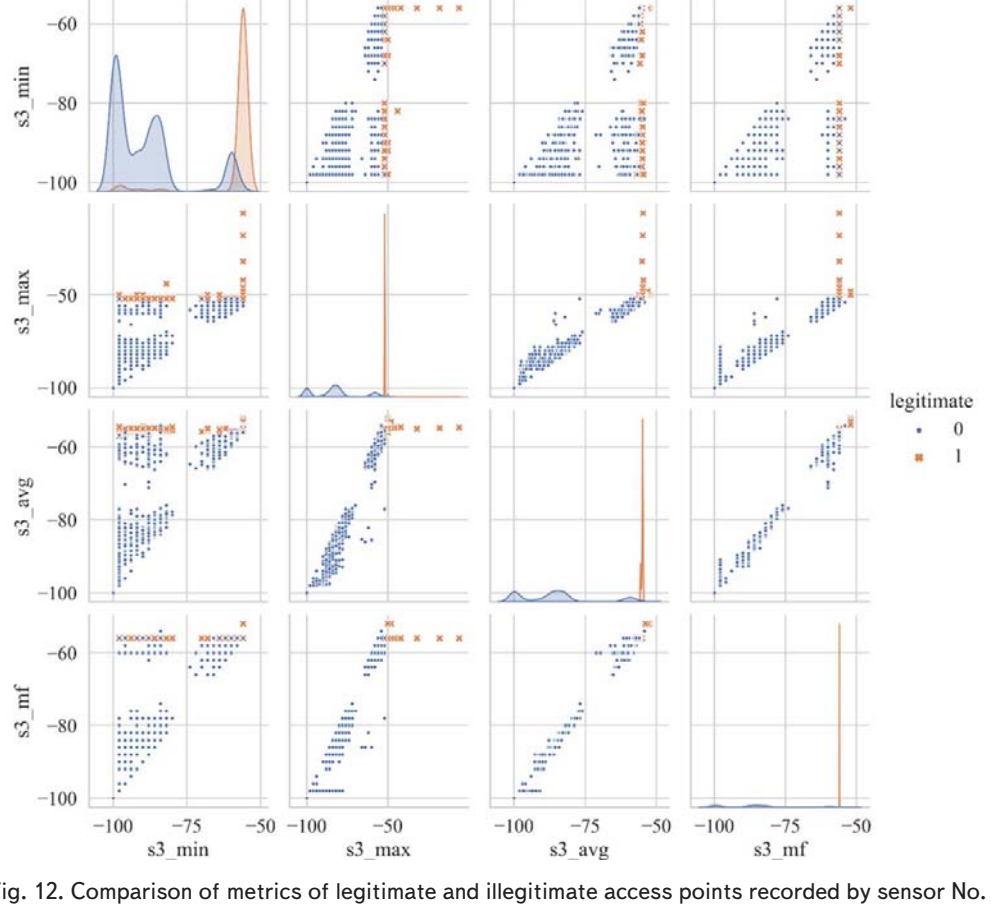

Fig. 12. Comparison of metrics of legitimate and illegitimate access points recorded by sensor No. 3

5. 3. Machine learning model training

To train the model on the input (*X_train*), the following metrics min, max, *avg, mf* were given for the single sensor experiment, and *s1_min, s1_max, s1_avg, s1_mf, s2_min, s2_max, s2_avg, s2_mf, s3_min, s3_max, s3_avg, s3_mf* for the group experiments with three sensors. The output metric (*y_train*) is the *legitimate* metric, which indicates whether the set of metrics is legitimate. Also, 20 % of the entire data set was used in testing (*X_test, y_test*).

The Scikit-learn library [22] was used to build a machine learning model. Scikit-learn offers a complete set of machine learning algorithms, including both trained and untrained methods. Some of the most commonly used algorithms in scikit-learn are linear and logistic regression, decision trees, random forests, *k*-nearest neighbors (KNN), support vector machines, and neural networks.

Since the data collected from the air may contain certain noises, that is, the signal level from the access point may not be stable at times, it was decided to use the KNN classification algorithm for data analysis in this work. KNN classifies the resulting data point based on its proximity to other data points in the training set. In this case, the model was trained relative to the maximum, minimum, and average signal strengths, as well as the most frequently occurring signal strength.

The distance metric is used to calculate the distance between two data points. Commonly used distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance [23].

In the training of our model, the Minkowski method was taken as the distance metric. The Minkowski metric is a generalized form of other distance metrics such as Euclidean distance and Manhattan distance. The distance of the Minkowski order p between two points is determined by formula (3):

$$D(x,y)=\left[\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{\frac{1}{p}}\right], \quad (3)$$

where x and y are two data points with n features, p is a parameter that determines the order of the distance metric, and $D(x, y)$ is the Minkowski distance between x and y.

When p=1, the Minkowski metric is reduced to the Manhattan distance, and when p=2, to the Euclidean distance. In general, a larger p value results in a stronger emphasis on larger differences between trait values and a weaker emphasis on small differences. For model training, the order of the distance metric was assigned to 2, and therefore the distance metric is equivalent to the Euclidean distance (4):

$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}. \quad (4)$$

Among the K nearest neighbors, the number of data points belonging to each class is counted. The class with the largest number of nearest neighbors is assigned to the invisible data point. The value of K determines the number of nearest neighbors and can be chosen based on experiment. In this study, the value of K was chosen experimentally to be 3.

Majority voting: when K nearest neighbors are found, the data point is assigned the class with the most nearest neighbors. If K is an odd number, the class with the largest number of nearest neighbors is the majority. If K is an even number, the majority class is determined by considering the values of the nearest neighbors in relation to the data point [24].

In mathematical terms, most votes can be expressed as follows:

$$y=majority(y_1,y_2,\ldots,y_K), \quad (5)$$

where $y_1, y_2,..., y_k$ are the class labels of the K nearest neighbors, and most ($y_1, y_2,..., y_k$) returns the class label that occurs most often among the K nearest neighbors.

The sequence of the classification process is shown in Fig. 13.

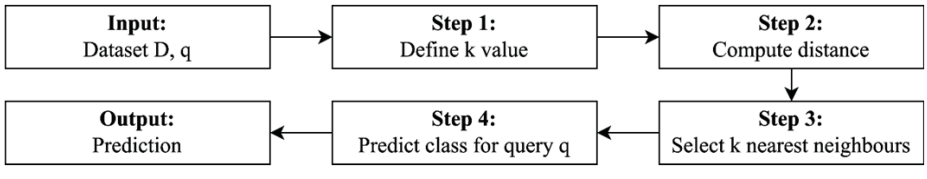

Fig. 13. The sequence of the classification process using the *k*-nearest neighbors method

A brief description of the classification process shown in Fig. 11 is presented as follows:

– the input data block represents the input data for the KNN algorithm, which consists of a data set D consisting of n data points $(x_1, y_1)...(x_n, y_n)$ and a query point q;

– step 1 involves setting the value of k, which determines how many nearest neighbors to take into account during prediction;

– step 2 involves calculating the distance between the query point q and each data point in the data set D;

– step 3 involves the selection of k data points from the data set D that have the shortest distance to the query point q;

– step 4 involves counting the number of data points in the selected set that belong to each class and selecting the class with the largest number as the predicted class for the query point q;

– the output block represents the output of the KNN algorithm, which is the predicted class for the query point q.

5. 4. Evaluating the effectiveness of the machine learning model

After training the model, tests were performed on twenty percent of the total data set. To obtain a classification report, the *classification_report* function from the *sklearn* library of the *metrics* class was used. As a result, reports were obtained for the scheme with one sensor (Table 1) and three sensors (Table 2).

In the classification reports (Tables 1, 2) you can see a set of metrics, namely:

– *precision* is the number of true positive results divided by the total number of positive predictions;

– *recall* is the number of truly positive samples divided by the total number of actually positive samples;

– *f1-score* is a harmonic average of precision and recall that provides a balance between precision and recall;

– *support* is the number of actual occurrences of the class in the specified data set. In other words, it is the number of samples in each class [25].

Table 1

Classification report for a single sensor circuit

Metric	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	2470
1	1.00	1.00	1.00	446
Accuracy	–	–	1.00	2916
Macro avg	1.00	1.00	1.00	2916
Weighted avg	1.00	1.00	1.00	2916

Table 2

Classification report for a three-sensor circuit

Metric	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	2792
1	1.00	1.00	1.00	1330
Accuracy	–	–	1.00	4122
Macro avg	1.00	1.00	1.00	4122
Weighted avg	1.00	1.00	1.00	4122

In addition to *precision, recall, f1-score*, and *support*, the *classification_report* function of the *sklearn* library also reports three other important metrics: *accuracy, macro avg*, and *weighted avg*. *Accuracy* is the proportion of correctly predicted labels among all samples in the data set. This is a metric that measures the overall performance of the model; *macro avg* is the average value of indicators (*precision, recall* and *f1-score*) for all classes. This metric is useful when you want to evaluate the overall performance of a model in a multiclass classification problem; *weighted avg* – the value of indicators (*precision, recall* and *f1-score*) for all classes, weighted by the number of samples in each class. This metric is useful when you have an unbalanced data set and want to evaluate the overall performance of the model taking into account class imbalance.

As can be seen from Table 1, from 20 % of the data set, namely from 2916 cases, the KNN model correctly classified all test cases. We can see the same result in the case of three sensors, and in this case, as you can see the number of test cases is already 4122. Unlike the data set with one sensor, the data set with three sensors has several times more test cases collected from a legitimate access point. Fig. 14, *a* shows the visualization of the inconsistency matrix of the KNN machine learning model for the case with one sensor, and Fig. 14, *b* shows the inconsistency matrix of the machine learning model for the case with three sensors.

The number of correctly classified illegitimate access points is displayed in the upper left corner (Fig. 14, *a, b*); the number of correctly classified legitimate access points is displayed in the lower right corner. Accordingly, the number of incorrectly classified illegitimate access points is displayed in the upper right corner; the lower left corner shows the number of misclassified legitimate access points.

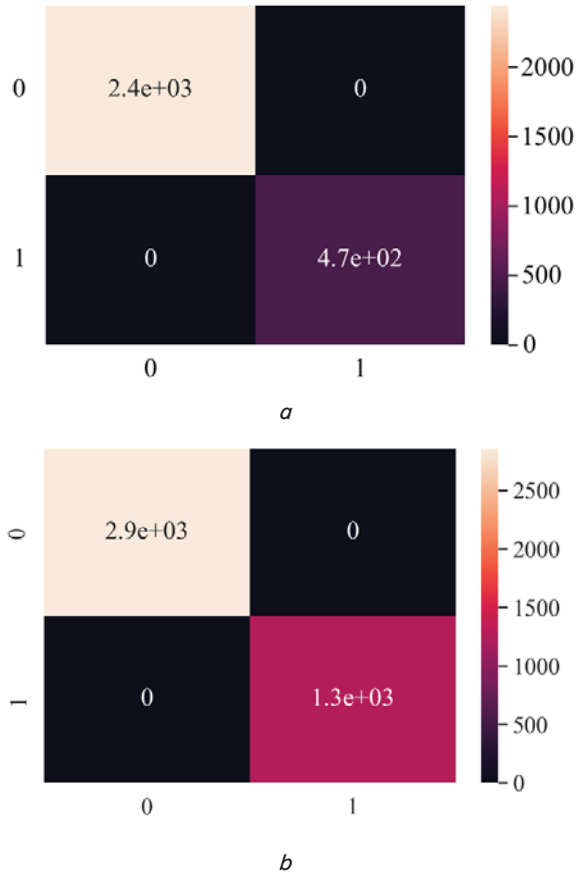

Fig. 14. Visualization of the inconsistency matrix for the KNN machine learning model for a scheme with: *a* – one sensor; *b* – three sensors

6. Discussion of results and directions for further research

This paper proposes a method for detecting intrusions by collecting and analyzing signal strength data from Wi-Fi access points collected from the air. As part of the implementation of this method, a hardware and software system was developed, which made it possible to monitor service packets on the air of IEEE 802.11 networks with high efficiency and low energy consumption. Fig. 3 presents the scheme and algorithm of the system. The effectiveness of the system can be explained by the correctly selected method of data aggregation, thanks to which it was possible to avoid the load on the service network and database (1), (2).

At the beginning of the study, it was assumed that the number of intercepted packets transmitted by legitimate and illegitimate access points could become a clear marker of the attack. But when analyzing the data, it was found that the correlation of this metric compared to others is very low (Fig. 5, 6). This can be explained by the fact that the number of packets from the illegitimate access point was about the same as from the legitimate one. Taking into account all the risks, it was decided to ignore this metric in order to ensure the high efficiency of the model (Fig. 7, 8). After cleaning the data set, a comparison of metrics was visualized, which showed a clear difference

between metrics obtained from a legitimate access point and from illegitimate ones, which became the basis for accepting the data for model training (Fig. 9–12).

Metrics such as min, max, avg, mf collected from sensors were used to train the machine learning model. The output metric was the legitimate metric, which indicates whether the metric set belongs to a legitimate access point. The Scikit-learn library was used to create the model, which made it possible to train the model without too much effort. The KNN classification algorithm was used for data analysis, the advantage of which is robustness to noise in the training data. This subsequently made it possible to achieve a high level of accuracy in the predictions of ET attacks. To calculate the distance, the Minkowski metric with the parameter p=2 (3), (4) was used. The value of K, the number of nearest neighbors, was set to 3, which was determined empirically. Class determination for unseen data points was done using majority voting among K nearest neighbors (5).

After training the model, tests were performed on 20 % of the total data set, after which a classification report was generated for the experiment of both experiments (Tables 1, 2). From the tables, all cases, more than 7,000 in number, were classified correctly. This can be explained by the fact that the data were properly prepared for the learning process, as well as by the correct choice of the machine learning algorithm (Fig. 14, *a*, *b*).

In contrast to [11], where a method of detecting intrusions based on the deviation from the reference value of the signal power is proposed, in this work the detection results are more accurate, and the number of false positives will be smaller. This becomes possible thanks to the implementation of a machine learning algorithm that can detect hidden dependences in data sets, as well as the proposed approach of retraining or retraining the model (Fig. 3). As part of the implementation of this method, a hardware and software system was developed, which allows monitoring of service packets on the air of IEEE 802.11 networks with high efficiency and low energy consumption. In contrast to the approach reported in work [14], it was possible to avoid the load on the client network within the scope of the study of this work. This became possible thanks to the use of an independent network of sensors that do not interact in any way with the network where the client traffic circulates.

During the experiment, to simulate an ET attack, data was collected from illegitimate access points that were located outside the laboratory. A limitation is that the data was collected from locations where the attacker might have been. This work did not take into account cases in which the attacker could use directional antennas. Simulation of such cases is quite difficult to cover if it is done manually since the attacker can have different types of equipment and be anywhere. Such cases and even more could be covered by a certain generative algorithm that would cover all combinations of signal strengths other than the legitimate access point.

The disadvantage is that even with a hundred percent coverage of all cases with the placement of a potential ET, although small, there is still a probability of not detecting an attack. By selecting the configuration and location, the digital trace of malicious equipment can resemble a legitimate device for the attack recognition system presented in this paper. Another drawback is the redundancy of the part of the system responsible for collecting data from the ether (Fig. 3). The operation of Raspberry Pi single-board computers requires the presence of an operating system, such as Linux, which is redundant since most of the functions of the operating system were not used within the scope of the task. And although this system copes well with the tasks, its price can be quite high, when used in enterprises with a large amount of wireless network equipment. And as it was already mentioned in this work, the more sensors, the better the quality of network monitoring.

As a result, it can be stated that with the help of continuous monitoring of the ether and the application of machine learning algorithms on the collected data, it is possible to significantly improve the mechanisms for determining attacks on Wi-Fi computer networks using signal strength analysis. The efficiency results are presented in Tables 1, 2, which prove that with the help of the KNN machine learning algorithm, it is possible to detect an ET attack with fairly high accuracy. This algorithm refers to algorithms with a trainer and requires the initial collection of information and subsequent training of the model based on the collected information. In addition, the ET attack can be used by an attacker not only to change identification data regarding Wi-Fi access points but also to change client devices, where it is quite difficult to apply algorithms with a trainer. Algorithms of machine learning without a trainer, although they are more resource-intensive, can quickly cope with tasks of this type since they do not require human participation at some stages of the process. So, this is a direction for further research.

7. Conclusions

1. A compact and energy-efficient prototype of a hardware and software system has been designed for the implementation of monitoring and analysis of service network packets on the air and saving data based on time series. In order to reduce the load on the computer network and, taking into account the limited computing power of the system, a method of data aggregation was proposed, which ensures fast data transfer. This system made it possible to collect and organize a large amount of data from legitimate and illegitimate access points, which was later used to train a machine learning model.

2. The collected data was analyzed and prepared for model training. During the analysis, it was found that the number of packets metric could potentially have a negative impact on the training of the machine learning model.

3. A method of detecting ET-type attacks in IEEE 802.11 wireless networks (Wi-Fi) using a machine learning algorithm with a KNN trainer is proposed. More than 35,000 Wi-Fi broadcast recordings from legitimate and illegitimate (which simulated an attack) access points were collected to train the model. The best results when training the model were obtained with the distance order (p) set to 2, which was chosen empirically.

4. The trained model made it possible to classify data sets with legitimate access point signal strengths from

illegitimate ones with high efficiency. All 20 % of the test cases (more than 7,000 cases) of the collected data, which were allocated for testing the machine learning model, were classified correctly. The results indicate that the KNN method effectively performs the given task with extremely high accuracy.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study and the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The data will be provided upon reasonable request.

References

1. Nagpal, J., Patil, R., Jain, V., Pokhriyal, R., Rajawat, R. (2018). Evil Twin Attack and Its Detection. International Journal of Emerging Technologies and Innovative Research, 5 (12), 169–171. doi: https://www.jetir.org/view?paper=JETIR1812326
2. Bednarczyk, M., Piotrowski, Z. (2019). Will WPA3 really provide Wi-Fi security at a higher level? XII Conference on Reconnaissance and Electronic Warfare Systems. doi: https://doi.org/10.1117/12.2525020
3. Vanhoef, M., Ronen, E. (2020). Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. 2020 IEEE Symposium on Security and Privacy (SP). doi: https://doi.org/10.1109/sp40000.2020.00031
4. Value of Wi-Fi. Wi-Fi Alliance. Available at: https://www.wi-fi.org/discover-wi-fi/value-of-wi-fi
5. Forbes, G., Massie, S., Craw, S. (2020). WiFi-based Human Activity Recognition using Raspberry Pi. 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). doi: https://doi.org/10.1109/ictai50040.2020.00115
6. Banakh, R., Piskozub, A. (2018). Attackers' Wi-Fi Devices Metadata Interception for their Location Identification. 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS). doi: https://doi.org/10.1109/idaacs-sws.2018.8525538
7. Lu, Q., Qu, H., Zhuang, Y., Lin, X.-J., Ouyang, Y. (2018). Client-Side Evil Twin Attacks Detection Using Statistical Characteristics of 802.11 Data Frames. IEICE Transactions on Information and Systems, E101.D (10), 2465–2473. doi: https://doi.org/10.1587/transinf.2018edp7030
8. Modi, V., Parekh, C. (2017). Detection of Rogue Access Point to Prevent Evil Twin Attack in Wireless Network. International Journal of Engineering Research And, V6 (04). doi: https://doi.org/10.17577/ijertv6is040102
9. Kuo, E.-C., Chang, M.-S., Kao, D.-Y. (2018). User-side evil twin attack detection using time-delay statistics of TCP connection termination. 2018 20th International Conference on Advanced Communication Technology (ICACT). doi: https://doi.org/10.23919/icact.2018.8323699
10. Agarwal, M., Biswas, S., Nandi, S. (2018). An Efficient Scheme to Detect Evil Twin Rogue Access Point Attack in 802.11 Wi-Fi Networks. International Journal of Wireless Information Networks, 25 (2), 130–145. doi: https://doi.org/10.1007/s10776-018-0396-1
11. Banakh, R., Piskozub, A., Opirskyy, I. (2018). Detection of MAC Spoofing Attacks in IEEE 802.11 Networks Using Signal Strength from Attackers' Devices. Advances in Computer Science for Engineering and Education, 468–477. doi: https://doi.org/10.1007/978-3-319-91008-6_47
12. Harsha, S. et al. (2019). Improving Wi-Fi security against evil twin attack using light weight machine learning application. COMPUSOFT, 8 (3). Available at: https://www.researchgate.net/publication/332344245_Improving_Wi-Fi_security_against_evil_twin_attack_using_light_weight_machine_learning_application
13. Dong, Y., Zampella, F., Alsehly, F. (2023). Beyond KNN: Deep Neighborhood Learning for WiFi-based Indoor Positioning Systems. 2023 IEEE Wireless Communications and Networking Conference (WCNC). doi: https://doi.org/10.1109/wcnc55385.2023.10118752
14. Yang, C., Song, Y., Gu, G. (2012). Active User-Side Evil Twin Access Point Detection Using Statistical Techniques. IEEE Transactions on Information Forensics and Security, 7 (5), 1638–1651. doi: https://doi.org/10.1109/tifs.2012.2207383
15. Scapy. Available at: https://scapy.net/
16. InfluxDB. Available at: https://www.influxdata.com/
17. NumPy. Available at: https://numpy.org/
18. Pandas. Available at: https://pandas.pydata.org/
19. Matplotlib. Available at: https://matplotlib.org/
20. Seaborn. Available at: https://seaborn.pydata.org/
21. Salkind, N. J., Frey, B. B. (2019). Statistics for people who (think they) hate statistics. SAGE Publications, 76–102.
22. Scikit-Learn. Available at: https://scikit-learn.org/stable/
23. Mladenova, T., Valova, I. (2021). Analysis of the KNN Classifier Distance Metrics for Bulgarian Fake News Detection. 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). doi: https://doi.org/10.1109/hora52670.2021.9461333
24. Taunk, K., De, S., Verma, S., Swetapadma, A. (2019). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems (ICCS). doi: https://doi.org/10.1109/iccs45141.2019.9065747
25. sklearn.metrics.classification_report. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html