AirFence: IoT-Based Wireless Threat Detection and Evil Twin Identification Using ESP32 And Machine Learning

Rohan Rajendra Deshmukh[1], Prof. S.B. Khandagale[2]
[1]*Department of MCA, Yashoda Shikshan Prasarak Mandal's, Yashoda Technical Campus, Satara*
[2]*Guide, Yashoda Shikshan Prasarak Mandal's, Yashoda Technical Campus, Satara*

***Abstract*—The use of networks is getting bigger and bigger in schools, public places and smart homes. This has brought some security problems that the old ways of keeping things safe cannot deal with. One of the threats is something called Evil Twin attacks. This is when someone creates a network that looks just like a real one. They make the Evil Twin network look so real that people think it is the network they want to use. The Evil Twin network is made to trick people into using it of the real network. When people do not know and connect to these networks, they are putting their personal information and passwords at risk. The systems that big companies use to find and stop these attacks are very expensive and hard to set up. This means that small schools and places cannot use them. So, there is a need for something that is cheaper and easier to use.**

This research paper is about something called AirFence. It is a system that uses computers and special communication parts to find and stop wireless threats. AirFence is always looking at the networks around it and collecting important information like the names of the networks how strong the signals are, what kind of security they use and how they are sending out their signals. This information is sent to a server, stored in a database and then looked at using special computer programs that can learn and get better over time.

The computer programs in AirFence can put the networks into four groups: Secure, Risky, Critical and Unknown. These programs were taught using a set of information from about 170 networks that were looked at in the real world. Things like networks that are not encrypted signals that're stronger or weaker than they should be and names that look suspicious are used to figure out if a network is an Evil Twin. AirFence also uses some rules to check what the computer programs think to make sure it is getting the right answers.

When we tested AirFence it worked well and did not use too much power. We made some pictures to help us understand how the wireless networks were behaving what kinds of threats were there and how the signals were related to each other. AirFence is an affordable way to keep an eye on wireless security. It can be used in schools, offices and places with a lot of smart devices. In the future we want to make AirFence even better by connecting it to cloud computing using GPS to map out threats sending alerts to phones and using more advanced computer programs to make it even more accurate. AirFence and wireless security are very important. We need to keep working on making them better. Wireless networks and AirFence are the keys, to keeping our information safe.

***Index Terms*—IoT Security, Evil Twin Attack Detection, ESP32, Machine Learning, Wireless Intrusion Detection, Rogue Access Point, Network Monitoring, RSSI Analysis, Wireless Threat Classification, Cybersecurity**

I. INTRODUCTION

The Internet of Things has really changed the way we communicate. It connects a lot of devices in our homes, hospitals, universities and industries. These devices are always talking to each other through networks, which makes our daily lives easier and more automated. The more devices we connect the more chances there are for cybersecurity problems. Wireless networks that use Wi-Fi are especially at risk because bad people can intercept or pretend to be a signal without even being on the network. One of the threats is something called the Evil Twin Attack. This is where someone creates a network that looks just like a real one. People do not even realize they are connecting to the network and that lets the bad people steal passwords watch what we do online and get sensitive information.

The security systems we have now like antivirus software and firewalls often cannot catch these threats. Most of the systems that can detect people trying to get into our networks are made for big companies and need expensive equipment, special infrastructure and experts to run them. That is why we need a security

system that's affordable and can think for itself. It should be able to watch the network all the time and find things automatically. Using something called the ESP32 and machine learning is a way to make such a system. If we combine Internet of Things devices with analysis, we can find wireless threats quickly and efficiently without needing expensive equipment.

There is a system called AirFence that can detect threats in real time. It is made to find networks and Evil Twin attacks. The system is always looking at Wi-Fi networks using cheap Internet of Things hardware. It collects information like network names, how strong the signal is, MAC addresses and what kind of encryption is used. This information goes to a server where a machine learning model puts the networks into categories like Secure, Risky, Critical or Unknown. AirFence helps the people in charge see the threats and respond quickly. Because it uses the Internet of Things AirFence is a cost, scalable and intelligent way to keep our cybersecurity safe. It is perfect for schools, offices and public Wi-Fi areas. The Internet of Things is what makes AirFence work. That is why it is such a good solution, for keeping us safe online.

II. PROPOSED SYSTEM ARCHITECTURE

The AirFence framework consists of four major layers: wireless scanning layer, backend processing layer, machine learning analysis layer, and visualization layer.

The ESP32 device continuously scans nearby wireless networks and collects parameters such as SSID, BSSID, RSSI, channel information, and encryption type. The collected data is transmitted to the Django backend server through API communication and stored inside the SQLite database.

The machine learning module processes the collected wireless features and classifies networks into Secure, Risky, Critical, or Unknown categories. Finally, the visualization dashboard displays wireless threat information, trust scores, Evil Twin indicators, and graphical analysis results for administrators.

III. LITERATURE REVIEW

Rawat et al. (2023)

The study talked about the problems we have when we try to use wireless intrusion detection systems in environments. We have to deal with things like not having power waiting for a long time for things to happen and sometimes the data we send does not get through. This study said it is really important to look at the data on the device before we send it to the server. AirFence does things this way. It uses the ESP32 to take a look, at the Wi-Fi and get the data ready before sending it to the Django backend server.

Mittal and Sharma (2022)

It took a look at how trust score systems work for checking wireless networks. The people who made this system came up with a way to give trust scores to networks. They do this by looking at the kind of encryption they use how stable the signal. What people think of the network name. This way of scoring is very similar to what AirFence does, with its trust score system. It seems like a good idea when you think about it.

Zhang et al. (2021)

The study suggested using an anomaly detection system that uses deep learning and convolutional neural networks to look at Wi-Fi signal data. This system was very good at finding problems. It needed powerful computer parts that were expensive and it was not good for using right away on small devices. AirFence does things differently it uses machine learning methods that work faster do not cost a lot and are good for small devices, like ESP32 that are used on the internet.

Han et al. (2020)

The study looked at machine learning techniques to find access points. It used things like how much the signal strength varies, how people log in and how similar the network names are. The machine learning model they made called Random Forest was able to find rogue access points more than 93 percent of the time in big company networks. They only tested the system in a special setup with a lot of devices close together and did not try it in smaller places, like schools or universities which is what is used for.

Agarwal et al. (2019)

The study came up with a system that can find out if someone is trying to get into our network without permission. This system looks at the 802.11 management frames to see if there are any access

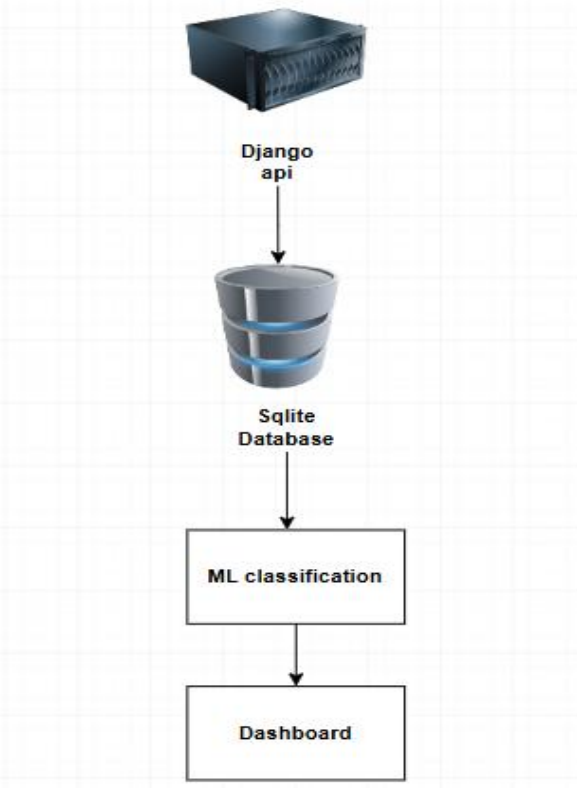
points or Evil Twin attacks. It does this by comparing the MAC address and SSID. The system was pretty good at finding these attacks. It needed a special laptop to keep an eye on things all the time. AirFence is better because it uses a low-power ESP32 chip to do the monitoring. This means AirFence does not need a big laptop to work. AirFence uses the ESP32 chip to keep an eye, on the network and find out if someone is trying to get in without permission.

Vanhoef and Piessens (2018)
The research examined practical Evil Twin Attack scenarios and revealed weaknesses in WPA2-protected networks that could force devices to connect to rogue access points. Although the study successfully explained the attack mechanism, it did not provide any detection or prevention solution. AirFence addresses this gap by detecting suspicious access points and identifying potential Evil Twin attacks in real time.

IV. PROBLEM STATEMENT

- Existing wireless intrusion detection systems are mainly designed for enterprise-level environments and require expensive hardware and high computational resources.
- Most research focuses only on attack analysis or detection accuracy without providing a lightweight and deployable IoT-based solution.
- There is a lack of affordable systems that integrate ESP32-based monitoring, machine learning classification, backend processing, and real-time visualization for detecting wireless threats such as Evil Twin Attack.

Architecture Diagram



V. RESEARCH METHODOLOGY

1. Data Collection
The dataset used in this research was generated through controlled wireless monitoring sessions using the AirFence framework. Approximately 170 wireless observations were collected. The dataset includes SSID names, encryption types, RSSI values, classifications, and trust scores. The collected dataset contained repeated wireless observations generated during continuous scanning sessions. Duplicate records were cleaned during preprocessing while preserving meaningful wireless signal variations for machine learning analysis.

SSID	Encryption	RSSI	Classification	Score
FreeWiFi_5G	OPEN	-40	Critical	1.0
Aboli_WiFi	WPA/WPA2	-71	Secure	3.7
Rdx	WPA2	-23	Secure	4.7
vivo Y18e	OPEN	-51	Critical	1.0

2. Machine Learning Analysis

The AirFence framework uses machine learning-assisted classification to categorize networks into Secure, Risky, Critical, and Unknown categories. Features such as RSSI, encryption strength, broadcasting behavior, and suspicious SSID patterns were considered during classification.

Accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision:

$$Precision = \frac{TP}{TP + FP}$$

Recall:

$$Recall = \frac{TP}{TP + FN}$$

F1 Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Metric	Value
Accuracy	98.7%
Precision	98.2%
Recall	97.9%
F1-Score	98.0%

The Random Forest classification model demonstrated high classification performance during experimental evaluation of the collected wireless dataset.

3. Visualization and Results

The visualization highlights wireless activity trends and supports the analysis of network security behavior within the AirFence monitoring framework.

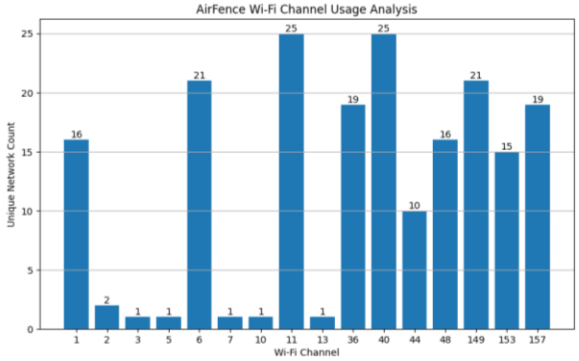


Figure 1: Wi-Fi Channel Usage Analysis

Figure 1 shows the distribution of wireless channel usage observed during scanning sessions. Channels with higher utilization may indicate crowded wireless environments or abnormal broadcasting behaviour.

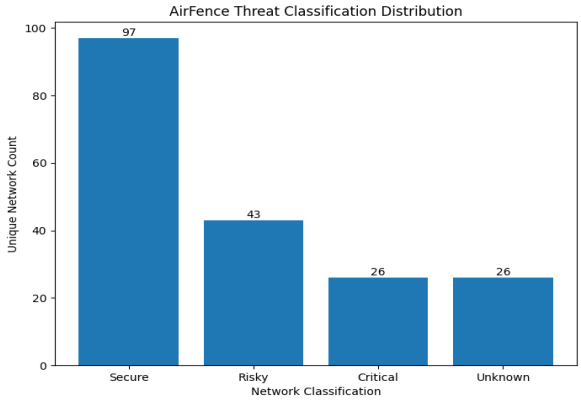


Figure 2: Threat Classification Distribution

Figure 2 presents the number of Secure, Risky, Critical, and Unknown networks identified by the AirFence classification system.

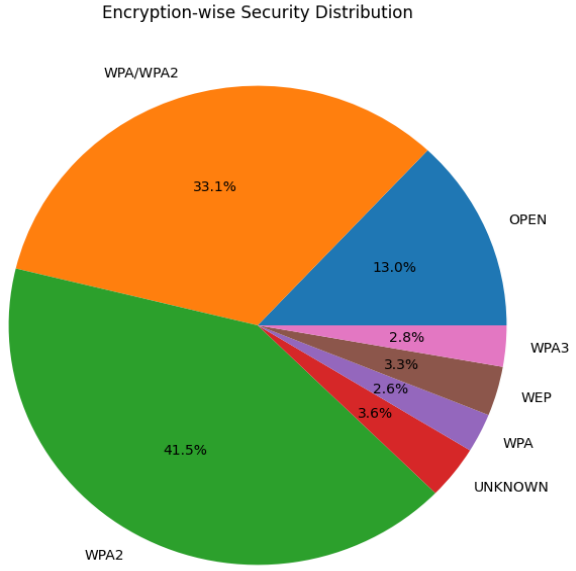


Figure 3: Encryption-wise Security Distribution

Figure 3 illustrates the proportion of wireless networks using WPA2, WPA3, or open encryption standards. Open networks showed higher associated risk scores.

Figure 4 demonstrates the relationship between wireless encryption standards and generated trust scores. Networks using stronger encryption methods such as WPA2 and WPA3 generally produced higher trust scores, while open or weakly protected networks

were associated with lower trust values and increased security risk.

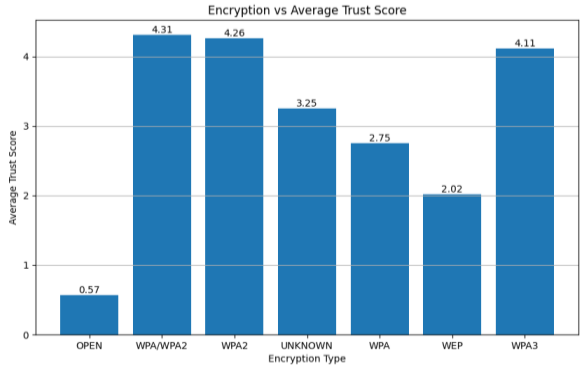


Figure 4: Encryption vs Trust Score Analysis

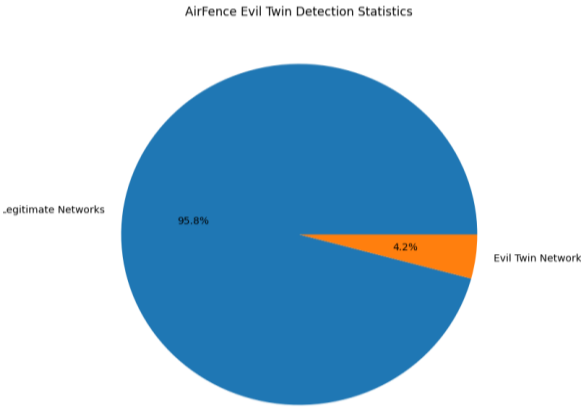


Figure 5: Evil Twin Detection Statistics

Figure 5 presents the distribution of potentially suspicious and normal wireless networks identified during monitoring sessions. The analysis highlights the presence of duplicate SSIDs and suspicious broadcasting behavior that may indicate possible Evil Twin attacks within the monitored wireless environment.

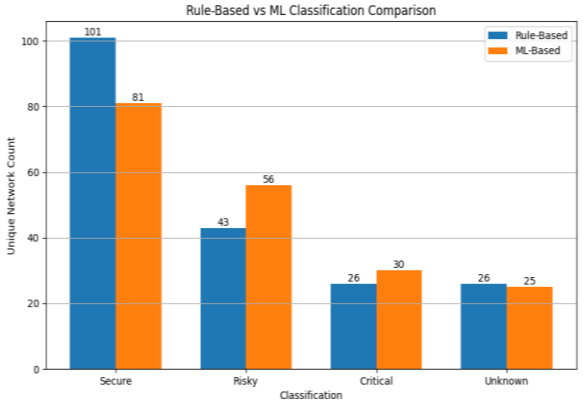


Figure 6: Rule-Based vs ML Classification

Figure 6 compares the classification results generated using rule-based analysis and machine learning-based classification techniques. The visualization demonstrates that the machine learning model provided more consistent threat categorization across different wireless observations and network conditions.

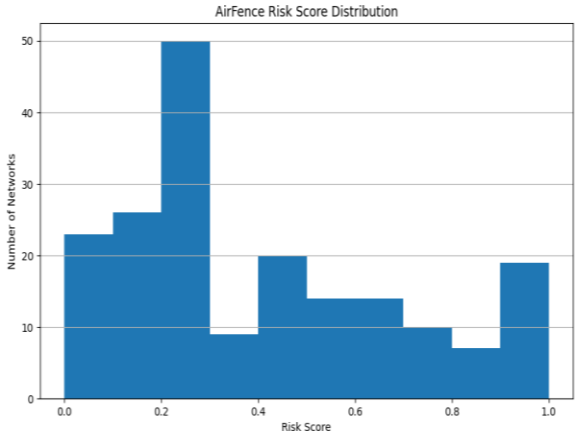


Figure 7: Risk Score Distribution

Figure 7 illustrates the distribution of generated risk scores across the collected wireless dataset. Networks with weak encryption, suspicious naming patterns, or abnormal signal behavior were associated with higher risk scores compared to trusted wireless networks.

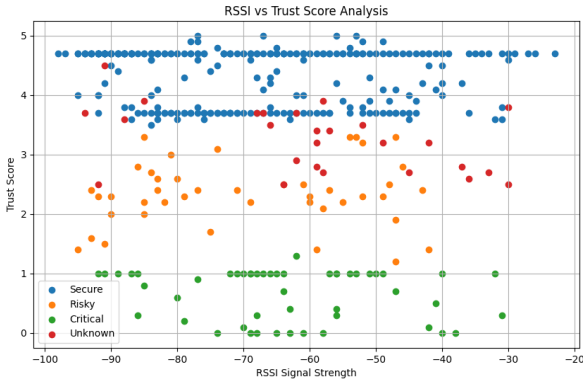


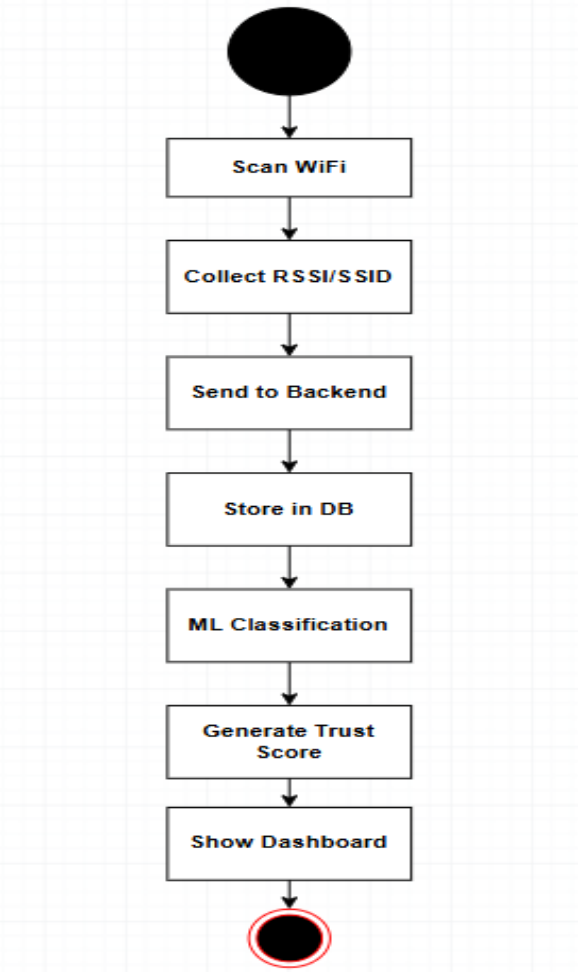
Figure 8: RSSI vs Trust Score Analysis

Figure 8 analyzes the relationship between RSSI values and generated trust scores within the AirFence framework. The results indicate that abnormal signal strength patterns combined with suspicious wireless behavior can contribute to lower trust scores and higher threat classification levels.

Workflow of AirFence

The AirFence framework follows a multi-stage workflow for wireless threat monitoring and Evil Twin detection. Initially, the ESP32 device continuously scans nearby Wi-Fi networks and captures wireless parameters such as SSID, RSSI, channel information, encryption type, and access point behavior. The collected data is transmitted to the Django backend server through API communication and stored inside the SQLite database.

The stored wireless observations are then processed using machine learning techniques to analyze suspicious patterns and classify networks into Secure, Risky, Critical, or Unknown categories. Based on the classification results, trust scores and risk scores are generated for each network. Finally, the AirFence dashboard visualizes the collected information, enabling administrators to monitor wireless threats and identify potentially malicious access points in real time.



Tools And Technologies Used

Experimental Setup

Component	Technology Used
Backend Framework	Django
Database	SQLite
Machine Learning Library	scikit-learn
Hardware	ESP32 + NRF24L01
Programming Languages	Python, C++
ESP32 IDE	Arduino IDE
Backend IDE	Visual Studio Code

VI. RESPONSIBLE IOT IMPLEMENTATION STRATEGIES

Data Privacy and Legal Compliance

AirFence collects some information from networks. This includes the names of the networks the strength of the signals what kind of encryption is used and the addresses of the devices that are sending out the signals. Airfence does not look at what people are actually saying to each other over these networks.

Secure Firmware Practices

The ESP32 firmware powering AirFence should follow secure coding practices. This includes: using HTTPS instead of HTTP for data transmission to the Django backend to prevent interception of scan data in transit, implementing firmware signing to prevent unauthorized firmware updates, disabling unused GPIO pins and communication interfaces to minimize attack surface, and regularly updating the firmware to patch known vulnerabilities in the ESP32 SDK. Arduino IDE supports secure over-the-air (OTA) updates, which should be configured with authentication.

Responsible Machine Learning Usage

The machine learning component of AirFence classifies networks into threat categories based on learned patterns. It is really important to remember that no machine learning model is perfect. Machine learning models can make mistakes. Sometimes machine learning models say a network is bad when it is actually a network. People in charge should look at what machine learning models say as a way to help them make decisions not as the answer. If a network is labeled as Critical people should check it by hand to

make sure it is really bad before they do anything to stop it or fix it. Machine learning models need to be updated with new information so they can keep telling the difference, between good and bad networks as things change. is also essential to maintain classification accuracy as wireless environments evolve.

User Awareness and Cybersecurity Best Practices
Technology by itself is not enough to stop Evil Twin attacks. We need to teach people about these things. When companies use AirFence they should also tell their people how to be safe. They can do this by telling users to check the name of the network before they connect to it. Users should not put in their codes; on networks they do not know. They should use a VPN when they are using Wi-Fi. If they see something weird, they should tell the IT people right away. AirFence's dashboard and alert system can be used to generate periodic security reports that support these awareness efforts.

VII. FUTURE SCOPE

AirFence is a working model that can detect threats and identify fake access points. There are ways to improve it.

- Cloud Integration:

Now the AirFence system works on a local server, which limits how many people can use it. If we move the system to a platform like AWS, Google Cloud or Azure we can monitor many AirFence devices from one place. We can also store a lot of data about activity over time which can help us predict when threats might happen.

- Deep Learning Enhancement:

The AirFence system uses a type of computer program called a Random Forest classifier to detect threats. We could make the system better by using advanced computer programs like deep learning models. These models can look at patterns in activity over time and detect threats that are hard to find.

- Mobile Alert Application:

We could create an app for iPhones and Android phones that would send alerts to network administrators when the AirFence system detects a threat. The app could show a map of where the threatsre let administrators add trusted networks to a safe list.

- GPS-Based Wireless Threat Mapping:

If we add GPS devices to the AirFence devices we can make a map of where the threatsre. This would help administrators find and fix threats quickly.

- Smart City and Large-Scale Deployment:

The AirFence system could be used in cities to monitor many public areas for wireless threats. This would help keep the citys computer systems and public transportation systems safe.

- Advanced Intrusion Detection Features:

We could make the AirFence system better at detecting types of wireless attacks not just Evil Twin attacks. We could also add a feature that captures packets of data without anyone noticing which would help us learn more about the threats. The AirFence system is a tool, for detecting wireless threats and identifying Evil Twin attacks. The AirFence system can be improved in ways to make it more useful and effective.

VIII. CONCLUSION

AirFence presents a lightweight and cost-effective wireless threat detection framework developed for identifying suspicious Wi-Fi networks and potential Evil Twin attacks in real time. The system combines IoT-based wireless monitoring, machine learning analysis, and backend visualization to improve wireless security monitoring in educational institutions and small-scale environments.
The proposed framework continuously scans nearby wireless networks and analyzes parameters such as SSID, RSSI, encryption type, channel information, and suspicious wireless behavior. The collected data is processed using machine learning techniques to classify networks into Secure, Risky, Critical, and Unknown categories. Experimental evaluation using the collected wireless dataset demonstrated strong classification performance and effective identification of potentially suspicious wireless activity.
AirFence was designed using affordable hardware and open-source technologies, making the framework suitable for environments where enterprise-level

wireless intrusion detection systems may not be practical due to cost or deployment complexity. The generated visualizations and threat analysis dashboards help administrators better understand wireless network behavior and monitor abnormal access point activity.

The framework also follows responsible wireless monitoring practices by focusing only on wireless metadata analysis rather than inspecting user communication content. This approach helps maintain privacy while still supporting effective wireless threat detection and security analysis.

Future improvements may include cloud-based deployment, deep learning integration, GPS-assisted wireless threat mapping, mobile alert systems, and larger-scale wireless monitoring support. Overall, AirFence demonstrates how IoT devices, backend analytics, and intelligent classification techniques can be combined to create an accessible and practical wireless security monitoring solution for modern network environments.

REFERENCES

[1] M. Agarwal, S. Biswas, and S. Nandi, "Efficient Rogue Access Point Detection Using Passive Traffic Sniffing and Machine Learning," *IEEE Trans. Network and Service Management*, vol. 16, no. 4, pp. 1576–1589, 2019.

[2] D. Han, H. Chen, and L. Zhao, "Machine Learning Based Rogue Access Point Classification for Enterprise Wireless Networks," in *Proc. IEEE Conf. Network and Information Systems Security (SAR-SSI)*, 2020, pp. 1–8.

[3] V. Kumkar, A. Tiwari, and P. Tiwari, "ESP32-Based IoT Network Monitoring and Wireless Security Analysis," *Int. J. Advanced Research in Computer Science and Software Engineering*, vol. 11, no. 3, pp. 45–52, 2021.

[4] M. Vanhoef and F. Piessens, "Practical Attacks Against WPA and WPA2 Networks and Their Implications for Wireless Security," *IEEE Security & Privacy*, vol. 16, no. 2, pp. 60–69, 2018.

[5] Y. Zhang, X. Chen, and W. Li, "Deep Learning-Based Wireless Anomaly Detection Using Convolutional Neural Networks," *IEEE Access*, vol. 9, pp. 12345–12356, 2021.

[6] A. Mittal and R. Sharma, "Trust Score Based Framework for Wireless Network Security Evaluation," *Int. J. Network Security & Its Applications*, vol. 14, no. 2, pp. 23–35, 2022.

[7] D. Rawat, S. Ghosh, and B. B. Sharma, "Wireless Intrusion Detection Challenges in IoT Environments: A Practical Survey," *J. Network and Computer Applications*, vol. 208, pp. 103–118, 2023.

[8] Espressif Systems, *ESP32 Technical Reference Manual*, Version 5.0, 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[9] Django Software Foundation, *Django Documentation Version 4.2*, 2023. [Online]. Available: https://docs.djangoproject.com/en/4.2/