STEM Education Review

REVIEW ARTICLE

# Machine Learning for Evil Twin Attack Detection

Norbert Awinbod Akparibo[1], Edem Kwedzo Bankas[2,*]

1 Department of Information Systems and Technology, School of Computing and Information Sciences, University of Technology and Applied Sciences, Navrongo, Ghana
2 Department of Business Computing, School of Computing and Information Sciences, University of Technology and Applied Sciences, Navrongo, Ghana

**ABSTRACT**

Evil Twin attacks are a significant threat to wireless networks because a malicious access point can impersonate a legitimate access point and induce clients to associate with attacker-controlled infrastructure. Existing detection approaches include received-signal-strength analysis, client-side traffic analysis, rogue-access-point monitoring, and multi-parameter detection. This study developed and evaluated a supervised machine-learning framework for detecting Evil Twin attacks using the Aegean WiFi Intrusion Dataset 3 (AWID3). The experimental pipeline included data cleaning, feature selection, standardization, class-imbalance handling, model training, comparative evaluation, and deployment of the best-performing model in a real-time monitoring interface. Random Forest, Logistic Regression, and Decision Tree classifiers were evaluated using accuracy, precision, recall, F1-score, false-positive rate, and the area under the receiver operating characteristic curve (AUC). Based on the experimental results, Random Forest and Decision Tree achieved 99.99% accuracy, while Random Forest obtained an AUC of 1.0000. Logistic Regression produced lower recall and a substantially higher false-positive rate. The selected Random Forest model was subsequently integrated with a real-time monitoring dashboard. Active wireless countermeasures were not executed; mitigation actions were simulated to avoid unauthorized disruption. The results indicate that multi-feature supervised learning offers a promising approach to Evil Twin detection.

**Keywords:** Evil Twin attack; rogue access point; wireless security; machine learning; Random Forest; AWID3; intrusion detection; WiFi

## INTRODUCTION

Wireless local area networks based on IEEE 802.11 have become an important access technology for Internet connectivity. Their widespread use has also increased the exposure of users and organizations to attacks that exploit weaknesses in wireless authentication, association, and management procedures. An Evil Twin attack is a form of rogue-access-point attack in which an attacker deploys an access point that impersonates a legitimate wireless network, typically by reproducing identifiers such as the service set identifier (SSID), with the objective of attracting legitimate clients. Once a client associates with the malicious infrastructure, the attacker may be positioned to observe, redirect, manipulate, or otherwise interfere with the victim's traffic[1,2].

Research has proposed several approaches for identifying fake or rogue access points. Client-side techniques have used received signal strength and statistical characteristics of 802.11 data frames, whereas network-side approaches have considered multiple wireless parameters and administrative monitoring[2–4]. These approaches demonstrate that no single observable feature is universally sufficient because wireless conditions vary with distance, interference, mobility, channel utilization, device characteristics, and attacker behavior.

***Corresponding Author:**
Edem Kwedzo Bankas, Email: ebankas@utas.edu.gh
Received: 8 February 2025; Revised: 17 April 2025; Accepted: 13 September 2025
https://doi.org/10.67229/stemer.26090025

The present study investigates whether supervised machine learning can combine multiple wireless features to distinguish legitimate access points from Evil Twin instances. The work uses AWID3, a publicly available wireless intrusion dataset containing contemporary attacks, including Evil Twin traffic[5]. The study compares Random Forest, Logistic Regression, and Decision Tree classifiers and subsequently integrates the best-performing model into a real-time monitoring prototype.

Traditional Evil Twin detection mechanisms can be sensitive to environmental conditions or depend on assumptions about attacker behavior. For example, signal-strength methods require suitable thresholds, while approaches that monitor de-authentication activity may be less effective when an attacker avoids generating conspicuous management-frame patterns. The research problem addressed in this study is how to develop a multi-feature detection model that can learn discriminative patterns from wireless traffic and provide timely identification of suspected Evil Twin access points.

The study developed a supervised machine-learning-based detection system capable of distinguishing legitimate access points from Evil Twin access points using relevant wireless traffic features. It identified the wireless parameters that contributed most significantly to classification performance and evaluated the effectiveness of Random Forest, Logistic Regression, and Decision Tree classifiers based on accuracy, precision, recall, F1-score, false-positive rate, and the area under the receiver operating characteristic curve (AUC). The best-performing classifier was then integrated into a real-time wireless monitoring prototype. Finally, the study designed a mitigation workflow that recorded and simulated appropriate response actions without executing unauthorized wireless countermeasures.

## BACKGROUND

Wireless networks have become an essential part of everyday communication, providing convenient internet access in homes, universities, businesses, healthcare facilities, and other organizations. However, the open nature of wireless communication also creates security risks. One such threat is the Evil Twin attack, in which an attacker sets up a fake access point that mimics a legitimate network. Users may unknowingly connect to this malicious access point, giving the attacker an opportunity to monitor their traffic, obtain sensitive information, or redirect them to harmful websites.

Detecting stealthy Evil Twin attacks can be difficult because attackers can make rogue access points closely resemble legitimate ones. For example, they may use the same SSID and similar network characteristics, making it difficult to distinguish the fake network from the genuine one. Traditional detection methods that rely on a single parameter, such as SSID, MAC address, or signal strength, may therefore generate false alarms or miss carefully disguised attacks. Machine learning offers a promising alternative because it can examine several wireless traffic characteristics simultaneously and learn patterns that may indicate malicious activity.

The Aegean WiFi Intrusion Dataset 3 (AWID3) provides a useful foundation for this type of research. The dataset contains realistic wireless network traffic and includes Evil Twin attack scenarios, making it suitable for training and testing supervised machine-learning models. Algorithms such as Random Forest, Logistic Regression, and Decision Tree can be trained to identify differences between legitimate wireless traffic and traffic associated with Evil Twin attacks.

Although previous studies have demonstrated the potential of machine learning for wireless intrusion detection, there is still a need for approaches that combine reliable detection, real-time monitoring, and safe mitigation. Therefore, this study proposes a Machine Learning-Based Detection and Mitigation of Stealthy Evil Twin Attacks in Wireless Networks framework. The study evaluates selected machine-learning classifiers, identifies the most important wireless features, and integrates the best-performing model into a real-time monitoring prototype. It also provides a controlled mitigation workflow for recording and simulating appropriate responses within an authorized environment, thereby supporting practical and responsible protection of wireless networks.

## LITERATURE REVIEW

### *Evil Twin Attacks*

An Evil Twin is a rogue access point that impersonates a legitimate access point and attempts to induce clients to associate with it. The attack is closely related to man-in-the-middle threats because the attacker can position the malicious access point within the communication path between a client and the intended network. Agarwal et al. describe Evil Twin attacks as a practical rogue-access-point threat and emphasize the difficulty of relying on simple signatures because legitimate and malicious network conditions may appear similar. Lu et al. similarly characterize the Evil Twin as a rogue access point that masquerades as a legitimate AP and evaluate a passive client-side detection method based on 802.11 data-frame characteristics[1,2].

### *Client-Side Detection*

Kim et al. proposed a client-side approach based on received signal strengths (RSS)[3]. The method collects RSS

measurements from nearby access points, normalizes them, and evaluates their similarity to identify potentially fake devices. The published record reports strong performance under the experimental conditions used by the authors, but the approach depends on signal characteristics that may vary in real wireless environments.

Lu et al. proposed a passive client-side method based on statistical characteristics of 802[2].11 data frames. Their ET-spotter implementation achieved 96% accuracy in distinguishing Evil Twin access points from legitimate access points in their evaluation. This result demonstrates the value of traffic-level features, while also illustrating that performance should be interpreted in the context of the experimental environment and feature assumptions.

### *Multi-Parameter and Network-Side Detection*

Vanjale and Mane developed a multi-parameter rogue-access-point detection approach and reported a 37.33% improvement in detection time relative to the methods considered in their study[4]. The work supports the broader argument that combining multiple network characteristics can reduce dependence on a single indicator.

Agarwal et al. proposed an intrusion-detection scheme for Evil Twin rogue access points and evaluated the approach in a laboratory environment[1]. Their study reported detection rates exceeding 92% and 100% accuracy across the reported experimental runs. Such findings provide motivation for automated detection, but laboratory performance should not be generalized automatically to heterogeneous operational networks.

### *AWID3 Dataset*

The Aegean WiFi Intrusion Dataset 3 (AWID3) was introduced by Chatzoglou et al. as an updated dataset for evaluating attacks against IEEE 802.11 enterprise networks[5]. The dataset is appropriate for this study because it includes modern wireless attacks and provides labeled network traffic that can support supervised learning. Nevertheless, dataset-specific performance can be affected by feature distributions, attack-generation procedures, class imbalance, and potential leakage between training and testing observations.

### *Research Gap*

The reviewed studies demonstrate the effectiveness of individual indicators, client-side traffic analysis, and multi-parameter approaches. However, the present study extends this line of work by combining feature selection, class-imbalance treatment, supervised machine learning, comparative model evaluation, and real-time prototype deployment within a single experimental pipeline. The principal methodological gap that remains after this study is external validation: high performance on AWID3 alone cannot establish the model's generalization to unseen enterprise networks.

## METHODOLOGY

### *Research Design*

An experimental research design was adopted. The methodology consists of five stages: dataset preparation, preprocessing and feature selection, class-imbalance treatment, supervised model training and evaluation, and prototype deployment. The source manuscript described both quantitative and qualitative components; however, the actual reported evaluation is predominantly quantitative, based on labeled network data and classifier performance metrics.

### *Dataset*

The study uses the AWID3 dataset developed for empirical evaluation of attacks against enterprise networks[5]. The Evil Twin subset contains labeled wireless traffic with multiple data types and a large number of candidate features. Only the variables retained after the study's preprocessing and feature-selection procedures were used for model development.

### *Data Cleaning and Preprocessing*

Columns containing excessive missing values were removed using the threshold specified in the supplied methodology. The target variable was separated from the predictors, categorical target labels were numerically encoded, and candidate predictors were filtered using keywords associated with wireless communication behavior, including SSID, BSSID, MAC, authentication, and signal-related attributes. Non-numeric predictors were then excluded from the final feature matrix.

Missing numerical observations were imputed using the mean. Features were standardized using the z-score transformation, $z = (x - \mu)/\sigma$, where x is the observed value, $\mu$ is the sample mean, and $\sigma$ is the sample standard deviation. Feature selection was performed with SelectKBest using the ANOVA F-test, retaining the 20 highest-ranked features. For a feature X and class variable Y, the ANOVA F statistic compares between-class and within-class variability.

### *Train-Test Split and Class Imbalance*

The data were divided into training and testing subsets using a 70:30 split. Synthetic Minority Over-sampling Technique (SMOTE) was applied only to the training data to increase representation of the minority class[6]. The supplied methodology also used Tomek Links to remove borderline majority-class observations; this combination is intended to improve class separation while avoiding contamination of the independent test set. The

Logistic Regression model additionally used class weighting.

### *Classification Models*

Three classifiers were compared. Random Forest is an ensemble method that aggregates multiple decision trees and can model nonlinear relationships[7]. Logistic Regression was included as an interpretable linear baseline. Decision Tree was included as a nonlinear, rule-based classifier. The comparative design provides a useful baseline-to-ensemble progression, although additional models such as gradient boosting and deep learning could be investigated in future work.

### *Evaluation Metrics*

The models were evaluated using accuracy, precision, recall, F1-score, false-positive rate (FPR), and area under the receiver operating characteristic curve (AUC). For binary classification, accuracy is the proportion of correctly classified observations; precision is the proportion of predicted positives that are true positives; recall is the proportion of actual positives detected; F1 is the harmonic mean of precision and recall; and FPR is the proportion of actual negatives incorrectly classified as positive. Logistic Regression hyperparameters were optimized using grid search with five-fold stratified cross-validation.

After model training and evaluation, the best-performing model was saved on the runtime machine and downloaded to a local machine for subsequent reuse and deployment. It was saved in Pickle format (.pkl) due to its widespread Python support and ability to preserve the model structure and learned parameters.

**Real-Time Detection:** The trained machine learning model was exported in pkl format and prepared for import into the Python-based detection script. The detection script uses Scapy in Kali Linux to capture wireless network traffic.

**Real-Time Mitigation:** After the detection script is deployed and running, the system automatically refreshes every ten (10) seconds, while the mitigation script can be executed once an Evil Twin attack is detected.

**Building and Evaluating the Real-Time Machine Learning-Based Detection System.** (Figure 1)

The Evil Twin Detection and Mitigation System consists of a Streamlit-based web dashboard that provides real-time monitoring and a historical overview of Evil Twin detections. It automatically refreshes every ten seconds to ensure that the system continuously monitors at short intervals for timely threat detection. Scapy is used to capture wireless network traffic in monitor mode in Kali Linux.).

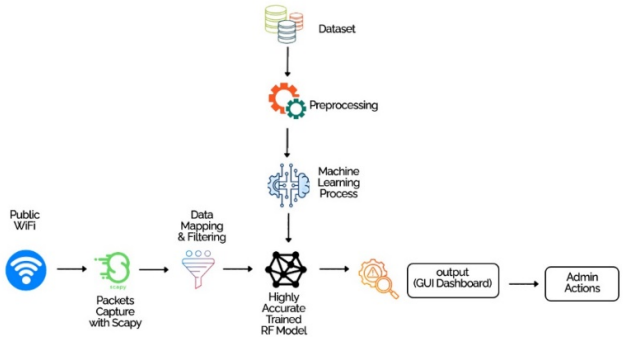


**Figure 1.** System Demonstration Diagram

During live detection, the features that are extracted from the beacon frames captured by Scapy did not directly align with the feature structure used by the trained model. To resolve this mismatch, a manual feature mapping strategy was employed. These values are then passed into a DataFrame, imputed and scaled using the same SimpleImputer and StandardScaler objects used during the training to maintain preprocessing consistency.

### *Simulating an Evil Twin Attack – The Setup*

To test the proposed Evil Twin Detection System's accuracy and robustness, a controlled simulation environment of an Evil Twin Attack had to be conducted. The attack was simulated using a pre-made off-the-shelf software available for free download from GitHub, named WiFi Attack Tool. (Figure 2)

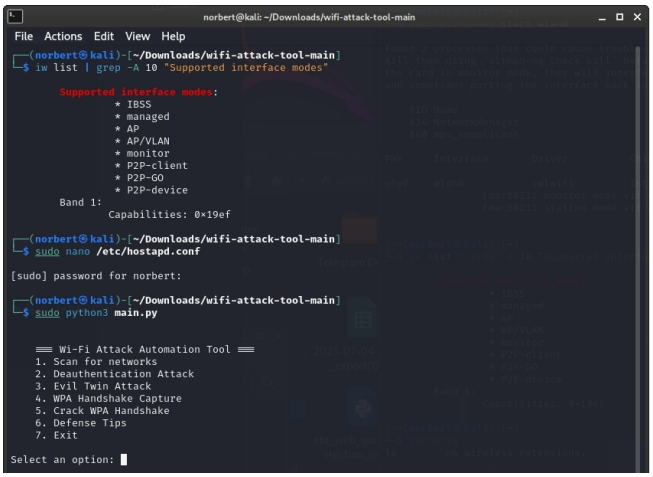


**Figure 2.** Simulating an Evil Twin Attack

### *Scanning and Selection of Target Network*

Before an attack was initiated in the simulated environment, a scan of the wireless environment was performed using the WiFi Attack Tool on Kali Linux. This scan identifies and displays all the available access points and their characteristics such as SSID, BSSID, channel, se-

curity type amongst others. (Figure 3)

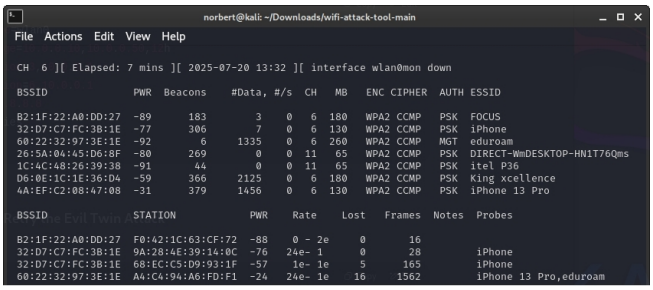


**Figure 3.** Scanning the network environment

### *Simulating the Evil Twin Attack*

After identifying the legitimate SSID to mimic, the tool is used to launch the Evil Twin Attack. This is done by choosing the Evil Twin Attack module in the selected tool interface; the SSID of the target access point was entered; the tool initializes the Evil Twin with the identical SSID and similar parameters configured using hostapd configurations. (Figure 4)

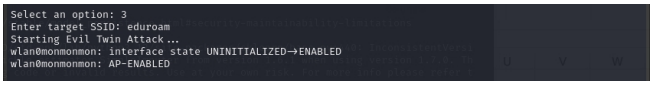


**Figure 4.** Initializing the Evil Twin Attack

While the Evil Twin Attack simulation was ongoing, the proposed real-time detection system was deployed on a separate system to detect the initiated attack.

### *Launching the System*

To run the detection script, all required libraries must be installed. Use the airmon-ng tool, which is provided by Aircrack-ng, to put the wireless network interface into monitor mode. Monitor mode allows the WiFi adapter to scan for network packets even if not directed at the host machine using the command: sudo airmon-ng start wlan0. After putting the wireless network interface into monitor mode, the detection and mitigation system can be launched. (Figure 5)

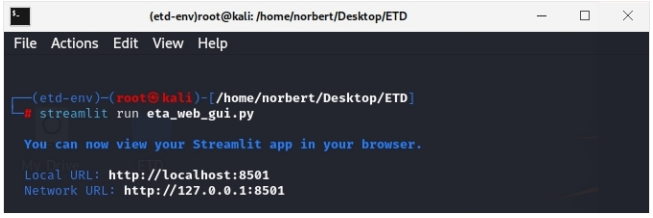


**Figure 5.** A screen capture of the Streamlit web GUI running.

This command will load the trained and saved Random Forest model (evil_twin_rf_model.pkl) and start real-time packet sniffing on the interface wlan0mon. This also initializes the background detection thread indicated earlier. The Streamlit web dashboard gets launched for real-time updates.

### *Real-Time Prototype*

The best-performing model was serialized and integrated into a prototype real-time detection application. Wireless traffic was captured in monitor mode, relevant features were mapped to the training representation, preprocessing objects were reused, and the classifier generated predictions for incoming observations. A Streamlit dashboard was used to display detection information and maintain logs. The prototype was evaluated in an isolated and controlled environment.

### *Controlled Attack Simulation and Ethics*

The attack simulation was conducted in a closed test environment using dedicated hardware. No real users were induced to associate with the malicious access point, and no real user credentials or communications were collected. Because active deauthentication or other countermeasures can disrupt legitimate wireless users, the study simulated mitigation responses rather than transmitting disruptive frames. Any future active deployment should be restricted to networks for which explicit authorization has been obtained.

## RESULTS AND DISCUSSION

### *Model Performance*

Table 1 is Performance of the evaluated classifiers based on the supplied experimental results.

Random Forest and Decision Tree achieved the highest reported accuracy, precision, recall, and F1-score, each at 0.9999. Random Forest produced an AUC of 1.0000, while Decision Tree produced an AUC of 0.9913. Logistic Regression achieved an accuracy of 0.8783, recall of 0.8786, F1-score of 0.9350, and AUC of 0.9053. The reported FPR values were 0.0172 for Random Forest and Decision Tree and 0.9747 for Logistic Regression.

These results indicate that the tree-based models captured nonlinear relationships in the selected feature space more effectively than the linear baseline. Random Forest was therefore selected for prototype deployment. However, the extremely high scores require cautious interpretation. An AUC of 1.0000 and accuracy of 99.99% on a single dataset may indicate strong separability within AWID3, but they do not establish robustness against changes in network topology, device vendors, channel conditions, attack implementation, or previously unseen attack variants.

**Table 1: Performance of the evaluated classifiers based on the supplied experimental results**

| Model | Accuracy | Precision | Recall | F1 | FPR | AUC |
|---|---|---|---|---|---|---|
| Random forest | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.0172 | 1.0000 |
| Logistic regression | 0.8783 | 0.9992 | 0.8786 | 0.9350 | 0.9747 | 0.9053 |
| Decision tree | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.0172 | 0.9913 |

Logistic Regression achieved a precision of 99.92% but performed poorly in terms of recall, with a score of 87.86%, and achieved an AUC of 0.9053. This indicates that while it correctly classified a high proportion of its predicted positive instances, it failed to identify a notable number of actual Evil Twins. Its false positive rate of 97.47% is exceedingly high, which also indicates a serious flaw in its ability to distinguish between legitimate and Evil Twin access points based on the selected features. This is consistent with the expectation that Logistic Regression is a linear model and may not be able to capture complex nonlinear interactions that are present in wireless traffic features and protocol behaviors. Considering the critical nature of wireless network security, where an undetected threat can have serious consequences, high recall and low false positives are essential. The Random Forest model's balance across all reported performance metrics, especially its robustness and ability to handle imbalanced and noisy data, made it the best choice for deployment into the real-time detection system. Random Forest's performance supports the use of ensemble-based learning techniques in intrusion detection systems, and this validates the effectiveness of the selected features (ie signal strength, authentication behavior, retry errors) in differentiating Evil Twin behaviors from legitimate WiFi behavior.

### *Performance Visualizations*

Figure 6 is Comparative classification Accuracy of the Evaluated Classifies.

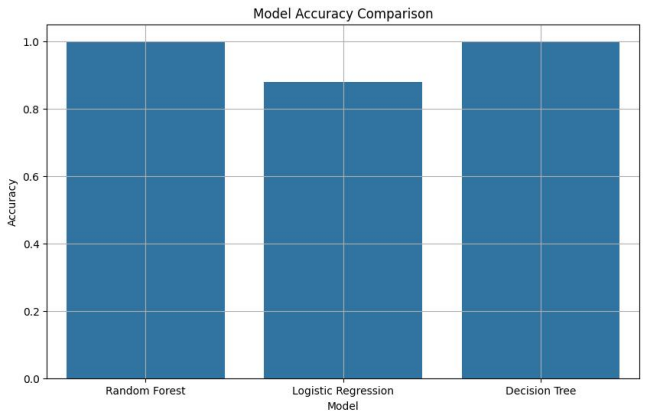


**Figure 6.** Comparative classification Accuracy of the Evaluated Classifies

This bar chart illustrates the classification accuracy of the three trained classifiers. Random Forest and Decision Tree achieved near-perfect accuracy scores (99.99%); they both outperform Logistic Regression (87.83). (Figure 7)

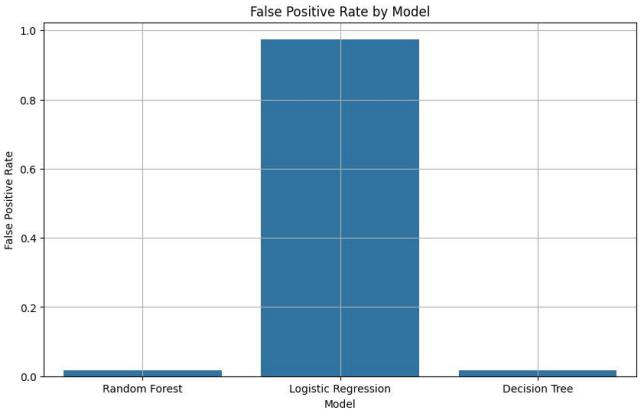


**Figure 7.** False Positive Rate by Model

From Table 7, Random Forest and Decisions Tree had low false positive rates of 1.72%. This indicates that they rarely misclassified legitimate access points as Evil Twins. On the other hand, Logistic Regression had a high false positive rate of 97.47%, making it unsuitable for this context despite its high precision. (Figure 8)

The Area Under the Curve (AUC) values go on to support the optimal performance attained by Random Forest (AUC = 1.0000) and Decision Tree (AUC = 0.9913) compared with Logistic Regression (AUC = 0.9053). (Figure 9)

### *Poor Performance of Logistic Regression*

Even with the systematic use of advanced imputation methods, over-sampling (SMOTE, SMOTE + Tomek), hyperparameter tuning, and five-fold stratified cross-validation, Logistic Regression still exhibited very poor performance, therefore affecting its false positive rate. This outcome may be related to both the statistical nature of the dataset and the mathematical assumptions surrounding Logistic Regression.

The extreme class imbalance and the limited representation of the minority class may have contributed to the

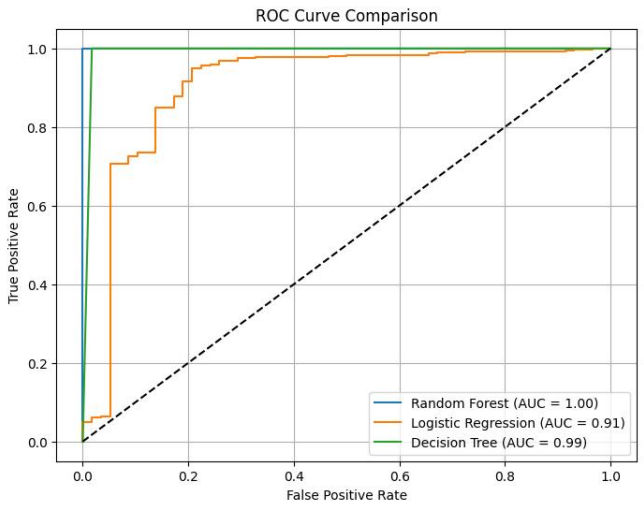


**Figure 8.** Receiver Operating Characteristic (ROC) Curve

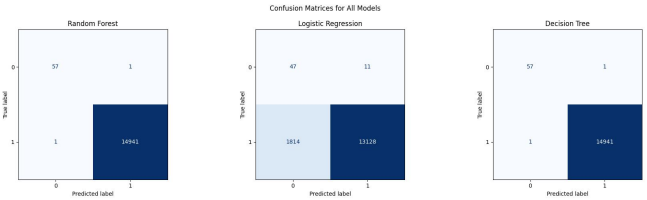


**Figure 9.** Confusion Matric for all models

performance. The minority class accounts for less than 0.5% (165) of all instances. Even though re-balancing techniques such as SMOTE + Tomek Links were applied to help mitigate class imbalance during the training process, the original imbalance and feature distribution remain in the unseen test set. In reality, in network security, the relevant attack and benign traffic patterns are most often non-linearly entangled in multiple dimensions and often exhibit nested manifolds[8]. Due to this, LR fails to differentiate minority instances that are embedded deep within the majority class distribution.

Moreover, tree-based models can be advantageous for nonlinear and imbalanced classification. RF and DTs have empirically proven to be more robust models in handling imbalanced datasets, especially when equipped with class-weighting or balanced sub-sampling[9].

### *Evil Twin Detection System*

Random Forest emerged as the best-performing model and was therefore exported and deployed within a Streamlit-based GUI. This live detection system received live inputs from ***wlan0mon***, the wireless network interface, through Scapy. The features that are extracted from live beacon frames were mapped to those used during training; the model classified incoming beacon frames in real time, logging each evil twin instance to evil_twin_log.csv, which is also available on the web GUI dashboard and can be downloaded with the click of a button. (Figure 10–12)

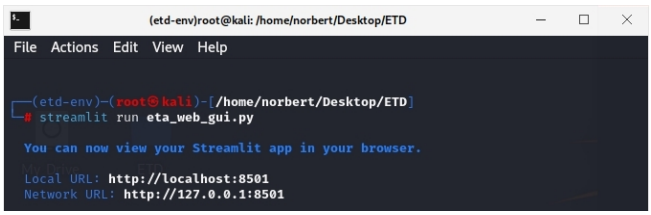


**Figure 10.** Running the web GUI dashboard with terminal

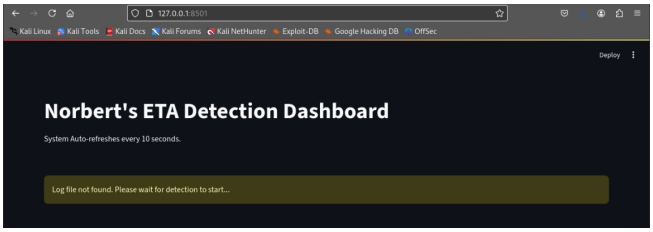


**Figure 11.** Web GUI dashboard before detection

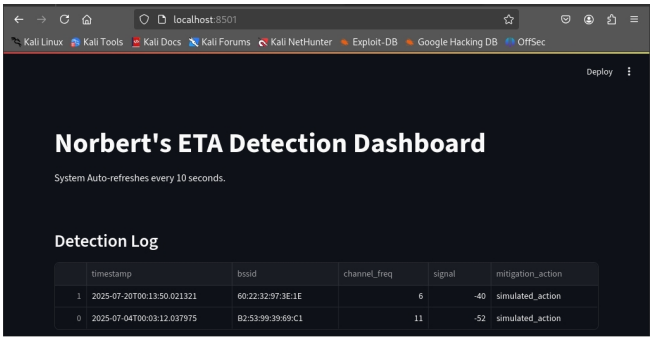


**Figure 12.** Web GUI dashboard after detection

The figure above is a screen capture of the task manager of the system running the detection system. Its lightweight nature helps it run properly without the need for any huge computational power, therefore making it ideal for ordinary systems. This indicates that the system can give optimal performance for any duration without system lags or any other technical difficulties.

### *Network parameters enhancing detection accuracy*

In order to identify which network parameters influence detection accuracy the most, a Random Forest model was analyzed for feature importance. The results shown in Fig…. below in the form of a chart illustrate that signal strength, retry flags, and certain authentication indicators are influential. The most influential feature was

**wlan_radio.duration**. This was followed by **radiotap.dbm_antsignal**, **wlan_radio.signal_dbm**, and **wlan.fc.subtype**. (Figure 13)

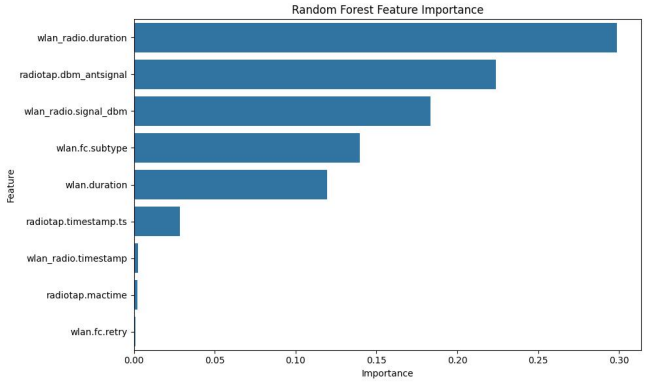


**Figure 13.** Most Influential Network Parameters

### Logistic Regression Performance

The Logistic Regression model produced a precision of 0.9992 but substantially lower recall and AUC than the tree-based models. Its reported FPR of 0.9747 is particularly problematic because it indicates that a very large proportion of legitimate observations were classified as attacks under the reported evaluation. The result is consistent with the possibility that the selected wireless features exhibit nonlinear class boundaries. Nevertheless, the unusually high FPR should be independently reproduced from the confusion matrix before publication because the manuscript's earlier narrative describes the value as a percentage while the table reports it as a proportion.

### Feature Importance

The supplied experiment identified wlan_radio.duration as the most influential feature, followed by radiotap.dbm_antsignal, wlan_radio.signal_dbm, and wlan.fc.subtype. These findings suggest that temporal, signal-strength, and frame-type characteristics contribute to the separation learned by the Random Forest model. Feature importance, however, should not be interpreted as causal evidence. Permutation importance or SHAP-based analysis would provide stronger interpretability in future work.

### Real-Time Detection

The trained Random Forest model was integrated into a Streamlit-based monitoring interface. The prototype captured wireless observations, mapped live features to the training representation, applied the same preprocessing pipeline, generated classifications, and recorded suspected Evil Twin instances in a log file. The prototype therefore demonstrates the feasibility of an end-to-end monitoring workflow rather than establishing production readiness.

### Mitigation Workflow

When the model classified an observation as an Evil Twin, the prototype generated a response event. The response actions included logging, simulated deauthentication, and recommendations for channel or credential-related defensive actions. Because the study did not execute active countermeasures, the reported system should be described as a detection system with simulated mitigation rather than a fully operational attack-response system.

### Comparison with Traditional Detection

The results support the proposition that combining multiple wireless features can outperform reliance on a single indicator under the experimental conditions. RSS-based detection can be useful but is sensitive to radio propagation and environmental variation[3]. Client-side statistical analysis can provide effective detection without requiring additional administrator-side infrastructure, as demonstrated by Lu et al[2]. Multi-parameter approaches similarly show the value of combining features[4]. The present approach differs by using supervised learning to learn a decision boundary from multiple selected features.

The study developed a supervised machine-learning pipeline for detecting Evil Twin attacks using AWID3 wireless traffic data. It presented a comparative evaluation of Random Forest, Logistic Regression, and Decision Tree classifiers to assess their effectiveness in distinguishing legitimate access points from malicious Evil Twin access points. The study also established a feature-selection and class-imbalance treatment pipeline incorporating SelectKBest, SMOTE, and Tomek Links, to improve the representation and preparation of the training data. In addition, it developed a real-time monitoring prototype by integrating the best-performing classifier with wireless packet-capture capabilities and a monitoring dashboard. Finally, the study implemented a mitigation workflow that recorded and simulated appropriate response actions within an authorized test environment, enabling the evaluation of mitigation strategies without executing unauthorized wireless countermeasures.

#### *Comparative Performance*

Table 2 is A table of comparative analysis.

The results indicate and support the idea that Machine Learning-based detection methods, particularly with Random Forest, can solve the issues faced by the traditional detection systems in place by using dynamic and multi-feature analysis.

#### *Mitigation Strategy Implementation*

Figure 14 is Mitigation Strategies Response.

**Table 2: A table of comparative analysis**

| Method | Accuracy | False positive rate | Key limitations | Proposed method benefits |
|---|---|---|---|---|
| RSSI-Based Detection | 90–97% | Moderate - High | Signal can be manipulated. | Uses multiple features beyond signal strength |
| MAC Filtering | Moderate (90%) | Stronger whitelist, low rate. | Needs frequent updates of lists. | No need for a whitelist; adaptive in learning. |
| Deauthentication Frame Monitoring | Low to Moderate | Moderate | Attackers can avoid sending frames | Uses multiple network parameters |
| Traceroute / Route Analysis | Moderate | Vulnerable to spoofing | Firewall hindrances cause replay attacks. | Does not rely on route consistency. |
| Client-Side Statistical Analysis | 90–96% | Variable | Sensitive to noise; needs training data | Passive, real-time, robust against noise |
| Proposed ML-Based Method | 99.9% (AWID3) | Very Low | Requires computational resources | High accuracy, scalable, detects in real-time. |

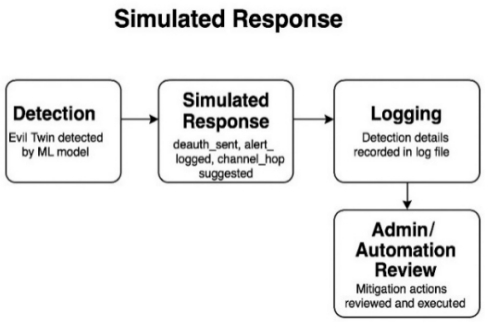


**Figure 14.** Mitigation Strategies Response

The main reason for this system is to detect Evil Twin Attacks in real time, the other part, very crucial, is to respond to these attacks once detected in a way that will best reflect best practices in both security and ethics. Due to the various legal and ethical restrictions involved, active mitigation and countermeasures like sending deauthentication frames or forcibly disconnecting devices could;

i. Violate regulations on wireless communication laws.

ii. Lead to a denial-of-service (DoS) condition unintentionally, which is a violation.

iii. Disrupt legitimate users who are on nearby networks.

Once an access point has been classified by the Random Forest Model as an evil twin, the detection system triggers the ***respond_to_threat(bssid)*** function. In order to not violate any wireless communication regulations or cause any sort of Denial of Service, while still enabling a robust response mechanism, the system implements simulated mitigation actions. These actions are logged for review and future execution under authorized environments. (Table 3)

During deployment, full deauthentication and automated channel hopping can be integrated into the system after securing all permissions regarding wireless network protocols.

Analyzing the results from the methodology, it became evident that Machine Learning detection methods and techniques excel and demonstrate higher performance when it comes to wireless network attacks, specifically evil twins.

Random Forest Classifier came out as the best-performing machine learning model after the training and testing across all metrics. It achieved an AUC of 1.0000, indicating a perfect class separation between legitimate access points and evil twins. It had high recall and low false positive rates. These two are essential in security-sensitive environments where a detection system missing an attack could have dire outcomes. The system built on the saved Random Forest Classifier model was able to process and detect simulated attacks in real-time, confirming the viability of deploying this system in real-time wireless environments.

Also, the feature importance analysis revealed the parameters that are most influential in differentiating evil twin attacks from legitimate network traffic. These were signal strength, frame duration, retry flags, and authentication flags.

The web GUI dashboard based on Streamlit helped to visualize feedback in real-time, showing the detected evil twins, the mitigation actions, and downloadable logs. This enhances usability for network administrators. The proposed mitigation strategies are industry standard and can be implemented in real life, with permissions obtained from the necessary regulatory bodies.

The system overall performed reliably without using much system resources, with the PC using just about 2% of its CPU. This was tested on a moderate-spec laptop. This proves its feasibility for deployment in low-re-

**Table 3: A summary table of mitigation strategies**

| Action | Description |
|---|---|
| **Alert Logged** | The detection is immediately saved to a log file, evil_twin_log.csv, for audit and monitoring. |
| **Deauthentication (Simulated)** | Indicates that the attacker or victim would be disconnected from the network. This does not actually do that since it is simulated. |
| **Automatic Certificate/Key Rotation (Trigger)** | New certificates are pushed to the clients connected to the network, and the old ones are revoked. This will contain the blast radius if credentials are exposed. |
| **Channel Hopping** | Channel switching for the legitimate access point once an evil twin is detected. |

source environments or even small enterprise networks.

## CONCLUSION

This study presented a supervised machine-learning approach for detecting Evil Twin attacks in wireless networks. Using the AWID3 dataset, the study compared Random Forest, Logistic Regression, and Decision Tree classifiers and integrated the best-performing model into a real-time monitoring prototype. Based on the supplied experimental results, Random Forest achieved 99.99% accuracy and an AUC of 1.0000, while Logistic Regression performed substantially worse under the reported evaluation. The findings support the usefulness of multi-feature machine learning for Evil Twin detection, but the results should be interpreted as dataset-specific until independently validated. The proposed prototype also demonstrates a safer approach to mitigation by logging and simulating response actions rather than transmitting disruptive wireless frames. Future work should focus on cross-dataset validation, leakage analysis, explainability, adaptive learning, and controlled real-world evaluation.

## DECLARATION

### *Author contributions*

*Conceptualization:* Norbert Awinbod Akparibo and Edem Kwedzo Bankas

*Investigation:* Norbert Awinbod Akparibo and Edem Kwedzo Bankas

*Methodology:* Norbert Awinbod Akparibo and Edem Kwedzo Bankas

*Formal analysis:* Norbert Awinbod Akparibo and Edem Kwedzo Bankas

*Writing – original draft:* Norbert Awinbod Akparibo and Edem Kwedzo Bankas

*Writing – review & editing:* Norbert Awinbod Akparibo and Edem Kwedzo Bankas

### *Acknowledgments*

Not applicable.

### *Funding*

No funding.

### *Ethics approval and consent to participate*

Not applicable.

### *Data availability statement*

Not applicable.

### *Conflict of interest*

The authors declare they have no competing interests or words to that effect.

## REFERENCES

1. Agarwal, M., Biswas, S., & Nandi, S. (2018). An efficient scheme to detect evil twin rogue access point attack in 802.11 wifi networks. International Journal of Wireless Information Networks, 25(2), 130-145.
2. Lu, Q., Qu, H., Zhuang, Y., et al. (2018). Clientside evil twin attacks detection using statistical characteristics of 802.11 data frames. IEICE Transactions on Information and Systems, E101-D(10), 2465-2473.
3. Kim, T., Park, H., Jung, H., et al. (2012). Online detection of fake access points using received signal strengths. 2012 IEEE 75th Vehicular Technology Conference (VTC Spring), 1-5.
4. Vanjale, S., & Mane, P. (2018). Multiparameterbased robust and efficient rogue AP detection approach. Wireless Personal Communications, 98, 139-156.
5. Chatzoglou, E., Kambourakis, G., & Kolias, C. (2021). Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset. IEEE Access, 9, 34188-34201.
6. Chawla, N., Bowyer, K., Hall, L., et al. (2002). SMOTE: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
7. Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.
8. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.
9. Fernández, A., García, S., Galar, M., et al. (2018). Learning from imbalanced data sets. Springer.