data

MDPI

Article

Data Mining Approach for Evil Twin Attack Identification in Wi-Fi Networks

Roman Banakh [1], Elena Nyemkova [1,*], Connie Justice [2], Andrian Piskozub [1] and Yuriy Lakh [1]

[1] Department of Information Technology Security, Lviv Polytechnic National University, 79013 Lviv, Ukraine; roman.i.banakh@lpnu.ua (R.B.); andriian.z.piskozub@lpnu.ua (A.P.); yurii.v.lakh@lpnu.ua (Y.L.)
[2] Purdue School of Engineering and Technology, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202, USA; cjustice@iupui.edu
* Correspondence: olena.a.niemkova@lpnu.ua; Tel.: +38-063-934-6768

Abstract: Recent cyber security solutions for wireless networks during internet open access have become critically important for personal data security. The newest WPA3 network security protocol has been used to maximize this protection; however, attackers can use an Evil Twin attack to replace a legitimate access point. The article is devoted to solving the problem of intrusion detection at the OSI model's physical layers. To solve this, a hardware–software complex has been developed to collect information about the signal strength from Wi-Fi access points using wireless sensor networks. The collected data were supplemented with a generative algorithm considering all possible combinations of signal strength. The k-nearest neighbor model was trained on the obtained data to distinguish the signal strength of legitimate from illegitimate access points. To verify the authenticity of the data, an Evil Twin attack was physically simulated, and a machine learning model analyzed the data from the sensors. As a result, the Evil Twin attack was successfully identified based on the signal strength in the radio spectrum. The proposed model can be used in open access points as well as in large corporate and home Wi-Fi networks to detect intrusions aimed at substituting devices in the radio spectrum where IEEE 802.11 networking equipment operates.

Keywords: data; machine learning model; KNN; generative algorithm; intrusion detection; IEEE 802.11; Evil Twin attack

Citation: Banakh, R.; Nyemkova, E.; Justice, C.; Piskozub, A.; Lakh, Y. Data Mining Approach for Evil Twin Attack Identification in Wi-Fi Networks. *Data* **2024**, *9*, 119. https://doi.org/10.3390/data9100119

Academic Editor: Fabio Crestani

Received: 8 August 2024
Revised: 8 October 2024
Accepted: 12 October 2024
Published: 14 October 2024

1. Introduction

Today, the WPA3 security protocol is the most robust in IEEE 802.11 standard networks, but shortly after its release in 2018, it was found to be vulnerable to a group of attacks [1]. One of these attacks is the downgrade attack to WPA2, which is preceded by an Evil Twin attack. The Evil Twin attack can be identified using signatures, such as identifying an access point with an identical SSID but a different MAC address or frequency channel. However, if the attacker sets the same channel and the same MAC address on the Evil Twin as the legitimate access point, in other words, completely clones the data of the legitimate access point for transmitting information over the air, an intrusion detection system that operates by using signature methods will not be able to detect such an attack.

Fortunately, each wireless electronic device is unique in some way. Even if the devices were manufactured in the same factory and used the same technological process, there is no guarantee that they will have identical characteristics. The components that make up each device can slightly differ, which, in aggregate, can create a unique digital fingerprint for a specific device [2–7].

When it comes to radio devices, their digital fingerprint, in addition to everything mentioned above, will depend on the signal strength and the antenna used for transmission. This can aid in intrusion detection because if the characteristic signal strength variations in a legitimate access point are known in advance, a sudden significant change in these

variations will indicate the presence of another—illegitimate—access point, which will be evidence of an Evil Twin attack.

The propagation of radio waves in the range used by Wi-Fi networks depends on many physical factors: attenuation, free space path loss, multipath propagation, reflection, scattering, noise, and others. As had been studied in [8,9], the signal strength can be affected even by the presence or absence of a person in the room as well as by varying the signal strength. The authors of [10,11] presented a new approach to two-factor user authentication (2FA), combining the unique characteristics of the user's environment and machine learning (ML) to verify their identity. The authors used Wi-Fi radio wave transmission and ML algorithms to analyze the characteristics of the Beacon frame and the received signal strength indicator (RSSI) values from Wi-Fi access points.

A method for identifying MAC address spoofing by examining the signal strength from a device and using triangulation methods and sensors was investigated in [12]. The conceptual model was described and aimed to monitor the signal strength from the client devices, but it did not consider the impact of the environment on signal strength. Additionally, changes in signal strength during long time periods, such as several weeks, were not considered.

An approach to intrusion detection using the monitoring of network packets based on the frequency of their appearance through the air was proposed in [13]. The collected data were used to train a machine learning model. This method allows for detecting anomalies over the air and identifying a wide range of attacks on IEEE 802.11 networks. This method is highly likely to detect an Evil Twin attack, but we believe that a limitation of this approach is that, during physical interaction with the power supply, if the legitimate device is conditionally disconnected from the network, attackers can replace the legitimate access point with their own. In such a case, the proposed system will most likely not be able to distinguish this replacement.

Machine learning techniques have shown significant promise in wireless network security, particularly for detecting sophisticated attacks like Evil Twin. Sarker [14] comprehensively reviewed AI-driven methods across various domains, emphasizing their potential to enhance security protocols. In Wi-Fi networks, such approaches can effectively identify anomalies and threats in real time, aligning with the growing need for advanced security measures in increasingly vulnerable wireless systems.

In line with recent advancements in structural health monitoring, methods utilizing limited sensors have proven highly effective for damage detection and classification in complex systems. Sony [15] demonstrated that it is possible to achieve multiclass damage localization with a reduced sensor network by leveraging novel approaches like multivariate empirical mode decomposition and deep learning techniques. This strategy aligns with our research goals.

In evaluating machine learning algorithms for classification tasks, recent studies such as Uddin et al. [16] have demonstrated that various KNN variants can be particularly effective for predictive analytics. Their comparative analysis showed that optimized versions of KNN outperform the traditional approach in terms of accuracy and robustness, particularly in complex scenarios like disease prediction. These insights support the relevance of KNN in our study of detecting Evil Twin attacks in Wi-Fi networks.

Earlier studies have highlighted the versatility of the nearest neighbor algorithm for a range of classification tasks. According to Taunk et al. [17], the KNN algorithm, despite its simplicity, remains a powerful tool for learning and classification problems due to its non-parametric nature and ease of implementation. This reinforces the choice of KNN for our study in detecting Evil Twin attacks, where we focus on leveraging its classification capabilities in a network security context.

Monitoring the signal strength from access points using information from the Beacon packet over a long period and training a classifier to distinguish a legitimate access point from an Evil Twin were investigated by the authors of [18]. One of the advantages of this approach is the original method of data aggregation, on which the machine learning

model is subsequently trained using the k-nearest neighbor (KNN) algorithm. Additionally, in [19], the authors used the Random Forest machine learning algorithm to identify MAC spoofing attacks. Although both of these works showed accuracy rates of 100% and 94.83%, respectively, the authors noted limitations, such as using data on attack occurrences in areas where an attacker is likely to be present. Another limitation is the devices used to simulate the attacker, as the classifiers were trained only on a single physical device simulating an Evil Twin attack. Despite the high accuracy rates demonstrated in these works, factors such as using different equipment, more powerful antennas, or situations where the attacker is moving were not considered. These scenarios are quite difficult to replicate in real life as far as each case is unique. Therefore, creating a dataset containing all the signal strength sets differing from the legitimate ones makes sense. This dataset can be generated using an algorithm that targets the range of all the possible signal strengths available from IEEE 802.11 standard devices in the radio frequency spectrum.

This study took as its basis the approach to building a network infrastructure for intrusion detection and interaction with the attacker, as described in [20]. In the presented work for intrusion detection, the authors used independent sensors, operating on single-board computers that may not interact with each other. This approach allows forthe creation of an interference-resistant system that can use different communication channels to transmit the collected data.

The aim of this study is to develop an advanced approach for implementing a detection system of Evil Twin attacks based on the classification of radiation power datasets using the KNN method. In this approach, the data acquisition of radiation power is performed over a long period of time from spatially distributed sensors. Such data characterize the set of radio signal powers from the investigated legitimate access point and a variable set of surrounding emitters, present in real life (mobile phones, car radios, etc.). The collected data are unique for a particular room as well as the daily and weekly fluctuations of the ranges of their changes.

As a result of the performed research, the following data mining tasks are solved:

- Radiation power parameters are selected. The system of wireless sensors using the triangulation method for their measurement and data acquisition is experimentally implemented. Data acquisition is carried out for a sufficiently long time. The collected data are taken as those characterizing the system normal functioning.
- The collected data are supplemented with all possible values of the range using a special generation algorithm. The generated data are taken as characterizing the appearance of an illegitimate access point, that is, the implementation of the Evil Twin attack.
- Training of the intrusion detection system is carried out on the complete set of data using the KNN method.
- An experimental simulation of the Evil Twin attack is carried out. The intrusion detection system recognizes the attack. Intrusion detection quality characteristics are calculated.

The article is structured as follows: Section 2 meticulously describes the materials and methods of the study. Section 3 presents the comprehensive and meticulously conducted experimental results. Section 4 is dedicated to a thorough discussion of the obtained results. And finally, Section 5 provides well-considered conclusions and promising directions for future research.

2. Materials and Methods

2.1. Location of the Study

The research was conducted in the educational laboratory of Lviv Polytechnic National University, adjacent to the server room where the research object, namely the Wi-Fi access point, is located. The access point provides Internet access to students and staff of the department and may be a potential target for cybercriminals (Figure 1).

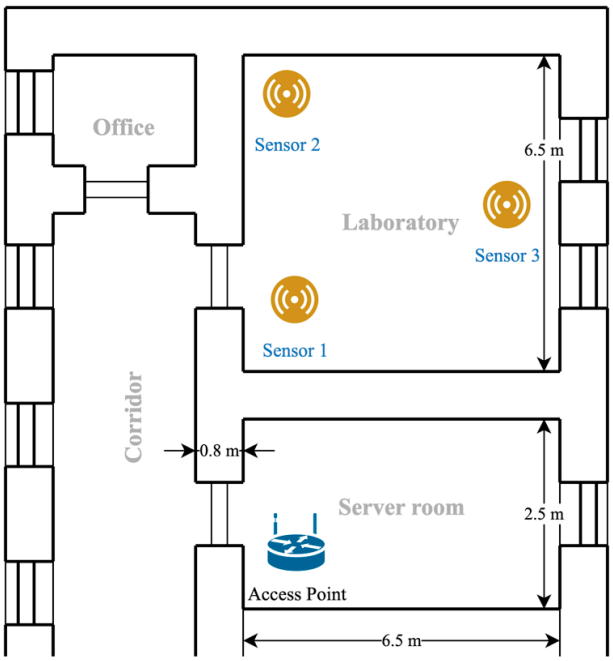

Figure 1. Room layout and equipment placement.

As shown in Figure 1, the sensors are arranged in a triangular formation, applying the triangulation method.

2.2. Hardware and Software

Access point UniFi AP-AC-LR was used as an object of research [21]. Raspberry Pi 4 single-board computers were used as energy-efficient and sufficiently powerful sensors for collecting and processing network packets [22]. The single-board computers were installed with the basic Linux Raspbian operating system without excess software. For signal strength monitoring, custom software was developed in Python 3 using the Scapy library, which allows for collecting and processing network packets [23]. The integrated Wi-Fi network card on the single-board computer does not support the monitor mode, so the Alfa AWUS036AXM network card, which uses the MediaTek MT7921AUN chipset, was chosen for these tasks [24]. The Alfa AWUS036AXM was selected from a list of recommended devices on the page of the open-source project for Wi-Fi networks penetration testing called Aircrack-ng as one of the most sensitive devices for working in the radio spectrum of IEEE 802.11 standard networks [25]. All three sensors had identical hardware and software configurations.

The time series database InfluxDB was used to record signal strength data, which allows for recording signal strength information from each sensor with a timestamp corresponding to when the data were received from a specific sensor [26]. To avoid overloading the service network of the sensors, a data aggregation method was used, which, at a specified period, selected the *max*, *min*, *average*, *mode*, and *number_of_packets* values of the signal strength from the access point. The *mode* metric was introduced to obtain a clear picture when detecting an attack. If an attacker moves, the *average* metric values during the attack might coincide with those of the legitimate access point. Introducing another metric that should account for the average value reduces the likelihood of false negatives. In this aggregation method, all values except the *average* are integers, and the average is a float, representing the arithmetic mean of all values.

As is known, the signal strength from an access point in Wi-Fi networks can range from −100 dBm to 0 dBm [27]. In natural conditions, these numbers fall within a somewhat narrower range, usually from −90 dBm to −30 dBm, where at −30 dBm, the signal is strong. At −90 dBm, any functionality is highly improbable. Nevertheless, this might still

be sufficient for monitoring the access point, although there will be a few packets due to the weak signal.

Obviously, the *min* value is always smaller than the *max* value, but in our case, for example, if there is only one packet during the iteration, the *min* and *max* will be equal. The same would happen with multiple packets, but the signal's stability is solid. Additionally, *mode* and *average* lie within the range between the *min* and *max* metrics (1)

$$min \leq avg,\ mode \leq max. \tag{1}$$

To generate a dataset that fits the proposed aggregation, an algorithm that fills all possible value intervals for *min*, *max*, and *mode* with a step of 1 was developed. For *avg*, a step within the interval (0; 1], for example, 0.5, can be set, as indicated in Algorithm 1.

Algorithm 1. Generate the array of all possible signal strength ranges.

```
Input: empty array R, step for feature average avg_step
for min = −100 to 0 do
    min = min + 1
    for max = min to 0 do
        max = max + 1
        for mode = min to max do
            mode = mode + 1
            for avg = min to max do
                avg = avg + avg_step
                * Append {min, max, avg, mode} into list R
            end for
        end for
    end for
end for
Output: Array with values R
```

Once a list of all the possible signal strengths is generated, it can be used for each sensor to adapt the data. From the generated dataset, it is necessary to remove data that falls within the range of values for the legitimate access point. Specifically, from each column of the dataset (*min*, *max*, *average*, and *mode*), the minimum and maximum values of the metric obtained from the measurements need to be selected, and all entries in the automatically generated dataset that fall within this range should be deleted (2).

$$\begin{aligned} R_{illegitim} = \quad & \left(\neg\left(x_{min} \in \left[x_{min_{min}} - b, x_{min_{max}} + b\right]\right)\right) \\ & \cap\left(\neg\left(x_{max} \in \left[x_{max_{min}} - b, x_{max_{max}} + b\right]\right)\right) \\ & \cap\left(\neg\left(x_{avg} \in \left[x_{avg_{min}} - b, x_{avg_{max}} + b\right]\right)\right) \\ & \cap\left(\neg\left(x_{mode} \in \left[x_{mode_{min}} - b, x_{mode_{max}} + b\right]\right)\right), \end{aligned} \tag{2}$$

where b is a buffer in dBm, defined to separate the legitimate range from the illegitimate ones. This way, we can obtain ranges with values that are atypical for the legitimate access point. By combining the data frame of the legitimate access point with the data frame generated based on data different from the legitimate one for each sensor, we obtain the data that can be used to train a machine learning model (Figure 2).

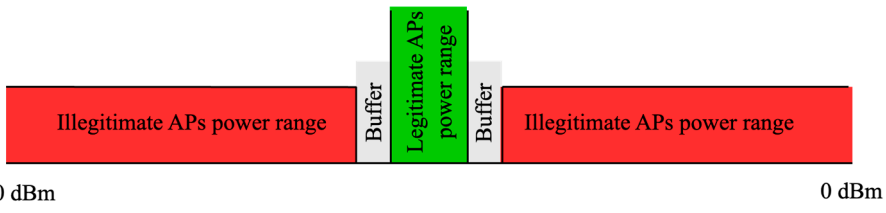

Figure 2. Schematic visualization of ranges generated by the algorithm.

In this study, the K-nearest neighbor (KNN) classifier was chosen as the machine learning method. KNN is a very simple algorithm to implement, which is based on the distance between points, chosen by a specified number of nearest neighbors (k), determined before training begins. KNN does not require the data to follow a specific distribution, as some other classification methods do, but instead compares the input point with the closest points in the feature space. KNN classification is based on the nearest neighbors, and if new data appear, the model can quickly adapt to changes without the need for retraining. This is quite relevant when working with Wi-Fi networks since the signal strength at the sensor locations can change with the appearance of obstacles in the room. Another factor influencing the choice of the KNN algorithm is that the described aggregation method has only 4 attributes for conducting the training. Otherwise, this algorithm might be unproductive, as the distance to all other records in the dataset would need to be calculated for each metric. Although there are many machine learning algorithms, such as logistic regression, decision trees, or SVMs (support vector machines), KNN proved the most suitable for several reasons.

KNN does not require explicit training of the model, which allows it to process new data quickly while maintaining high accuracy on already known samples. It is essential when working with Wi-Fi signals, as any new element in the room may change its physical properties.

With a dataset containing only five features, using more complex algorithms such as SVM was impractical due to their tendency to overtrain small features. SVM also requires careful selection of kernel parameters, which is computationally intensive and does not always guarantee improved results for small datasets.

Although easy to interpret, decision trees tended to overtrain even after regularization, leading to less robust results. Logistic regression, on the other hand, proved to be less efficient for tasks with non-linear dependencies between features. At the same time, KNN demonstrated flexibility in modeling non-linear decision boundaries without additional optimization. Time series analysis methods were not used in this study, as the initial approach to data processing focused on classification based on features that did not consider the time aspect. The dataset we used contains information about specific characteristics of connections, which is more suitable for classical classification methods such as KNN. In doing so, each sample in the dataset was treated as independent and not part of a sequence of events or change over time.

A time series analysis is usually applied to data that show clear temporal dependence or trends that develop over time. Modern Wi-Fi access points, usually those that work in mesh mode, are intelligent enough to find the weakest area with their clients that needs to be covered. This means that signal strength may be reoriented at any moment, and as shown in the next section, some sensors may obtain a more robust signal or the opposite. In our case, each sample describes a specific event or state of the system, and there is no pronounced temporal structure or relationships between events that would justify using such methods.

We experimented with other methods in the current research, but the KNN results provided the best balance between accuracy and computational efficiency. Moreover, the KNN algorithm's intuitiveness and ease of setup allowed us to focus on analyzing the results without complex optimization procedures, which is essential in our context.

3. Results

3.1. Data Extraction

Signal strength data were collected from the legitimate access point over two weeks. Data aggregated by the mean with a 15 min window is visualized in Figures 3–5.

In Figures 3–5, a clear daily cyclic pattern of signal strength changes can be observed. Regardless of the day of the week, every day around 7 AM, the *max* metric increases and the *min* metric decreases, while the *avg* and *mode* metrics remain stable and usually do not differ much from each other. Closer to 11 PM, the *min* and *max* metrics converge again.

We can also observe significant changes in the signal strength at random time intervals; in some cases, these changes are mirrored—if the signal strength increases on one sensor, it decreases on another.

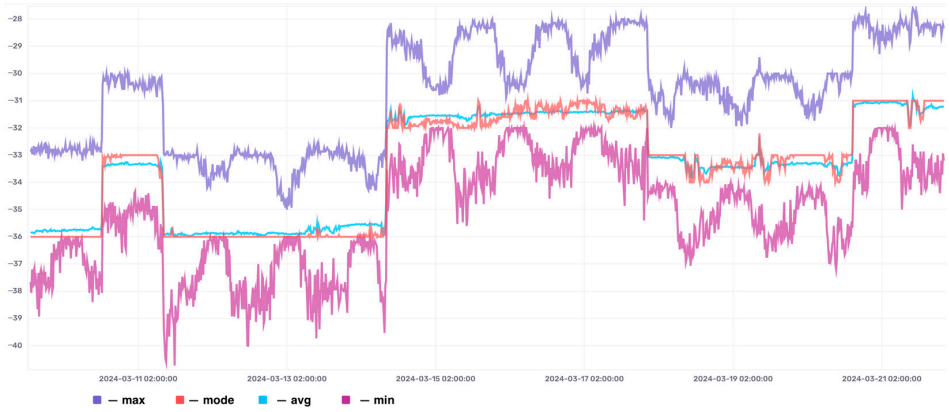

Figure 3. Signal variation for two weeks gathered by sensor #1.

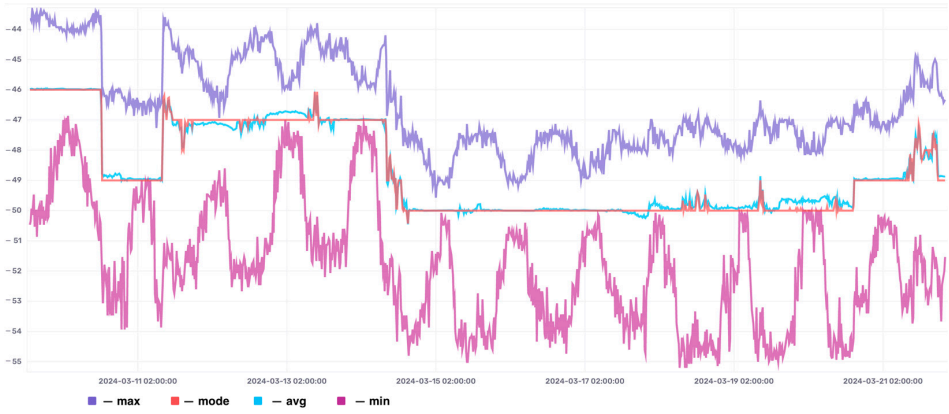

Figure 4. Signal variation for two weeks gathered by sensor #2.

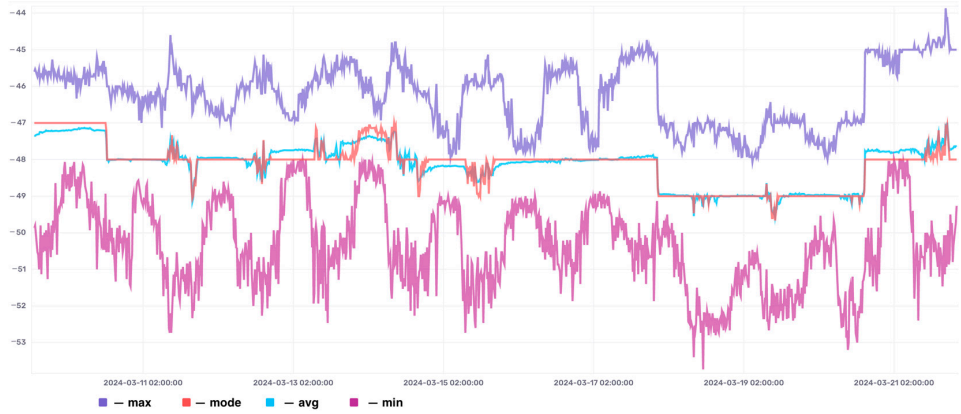

Figure 5. Signal variation for two weeks gathered by sensor #3.

Network load and interference can explain the main reason for the divergence of the *min* and *max* signal strengths during the day and their convergence at night.

During the day, when more devices are connected to the Wi-Fi network, it experiences greater loads [28]. This can cause a decrease in the minimum signal strength due to the increased interference and competition for access to the access point. More devices generate more radio signals, leading to higher noise levels and reduced signal quality, which is expected in a university building.

When activity decreases, the signal becomes more stable at night, and the range between *min* and *max* values narrows. This leads to a more stable signal and less variation between the minimum and maximum signal strengths. Fewer active devices at night decrease radio frequency noise, making the signal more stable. Thus, during the day, we observe a more comprehensive range between the *min* and *max* signal strengths due to a more significant number of active users and increased interference.

The increase in signal strength and noise levels can be explained by the fact that modern Wi-Fi access points use adaptive mechanisms to maintain connection quality. They can automatically increase the signal strength to maintain a stable connection when they detect increased noise levels or decreased connection quality due to interference. Each day, when more users connect to the network, the access point may increase signal strength to provide better coverage and service to more devices. This can lead to higher average and maximum signal strengths. Some devices may use power-saving modes, reducing signal strength at night when user activity decreases, which can also explain the lower maximum values at night.

Therefore, the signal level can increase during the day not only due to the increased noise but also due to the adaptive mechanisms of access points, higher user activity, and device placement. These factors contribute to higher average and maximum signal strengths even under increased noise conditions.

Figure 6 shows the frequency distribution of signal strengths from the legitimate access point for the three sensors that are monitored using four metrics.

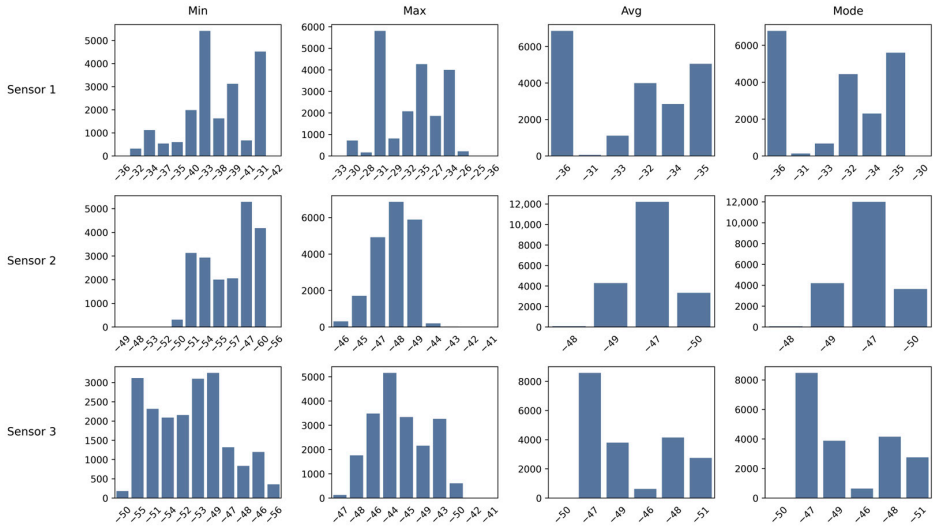

Figure 6. Signals' strength distribution.

As Figure 6 shows, each sensor has its own unique distribution of values. Some are highly strong, and some are weak. From the first view, weak values should be removed, for example, by applying percentile filtering, to avoid false positives in the future and to avoid the persistent dominance of illegitimate values; it was decided to leave them in place.

3.2. Data Analysis

After two weeks of monitoring the legitimate access point, an Evil Twin attack was simulated using the utilities airbase-ng and macchanger. These utilities allowed for the creation of a fake access point with the same attributes (SSID, MAC address, and channel) as the legitimate one. Over two hours, the device acting as the Evil Twin was moved around the room near the sensors and the legitimate access point. Figure 7 shows the variation in signal metrics during the simulated Evil Twin attack, as obtained from InfluxDB.

Figure 7. Variation in signals' strength gathered by all sensors during the Evil Twin attack.

Figure 8 provides a detailed depiction of the signal strength variation from one of the sensors during the simulated Evil Twin attack.

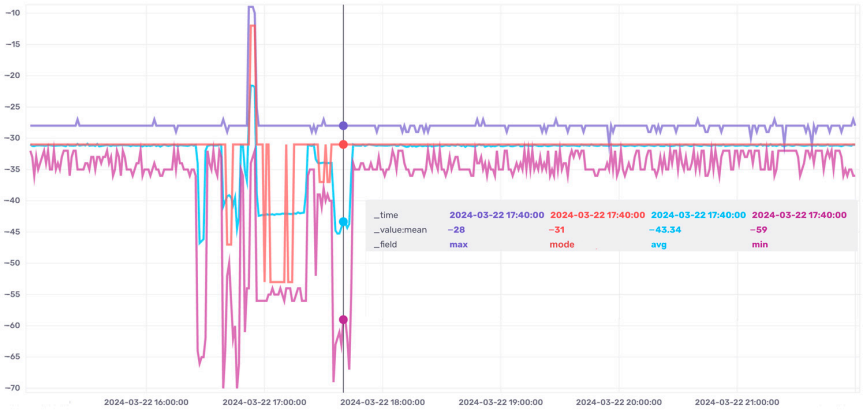

Figure 8. Detailed signals' strength gathered by sensor #1 during the Evil Twin attack.

As shown in Figure 8, the introduction of the *mode* metric is fully justified. In the absence of an attack, the *avg* and *mode* metrics are closely aligned, but during an attack, their values diverge. This divergence can significantly enhance the training of the machine learning model. The data were then exported from the InfluxDB database and imported into the Jupyter interactive computing platform [29]. After merging the data frame with legitimate data and the generated illegitimate data, we obtained three separate data frames for each access point. The data correlation is depicted in Figures 9–11.

Figures 9–11 show plots that illustrate the distribution of each metric individually. The main diagonal (from left to right) shows the frequency of data appearance (*Y*-axis) about a specific signal power in dBm (*X*-axis). Outside the main diagonal, there are scatter plots, illustrating the relationship between pairs of metrics.

From these plots, we can see that the values of the metrics (0—data from the illegitimate access point, 1—data from the legitimate access point) for *min* and *max* exhibit a distribution close to a Poisson distribution, while *avg* and *mode* exhibit a distribution close to a normal distribution. The Poisson distribution can be explained by the fact that minimum values occur more frequently closer to the lower boundary and cannot exceed the maximum value, and vice versa. The normal distribution for *avg* and *mode* is due to these values falling between *min* and *max* at each point, with a step of at least 1 dBm.

The scatter plots illustrate the relationship between the *legitimate* values of the target variable. From these plots, we can observe that the points of the illegitimate signal are evenly scattered, as predicted by the algorithm. There is no solid linear dependency between the two variables regarding the relationship between the data from the legitimate access point and the generated data, which could belong to a device performing an Evil

Twin attack. Based on the pairplot, we can conclude that the correlation between the two values of the target variable could be more robust, but this is expected since the points overlap because some signal strength values from the legitimate and illegitimate access points can coincide during an attack. For example, such a situation may occur when the legitimate access point has *min* = −50 dBm, *max* = −40 dBm, *avg* = −45 dBm, and *mode* = −44 dBm. If the attacker approaches and moves away from the access point during the Evil Twin attack, the average metric value will likely be close to that of the legitimate access point. Therefore, completely removing all values from the data frame that contain the signal strength of the legitimate access point is impractical.

The *corr()* function from the Pandas library was used to understand how well the data correlate with each other [30]. By default, this function computes the Pearson correlation coefficient to measure the linear correlation between the data frame columns. The Pearson correlation coefficient measures the linear dependence between two variables. It ranges from −1 to 1, where 1 indicates a complete positive linear correlation, −1 indicates a complete negative linear correlation, and 0 indicates no linear correlation.

A correlation heatmap, shown in Figures 12–14, was constructed to evaluate the metrics' correlation for each sensor in detail.

The threshold for a "low" correlation can depend on the context and specific application. However, generally, a correlation coefficient less than 0.3 or greater than −0.3 is often considered a low correlation [31]. Although the correlation is not very strong, we can still talk about a positive correlation since most features do not fall within the range [−0.3, 0.3]. The strength of the correlation varies depending on the pair of features. The strongest correlation is observed between the features *min* and *avg*. Overall, the features *min*, *max*, *avg*, and *mode* are closely related to each other, and this level of correlation usually indicates that they measure phenomena related to each other—in this case, the signal strength is from the same source.

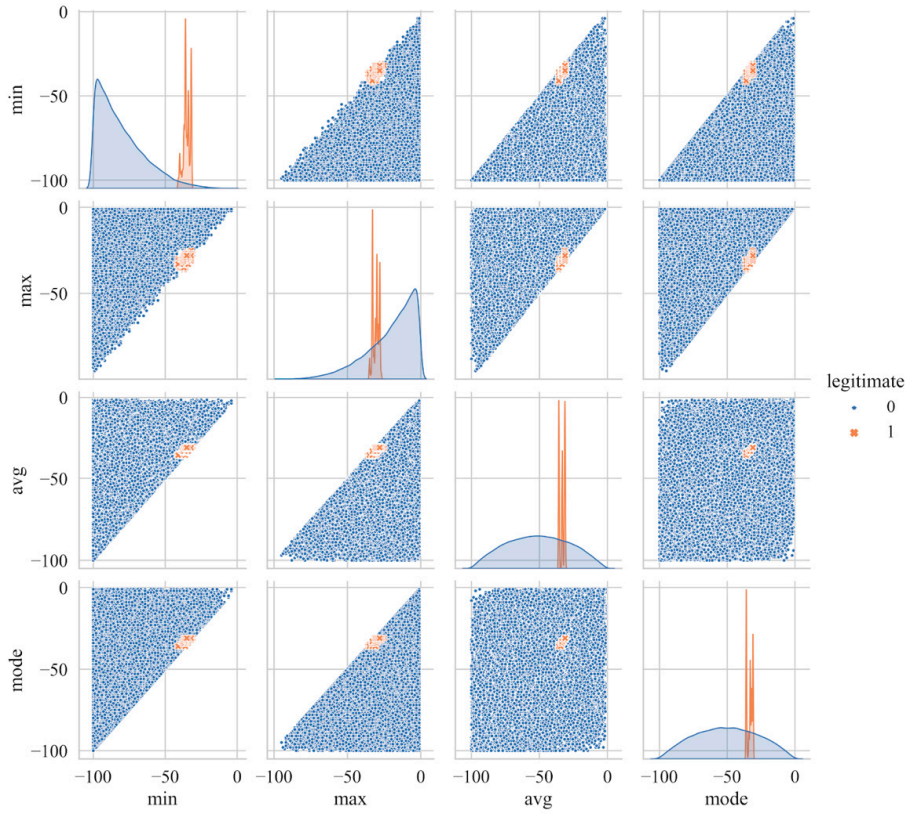

Figure 9. Visualization of legitimate and illegitimate data correlation for sensor #1.

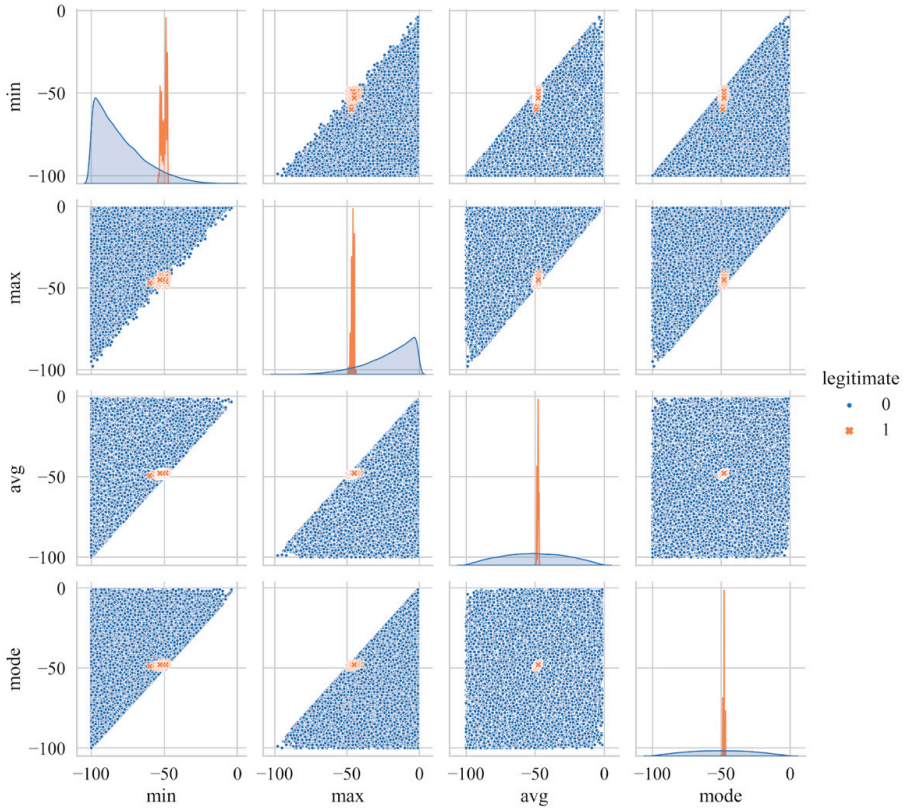

Figure 10. Visualization of legitimate and illegitimate data correlation for sensor #2.

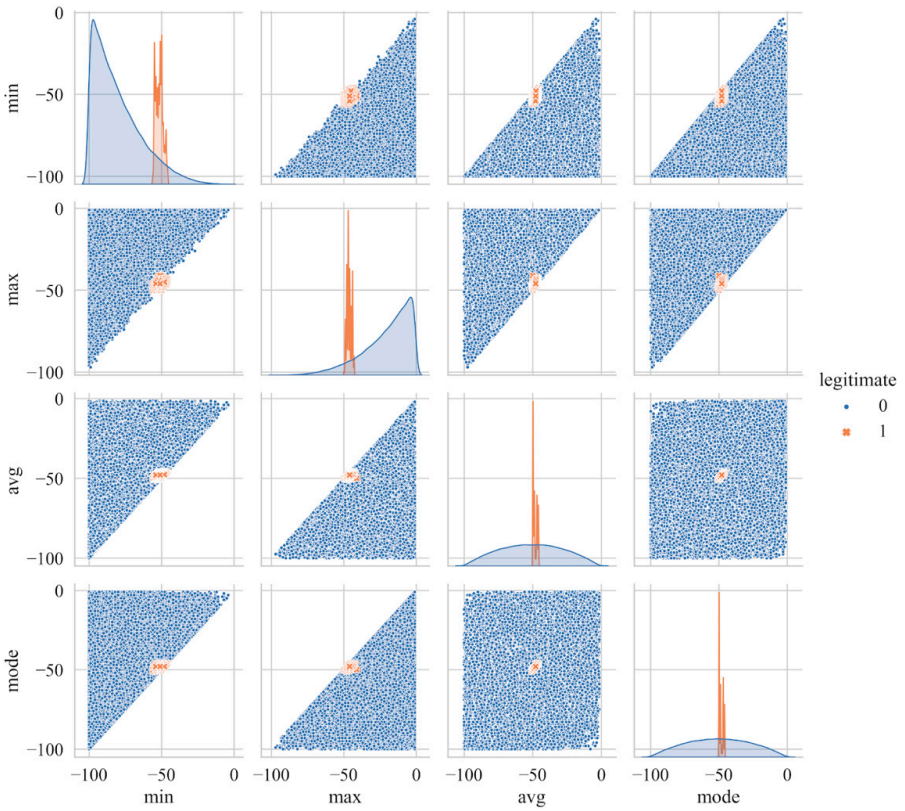

Figure 11. Visualization of legitimate and illegitimate data correlation for sensor #3.

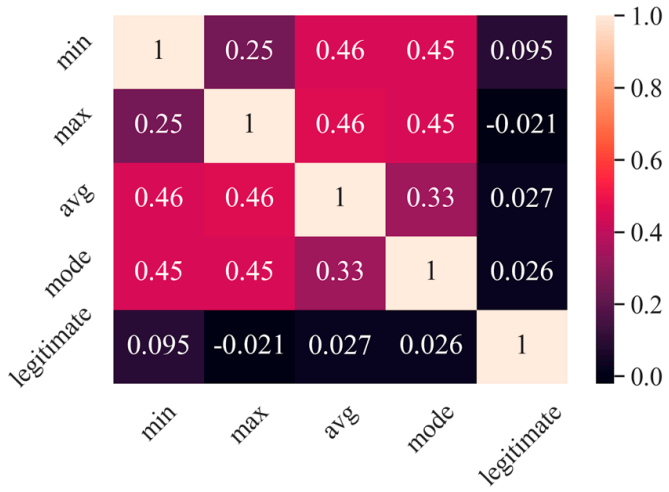

Figure 12. Heatmap of the metric correlation for sensor #1.

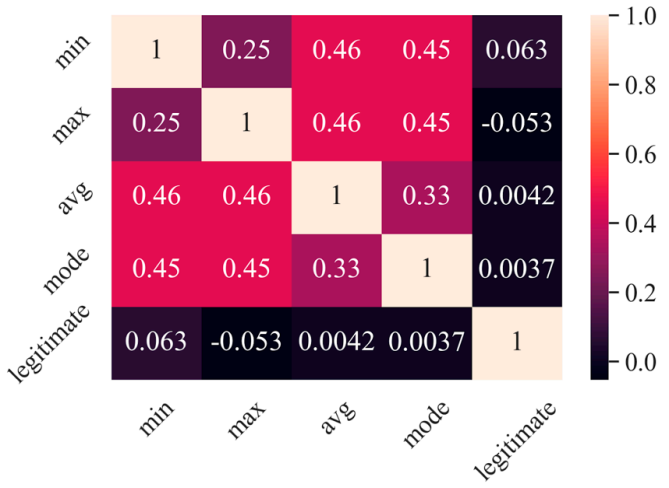

Figure 13. Heatmap of the metric correlation for sensor #2.

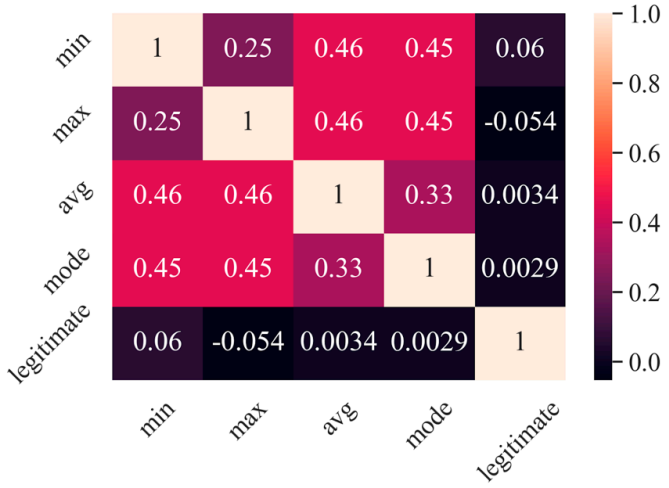

Figure 14. Heatmap of the metric correlation for sensor #3.

The feature that correlates the least with the others is *legitimate*, indicating the access point's legitimacy. This is because some metric values of the legitimate and illegitimate access points overlap in the same range. Hence, the correlation is weak in the data preprocessing stage. It happens since the generative algorithm creates a large number of records,

and even if all the records that satisfy the conditions of Formula (2) are removed, there still remain metric values intersecting with the legitimate ones.

3.3. Model Training

The next step was to split the data into training and testing datasets in an 80% to 20% ratio. Hyperparameter tuning was used to find the best parameters for the model training. To measure that, two additional classes were used: GridSearchCV from sklearn.model_selection and accuracy_score from sklearn.metrics [32]. The model was tested using a combination of three parameters—the number of neighbors (3, 5, 7, 9, and 11), weight (uniform and distance), and the power parameter for the Minkowski metric (1 and 2). With our dataset, the higher the neighbor number, the worse the accuracy score becomes. The degradation of the accuracy score started from seven neighbors.

Even though several combinations of parameters showed the best accuracy score, the number of neighbors for comparison was set to 3 to avoid a large number of computations [33]. The uniform weight was chosen, meaning all points in each neighborhood are weighted equally. The parameter for the Minkowski metric was assigned to 2, corresponding to Euclidian distance.

The *sklearn* library and the neighbors' KNeighborsClassifier class were used to train the machine learning model [34].

Figure 15 shows the confusion matrix based on the classification report results.

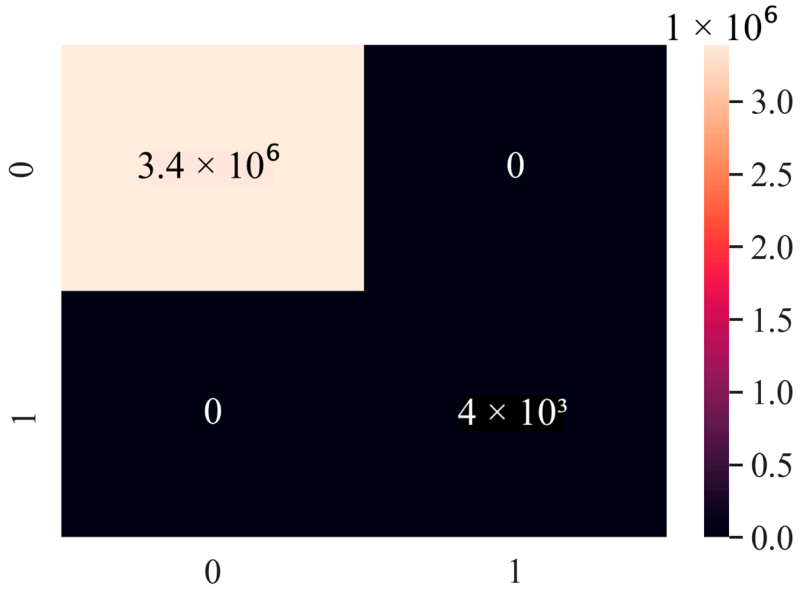

Figure 15. Confusion matrix for sensors.

This means that the model makes no classification errors. Although this scenario is relatively rare in practice, in this case, the model is quite simple, and sufficient data are available for training this model in our context. Additionally, the data for legitimate and illegitimate classes are clearly distinct.

As can be seen from Figure 15, the data on legitimate and illegitimate access points are not balanced. About 3.4 M illegitimate samples and 4 K legitimate ones got into the test set. This difference between the values is explained by the fact that the number of samples for the legitimate access point was collected every minute for two weeks, so the total number of all records is 20,160. Accordingly, 20% of this number is 4032 records.

The large amount of data about an illegitimate access point is explained by the fact that the generative algorithm, according to Algorithm 1, creates many records on each distance between *max* and *min*. The function *Single*(Δ) describes the number of *mode* and

avg records generated by Algorithm 1 for a given distance between *max* and *min*, taking into account $mode_{step}$ = 1 dBm and avg_{step} = 0.5 dBm (3)

$$Single(\Delta) = \frac{\Delta}{mode_{step}} \cdot \frac{\Delta}{avg_{step}} = \frac{2}{[dBm]^2} \cdot \Delta^2, \quad (3)$$

where Δ is the distance between *min* and *max* metrics.

The function $Sum(\Delta)$ displays the sum of all records that will be added to the dataset at segments with a specific distance (4).

$$Sum(\Delta) = Single(\Delta) \cdot \left(100 - \left(\frac{\Delta}{dBm} - 1\right)\right) = \frac{2}{[dBm]^2} \cdot \Delta^2 \cdot \left(101 - \frac{\Delta}{dBm}\right). \quad (4)$$

For example, 200 records will be generated on one of the segments with a difference of 10 dBm and with avg_{step} = 0.5 dBm. Figure 16 shows the change in the number of records when the distance between min and max changes. As can be seen from Figure 16, the function *Single* changes rapidly. With the distance is between *min* and *max* of 100 dBm, the algorithm will generate 20 K records. However, the 100 dBm distance can exist only in a single variant—on the interval from −100 dBm to 0 dBm. In contrast, at the 1 dBm distance, there will be 100 such segments.

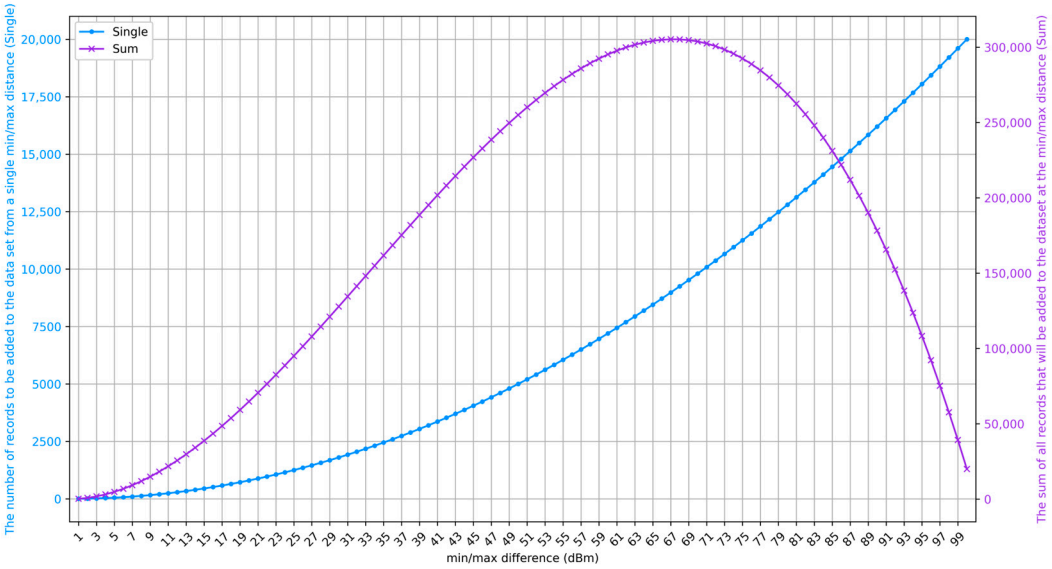

Figure 16. Characteristics of the distribution of the number of records in the dataset depending on the distance between the min and max metrics, obtained during the operation of the generative algorithm.

As can be seen from Figure 16, the peak point belongs to the interval with the difference between *min* and *max* of 67 dBm, where the number of records added to the dataset will be 305,252 records. The total number of records generated by the generative algorithm at avg_{step} = 0.5 dBm is determined by expression (5).

$$\sum_{\Delta=1}^{100} Sum(\Delta), \quad (5)$$

where the Δ step equals 1. Hence, there are more than 17.3 M records; 20% of this value will be about 3.4 M, corresponding to the value in Figure 15.

4. Discussion

4.1. Comparison of the Model's Prediction with the Signature

The feature *number_of_packets* was used to validate the model on real-world data, which were not utilized during the model training. An increase in the number of packets from the access point indicates an Evil Twin attack and is one of the signature methods used to detect such attacks [35].

The standard number of Beacon packets captured by a single sensor in the absence of an Evil Twin attack is approximately 590–600 packets per minute. To assess the effectiveness of the machine learning model, the data during the simulated attack period were analyzed, and the results were compared with the packet count graph (Figure 17).

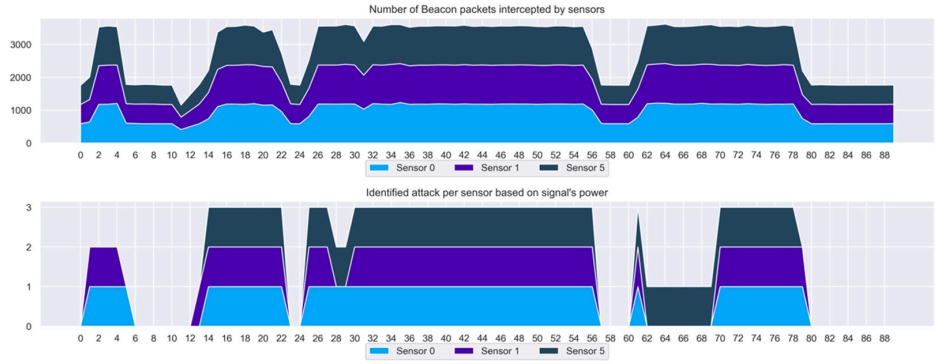

Figure 17. Number of packets indicated and identified by classifier comparison.

Figure 17 illustrates the relationship between signal power, acting as a signature, and intrusion detection using the trained machine learning model. Different colors are used to represent each sensor. As observed during time intervals when the number of packets increased, the model also reacted, identifying it as an attack. In contrast, the model did not identify any attacks where the packet count normalized. From Figure 17, monitoring the number of packets may be sufficient for intrusion detection, but some conditions will be described in the conclusions of this work.

4.2. Prediction Analysis

It is worth noting that not all sensors identified the attack simultaneously at some intervals. This can be observed in the interval 2–5, where sensor #3 did not identify any attack. In Figure 6, we can see that for sensor #1, $min \in [-42, -31]$, $max \in [-36, -25]$, $avg \in [-36, -31]$, and $mode \in [-36, -30]$; for sensor #2, $min \in [-60, -47]$, $max \in [-49, -41]$, $avg \in [-50, -47]$, and $mode \in [-50, -47]$; and for sensor #3, $min \in [-56, -46]$, $max \in [-50, -41]$, $avg \in [-51, -46]$, and $mode \in [-51, -46]$. In Table 1, the values of signal power in dBm intercepted by the sensors during the attack are provided.

The cells in Table 1 highlighted in red are the metric values that fall outside the range of the legitimate access point, and those in green are the values that fall within the range.

As shown in Figure 17 and Table 1, sensors #1 and #2 identified an attack in the range of 1–4, as the signal strengths did not fall within the legitimate range.

In the range of 28–29, sensor #2 stopped identifying the attack, although the attack had been detected earlier. A slightly larger interval, specifically 25–30, was examined to investigate this issue in more detail. In this interval, the behavior is somewhat unexpected because, if we look at timestamp 26, all metrics fall within the range of the legitimate access point. Still, the machine learning model classified these data as illegitimate. At the same time, the attack was not detected in the range of 28–29, where the *avg* metric does not fall within the range of legitimate signal strengths.

Table 1. Values of signal strength intercepted by sensors during the Evil Twin attack.

Sensor's id	**Sensor 1**				**Sensor 2**				**Sensor 3**			
Timestamp	***min***	***max***	***avg***	***mode***	***min***	***max***	***avg***	***mode***	***min***	***max***	***avg***	***mode***
0	−36	−28	−31.1	−31	−49	−45	−47.86	−48	−52	−46	−47.95	−48
1	−64	−28	−33.97	−31	−68	−45	−50.63	−48	−53	−46	−48.53	−48
2	−66	−28	−46.76	−31	−67	−45	−56.51	−48	−54	−46	−49.42	−48
3	−65	−28	−46.44	−31	−69	−45	−56.7	−48	−56	−46	−49.19	−48
4	−65	−28	−46.14	−31	−68	−45	−56.44	−48	−53	−45	−49.4	−48
5	−62	−28	−32	−31	−53	−45	−47.84	−48	−49	−46	−47.97	−48
Skipped												
25	−61	−28	−35.17	−31	−70	−42	−50.36	−48	−64	−32	−47.96	−48
26	−54	−28	−36.79	−31	−58	−43	−49.9	−48	−59	−35	−46.02	−48
27	−54	−9	−26.97	−31	−57	−35	−45.96	−47	−55	−27	−43.68	−48
28	−34	−9	−21.59	−12	−47	−41	−45.04	−47	−49	−37	−43.69	−40
29	−33	−9	−21.61	−12	−51	−41	−45.14	−47	−53	−37	−43.73	−40
30	−32	−10	−21.84	−12	−53	−34	−45.17	−47	−51	−28	−43.4	−40
Skipped												
59	−35	−29	−31.22	−31	−51	−45	−47.9	−48	−50	−46	−47.9	−48
60	−25	−28	−31.05	−31	−53	−45	−47.92	−48	−52	−46	−47.9	−48
61	−52	−29	−33.34	−31	−66	−45	−50.5	−48	−64	−46	−50.42	−48
62	−39	−28	−33.92	−31	−55	−45	−49.75	−47	−64	−45	−54.25	−61
63	−39	−28	−34	−37	−54	−45	−49.8	−47	−63	−45	−54.53	−61
64	−41	−28	−33.95	−37	−56	−45	−49.78	−47	−63	−45	−54.42	−61
65	−38	−27	−33.9	−37	−55	−45	−49.84	−47	−62	−45	−54.36	−61
66	−40	−28	−33.98	−31	−55	−45	−49.76	−47	−62	−45	−54.34	−61
67	−39	−28	−33.96	−37	−55	−45	−49.76	−47	−62	−45	−54.49	−47
68	−40	−28	−33.98	−37	−56	−46	−49.47	−47	−62	−45	−54.47	−47
69	−40	−28	−33.96	−31	−54	−46	−49.74	−47	−63	−45	−54.44	−47
70	−69	−28	−39.5	−31	−75	−47	−54.4	−48	−70	−45	−54.18	−48
71	−63	−28	−44.75	−31	−71	−46	−57.58	−48	−60	−46	−52.41	−48
Skipped												
77	−62	−28	−44.33	−31	−71	−45	−57.38	−48	−69	−46	−55.8	−48
78	−67	−28	−43.71	−31	−74	−45	−55.12	−48	−67	−45	−53.01	−48
79	−55	−28	−35.81	−31	−60	−45	−49.94	−48	−57	−45	−49.3	−48
80	−34	−28	−31.01	−31	−50	−45	−47.87	−48	−53	−45	−47.86	−48
81	−35	−28	−31.16	−31	−52	−45	−47.88	−48	−52	−46	−47.94	−48
82	−35	−28	−31.16	−31	−48	−45	−47.89	−48	−53	−45	−47.91	−48
83	−35	−28	−31.15	−31	−52	−46	−47.89	−48	−53	−45	−47.88	−48
84	−34	−28	−31.19	−31	−52	−45	−47.89	−48	−52	−46	−47.88	−48

Then, in the interval 62–69, sensors #1 and #2 stopped identifying the attack, although sensor #3 identified the activity as malicious. This can be explained by the fact that almost all metrics fell within the legitimate range during this interval, except for the *mode* metric of sensor #1. The model's behavior can be attributed to a slight deviation from the legitimate values, specifically by −1 dBm.

In the interval 77–84 in Table 1, the transition from identifying a state where an intrusion is detected to a state where no intrusion is recorded is shown. This is corroborated by the decrease in the number of packets detected and intercepted from the monitored access point, indicating the disappearance of one of the devices from the air. Importantly, this prevents us from determining whether the attacker left the monitored area or if the legitimate device malfunctioned.

The legitimate device's disappearance from the air is quite likely if the legitimate access point is using the WPA3 security protocol. This is because one of the steps that attackers often take during an attack on WPA3 is a DoS attack on the access point, aiming to disconnect clients and reconnect them to the Evil Twin.

Figure 17 shows that no false positive occurrences were observed, although false negative cases did occur. However, it is fair to consider that the situation might change in

other environments with different access points under observation. In our case, as shown in Figure 17, a single sensor trigger is sufficient. However, if false positive occurrences arise, a higher trigger threshold could be set, such as requiring two or more sensors to trigger simultaneously.

Overall, the trained KNN model using a generative approach to create data frames of illegitimate signal power allows for a fairly clear distinction between Evil Twin attacks.

One of the limitations of this approach is that before starting work, it is necessary to collect the required amount of data from the legitimate access point. When collecting this information, ensuring that no illegal actions are taken against the network is essential. It is also worth noting that the signal power may change when the access point's channel changes. Another factor could be various physical factors, such as additional furniture with metallic hardware in the office. In this case, the physical properties of the premises will be altered, which may affect the propagation of the signal and, therefore, its power. However, in this case, the KNN algorithm can quickly adapt to changes, and new data can be applied almost instantly.

The aim of the detection system described in the paper was to detect all the signal strength deviations outside of the legitimate power range, as illustrated in Figure 2.

Figure 7 illustrates the variation in signal strength gathered by all sensors during the Evil Twin attack, while Figure 8 illustrates that for sensor #1, and here, we can easily detect the presence of an Evil Twin due to a sharp signal and significant changes. Both Figures 7 and 8 demonstrate maximal values for the *max*, *avg*, and *mode* metrics when the Evil Twin's localization is closer than that of all three sensors to the legitimate access point.

In all other cases when the Evil Twin device is far from the access point, we observe a sharp decrease in the signal strength as well as the *min*, *avg*, and *mode* metrics. The authors are going to continue their earlier investigations [12] determining Evil Twin's device localization using more sensors, perhaps up to 7.

5. Conclusions

The Evil Twin attack has become more widespread in Wi-Fi networks since vulnerabilities in the WPA3 security protocol were discovered, as an attacker's success now depends on this attack when attacking a network using this protocol. However, this study did not conduct a full-fledged attack on WPA3 because it was necessary to gather information about the number of Beacon packets from the access point at the time of the attack to demonstrate the effectiveness of the machine learning model.

One of the steps an attacker takes during a WPA3 attack is Denial-of-Service, in which they initiate many authentication attempts. Due to the heavy encryption algorithms, the access point may struggle to handle the load and temporarily stop functioning. If the access point ceases operation, Beacon packets will also not be transmitted, rendering the intrusion detection method based on packet count ineffective. However, the intrusion detection method based on signal strength and artificial intelligence can effectively handle this scenario.

This work proposed a generative algorithm for automatically creating segments of illegitimate signal power. This algorithm covered all possible segment values when aggregating the minimum, maximum, average, and most frequently occurring signal power values, ensuring the identification of Evil Twin attacks regardless of the attacker's device configuration and the location of the attack.

As presented above, the system identified the Evil Twin attack in 100% of cases. In the configuration using three sensors during the attack, at least one sensor identified the attack at its occurrence. As demonstrated, this method may fail to recognize an attack if only one sensor is used. Therefore, it is logical to argue that coverage by at least three sensors is needed to monitor a single access point. However, if those three sensors receive packets from other access points with a power of more than -80 dBm and hence receive a sufficient number of packets, then they may cover more access points than one.

This approach can be applied independently and in conjunction with existing intrusion detection methods and tools to enhance the security of IEEE 802.11 networks against intrusions.

The interesting effect observed within this study was the variation in signal power throughout the day. During the day, the minimum and maximum values diverged, while at night, they converged. This effect could be utilized in training the model, not only based on the metrics of *min*, *max*, *avg*, and *mode* but also based on the distance between their power levels at specific times of the day.

Another limitation of this study is that monitoring the power from the legitimate access point occurred only on one of the fourteen channels available in the IEEE 802.11bg standard. Further development in this field could involve measuring power on the other channels and introducing another metric to represent the monitored channel number. This is important because signal power levels can differ when the access point changes its working channel. Therefore, when the channel changes, it is plausible that the intrusion detection system presented in this study could identify intrusions even in the case of their absence.

Author Contributions: Conceptualization, R.B., A.P., E.N., C.J. and Y.L.; methodology, R.B., A.P., E.N., C.J. and Y.L.; software, R.B.; formal analysis, C.J. and A.P.; investigation, R.B., A.P., E.N. and Y.L.; writing—original draft preparation, R.B. and E.N.; writing—review and editing, R.B., A.P., E.N., C.J. and Y.L.; supervision, E.N.; project administration, Y.L.; funding acquisition, E.N. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CRDF Global, grant agreement G-202401-71626 "Improvement of the complex of endpoints dynamic authentication by machine learning and protection from cyberattacks in the corporate networks", supported by the U.S. Department of State, the Bureau of European and Eurasian Affairs.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets and source code for machine learning are publicly available. URL: https://github.com/99stealth/wifi-evil-twin-detection-with-knn (accessed on 13 October 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Vanhoef, M.; Ronen, E. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. In Proceedings of the 41st IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 18–21 May 2020. [CrossRef]
2. Sikora, A.; Nyemkova, E.; Lakh, Y. Accuracy Improvements of Identification and Authentication of Devices by EM-Measurements. In Proceedings of the 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, Dortmund, Germany, 17–18 September 2020. [CrossRef]
3. Yang, C.; Sample, A.P. EM-ID: Tag-less Identification of Electrical Devices via Electromagnetic Emissions. In Proceedings of the 2016 IEEE International Conference on RFID, Orlando, FL, USA, 3–5 May 2016. [CrossRef]
4. Wang, X.; Zhang, Y.; Zhang, H.; Wei, X.; Wang, G. Identification and Authentication for Wireless Transmission Security Based on RF-DNA Fingerprint. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 230. [CrossRef]
5. Chen, S.; Xie, F.; Chen, Y.; Song, H.; Wen, H. Identification of Wireless Transceiver Devices Using Radio Frequency (RF) Fingerprinting Based on STFT Analysis to Enhance Authentication Security. In Proceedings of the IEEE International Symposium on Electromagnetic Compatibility, Beijing, China, 28–31 October 2017. [CrossRef]
6. Fadul, M.K.M.; Reising, D.R.; Loveless, T.D.; Ofoli, A.R. RF-DNA Fingerprint Classification of OFDM Signals Using a Rayleigh Fading Channel Model. In Proceedings of the Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019. [CrossRef]
7. Bihl, T.J.; Bauer, K.W.; Temple, M.A. Feature Selection for RF Fingerprinting With Multiple Discriminant Analysis and Using ZigBee Device Emissions. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1862–1874. [CrossRef]
8. Forbes, G.; Massie, S.; Craw, S. Wifi-based human activity recognition using Raspberry Pi. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence, Baltimore, MD, USA, 9–11 November 2020; Alamaniotis, M., Pan, S., Eds.; IEEE: Piscataway, NJ, USA, 2020. [CrossRef]

9. Jukić, D.; Domazet, S.; Ivanko, A.; Raca, D.; Nikolić, S.; Knežević, M.; Jović, F.; Raca, N.; Buljan, H. Determining the presence and the number of people by using a Wi-Fi signal. *Electr. Eng. Syst. Sci. Signal Process.* **2023**, arXiv:2308.06773v1. [CrossRef]
10. AlQahtani, A.A.S.; Alshayeb, T. Zero-Effort Two-Factor Authentication Using Wi-Fi Radio Wave Transmission and Machine Learning. *Comput. Sci. Cryptogr. Secur.* **2023**, arXiv:2303.02503v1. [CrossRef]
11. AlQahtani, A.A.S.; Alshayeb, T.; Nabil, M.; Patooghy, A. Leveraging Machine Learning for Wi-Fi-based Environmental Continuous Two-Factor Authentication. *Comput. Sci. Cryptogr. Secur.* **2024**, arXiv:2401.06612v1. [CrossRef]
12. Banakh, R.; Piskozub, A.; Opirskyy, I. Detection of MAC spoofing attacks in IEEE 802.11 networks using signal strength from attackers' devices. *Adv. Intell. Syst. Comput.* **2019**, *754*, 468–477. [CrossRef]
13. Kolias, C.; Kambourakis, G.; Stavrou, A.; Gritzalis, S. Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 184–208. [CrossRef]
14. Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 160. [CrossRef] [PubMed]
15. Sony, S. Preprint: Towards Multiclass Damage Detection and Localization using Limited Vibration Measurements. Ph.D. Thesis, University of Western Ontario, London, ON, Canada, 2021. [CrossRef]
16. Uddin, S.; Haque, I.; Lu, H.; Ali Moni, M.; Gide, E. Comparative Performance Analysis of K-Nearest Neighbour (KNN) Algorithm and its Different Variants for Disease Prediction. *Sci. Rep.* **2022**, *12*, 6256. [CrossRef] [PubMed]
17. Taunk, K.; De, S.; Verma, S.; Swetapadma, A. A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. In Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 15–17 May 2019. [CrossRef]
18. Banakh, R.; Piskozub, A.; Opirskyy, I. Devising a method for detecting "evil twin" attacks on IEEE 802.11 networks (WI-FI) with KNN classification model. *East.-Eur. J. Enterp. Technol.* **2023**, *3*, 20–32. [CrossRef]
19. Alotaibi, B.; Elleithy, K. A New MAC Address Spoofing Detection Technique Based on Random Forests. *Sensors* **2016**, *16*, 281. [CrossRef] [PubMed]
20. Banakh, R.; Piskozub, A.; Stefinko, Y. External elements of honeypot for wireless network. In Proceedings of the 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 23–26 February 2016. [CrossRef]
21. UniFi, Access Point AC Long-Range. Available online: https://store.ui.com/us/en/products/unifi-ac-lr (accessed on 6 July 2024).
22. Raspberry Pi 4 Model, B. Available online: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed on 6 July 2024).
23. Library Scapy. Available online: https://scapy.net/ (accessed on 6 July 2024).
24. AWUS036AXM Alfa Networks Inc. Available online: https://www.alfa.com.tw/products/awus036axm?variant=39913640198216 (accessed on 6 July 2024).
25. Aircrack-ng. FAQ. What Is the Best Wireless Card to Buy. Available online: https://www.aircrack-ng.org/doku.php?id=faq#what_is_the_best_wireless_card_to_buy (accessed on 6 July 2024).
26. Database InfluxDB. Available online: https://www.influxdata.com/ (accessed on 6 July 2024).
27. Kapgate, Y.; Vatti, R.; Jadhav, S. WiFi Tools and Signal Strength Analysis. *GRD J. Glob. Res. Dev. J. Eng.* **2017**, *2*, 17–21.
28. Syafrizal, N.; Pontia, F.; Tjahjamooniarsih, N. Analysis of Wi-Fi Network Quality in Tanjungpura University Library Building. *Telecommun. Comput. Electr. Eng. J.* **2023**, *1*, 13–22. [CrossRef]
29. Project Jupyter. Available online: https://jupyter.org/ (accessed on 6 July 2024).
30. Library Python Data Analysis (Pandas). Available online: https://pandas.pydata.org/ (accessed on 6 July 2024).
31. Mladenova, T.; Valova, I. Analysis of the KNN Classifier Distance Metrics for Bulgarian Fake News Detection. In Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021. [CrossRef]
32. Sckit Learn, API Reference, GridSearchCV. Available online: https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed on 6 October 2024).
33. Lopez-Bernal, D.; Balderas, D.; Ponce, P.; Molina, A. Education 4.0: Teaching the Basics of KNN, LDA and Simple Perceptron Algorithms for Binary Classification Problems. *Future Internet* **2021**, *13*, 193. [CrossRef]
34. Sckit Learn, API Reference, KNeighborsClassifier. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (accessed on 6 July 2024).
35. Thankappan, M.; Rifa-Pous, H.; Garrigues, C. A Signature-Based Wireless Intrusion Detection System Framework for Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks. *IEEE Access* **2024**, *12*, 23096–23121. [CrossRef]