National College of Ireland

Optimizing Rogue Access Point Detection with CART and Deep Learning Techniques

MSc Research Project
MSc. Cyber Security

Jay Chauhan
Student ID: 22137092

School of Computing
National College of Ireland

Supervisor: Prof. Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Jay Manishkumar Chauhan
Student ID:	22137092
Programme:	MSc. Cyber Security **Year:** 2023-2024
Module:	MSc Research Project
Supervisor:	Prof. Michael Pantridge
Submission Due Date:	14/12/2023
Project Title:	Optimizing Rogue Access Point Detection with CART and Deep Learning Techniques
Word Count:	7057 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jay Manishkumar Chauhan
Date:	13/12/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Rogue Access Point Detection with CART and Deep Learning Techniques

Jay Chauhan
22137092

Abstract

Wireless networks have become very common these days in various fields, such as big organizations, schools, and even public WiFi hotspots. These networks often handle the transfer of critical and sensitive information, making them targets for cybercriminals. The Rogue Access Point (RAP) used to subvert these networks are a very huge threat because this may cause great financial and personal loss due to theft of information. In addressing a pressing security challenge, this research presents comprehensive study of various models. The research evaluates the effectiveness of CART, FCNN and an Ensemble model (FCNN + XGBoost) using the AWID 3 dataset. The CART model showed remarkable accuracy and low false positives, making it highly suitable for real-world application. The FCNN provided insight into further refinement for this study, while the Ensemble model delivered a balanced performance in precision and recall. This study contributes significantly in the field of network security, offering advanced methodologies for Rogue Access Point detection and enhancing wireless network security against emerging threats.

Keywords: WiFi hotspots, Rogue Access Point, Machine learning, CART, FCNN, Ensemble

1 Introduction

In this technological era, telecommunications have been closely linked to other aspects of the life. Wireless fidelity, also known as WiFi technology is one the innovative approaches and widely used in day-to-day life. WiFi allows users to access the internet from anywhere without using any cable. WiFi technology has advanced to offer faster wireless access to the internet applications and data via radio network. Since anyone can access WiFi from anywhere within the service area, this is useful to everyone. Moreover, WiFi can be easily integrated with any mobile devices. Even though WiFi makes things easier, users are still at risk because attacker can hide between WiFi users and the access point. The users send their information to attacker instead of the access point. Attacker can access the user's information including sensitive data, mails, credit cards, and credentials too.

Rogue access point also known (RAP) refers to a wireless Access Point (AP) that has been installed on a secure network without obtaining explicit authorization from a network administrator. Alternatively, it may have been intentionally setup by a hacker to enable Man-In-The-Middle attacks, where communications between devices on the network is intercepted (Arisandia, D. at el.,2022).

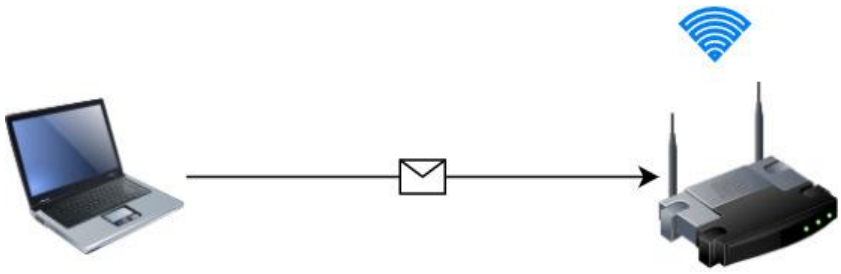


Figure 1. Device connected to authorized access point

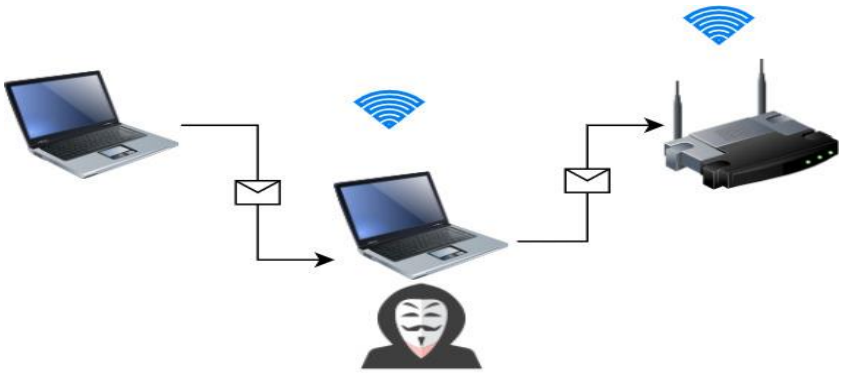


Figure 2. Device connected to an unauthorized access point

The increasing complexity of the cyber threats in the digital age has made the detection of rouge access point crucial part in the field of cybersecurity. As the presence of inappropriate access point will lead to many other kinds of attack in the wireless network. These attacks can result into data and financial lose, which shows that the detection of such access point is the need of the hour. The system proposed here will employ the machine learning and deep learning algorithm which will effectively detect the Rogue Access Points (RAP).

The primary question of the research revolves around: "**What are the capabilities of the Classification And Regression Trees (CART), deep learning and Ensemble model in detecting rogue access points (APs) in wireless networks. How well the models are performing compared to each other and why?**"

The aim of the research is to make substantial contribution in network security. By proposing and evaluating CART, FCNN and Ensemble model (FCNN + XGBoost), this research offers a novel approach to rogue AP detection. The use of AWID 3 dataset, which provides the diverse range of attack scenarios, ensures that the proposed model is robust and relevant to modern day network infrastructure.

The structure of this reports is as follows:
Following this introduction, Section 2: presents a review of related work. Section 3: Outlines the methodology of the proposed solution and details of models. Section 4: Discusses the prerequisites, architecture of the model, and functionality of the models. Section 5: Delves into actual implementation of proposed solution, data prepressing, feature extraction, model-development and web application development. Section 6: Evaluation section talks about the results of each model, and comparison of algorithms with each other. Section 7: concludes the report with summary of the key findings and suggestions for future work.

2 Related Work

Previous research has examined that can detect the rouge access point in wireless network. The insights obtained from the relevant literature will help evaluate the system's outcomes developed in this study.

2.1 Rouge Access point detection

In the literature on rogue access point detection, a comprehensive review of studies offers insights into the existing detection systems with their methodologies and limitations.

Kim, D. et al. (2018) utilize the Round-Trip Time values with machine leaning algorithms such as and compare algorithms like Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Multilayer Perceptron MLP. These algorithms demonstrate the practical usage in identifying the rouge access points. However, the main drawback of this study is the limited focus on RTT values. Which may not capture the different types of unauthorized access point features. This focus potentially restricts the detection scope. The study by Shetty, S. et al. (2007) use a novel approach to identify the inappropriate access point by analysing the network traffic pattern automatically. The main advantage of this approach is, its flexibility to adapt to different environments, as this approach does not rely on any wireless technology. However, this approach heavily relies on predefined traffic patterns and thresholds that can lead to high false positives and false negatives in dynamic network conditions. The other study by Wu, W. et al. (2018) focuses on identifying rogue access point using a unique method based on RSS (Received Signal Strength). This approach effectively manages the missing values by using pre-processing techniques and ensures robustness against the noise by implementing k-medoid clustering. Although the method is innovative, its heavy dependence on RSS data may not be sufficient for complicated datasets. Moreover, the study's performance measurements focus on the detection rate, which may not be the most precise measure in all situations. In this paper, author Vaidya, T. S. (2023) utilizes the traditional machine learning algorithm such as KNN, SVM, and Random Forest. The AWID 2 dataset is used in this study. The proposed solution successfully identifies the inappropriate access point. However, the proposed solution cannot identify the complex traffic patterns which are crucial for network security.

In contrast, the proposed approach employing CART, deep learning model FCNN, and Ensemble model provides a more comprehensive analysis, capable of identifying intricate patterns in large datasets. FCNN's capability to detect complex traffic patterns, CART's ability to capture non-linear relationships, and the Ensemble model address the limitation of reliance on specific patterns, ensuring effective detection. Moreover, the proposed solution offers a broader and more effective solution by not relying on RTT data, which provides a more accurate and comprehensive rogue AP detection system.

2.2 Network Intrusion Detection Using Deep Learning and Machine Learning

The detection of unauthorized access points in wireless is similar to the intrusion detection system (Khan et al., 2020) and existing studies can be reviewed. Shone, N. et al. (2018) utilize the deep learning algorithm to classify the network behaviours and it also achieves an accuracy of 99% which might be the result of potential overfitting. Also, the model trained on the NSL-KDD dataset which is quite old and does not represent the dynamics of current network threats. This potentially can limit its real-world applicability. Author Wei, P. et al. (2019) in their study presents a novel approach in the area of intrusion detection. This approach optimizes the Deep Belief Networks by using Particle Swarm Optimization (PSO) and genetic algorithm (GA). This developed approach outperforms the other DBN optimization techniques in terms of accuracy. However, the complexity of this optimization increases the performance overhead which might pose challenges for real-time applications. Alsughayyir, B. et al. (2019) introduces the novel approach NDAE (Nonsymmetric Deep Auto Encoder) for feature learning in intrusion detection systems. This methodology increases the accuracy and reduces the need for human intervention in intrusion detection system. While this approach is effective, but relies on older datasets like KDD cup'99 and NSL-KDD. This reliance on the older dataset limits the effectiveness of the approach against the newer and more sophisticated types of network intrusion.

The proposed approach contrasts with previous studies by especially targeting the identification of rogue access points. The approach utilizes both CART and advanced deep learning algorithms, specifically FCNN and XGBoost. This approach effectively identifies the rogue access points by using AWID 3 (Aegean WiFi Intrusion Dataset). This dataset includes a variety of modern attack scenarios which shows that the proposed approach not only overcomes the limitation of reliance on specific, outdated datasets but also offers the complete solution for identifying Rogue Access Points (RAP) in modern network environments.

Several studies have shown significant importance in the field of network intrusion detection by using machine learning model. However, each study has their own methodology, advantages and limitations. Author Sumaiya, I. et al. (2018) utilize a custom dataset of 1130 instances to classify network traffic using Naïve bayes, SVM, Random Forest and KNN algorithms. Although the accuracy of this study shows high accuracy, its limitation is the comparatively small dataset. Also, high accuracy shows the concerns about overfitting. Babu, B. S. et al. (2023) utilize PCA feature selection technique on the NSL-KDD dataset to identify important features and employs Random Forest, Decision Tree, and SVM machine learning algorithm. The high accuracy of 97.38% showing the potential algorithmic biases. Moreover, the NSL-KDD dataset is quite old and does not represent the characteristics of the modern network intrusions. Author Hossain, Z. et al. (2021) in their study utilize UNSW-NB15 dataset which shows comparatively improved accuracy with the classifiers such as Decision Tree, Random Forest, SVM, and Nave Bayes. Although this study shows better performance, but it limits with binary classification only and also does not uses any feature selection approaches which raises the concerns about the potential false accuracy. Taher, K.et al. (2019) introduce the novel approach by combining the SVM and ANN algorithm to enhance the performance of the intrusion detection system. This method uses NSL-KDD dataset to train the novel machine learning model and achieves the high accuracy of 94.02%. The complexity of ANN model and the old dataset limits the performance of the approach. Author Dsouza, A. et al. (2022) in their research focus on real-time intrusion detection using various algorithm trained such as Nave Bayes, SVM and Random Forest on NSL-KDD dataset. The dataset lacks comprehensive features which limits this approach.

In contrast, the proposed solution advances the field by overcoming the shortcoming described above and offers a more robust approach. The solution implements the CART model with deep learning techniques, and ensemble model combining FCNN deep learning model with XGBoost model. Comprehensive AWID 3 dataset is used for this study. The approach uses sophisticated feature extraction technique with ANOVA. This technique ensures a more balanced and nuanced classification through the suggested machine learning models. This study also addresses the overfitting issues which commonly seen in the singular algorithm approaches. All in all, the proposed method's focus on rogue AP detection offers a more specialized in-depth analysis.

2.3 Summary Table

Research Paper	Approach	Dataset	Algorithms	Limitations	Upsides
Kim, D. et al. (2018)	Using RTT value dataset to detect authorized and unauthorized APs, analyzed with ML algorithms.	RTT dataset	SVM, Decision Tree, KNN, MLP	Narrow focus on RTT values, may not represent all unauthorized AP characteristics.	Effective comparison of ML techniques, KNN shows highest accuracy
Shetty, S. et al. (2007)	Automated rogue AP detection in heterogeneous networks via traffic analysis at network edge.	Heterogeneous network comprised of wired and wireless subnets.	Traffic analysis	Reliance on specific traffic patterns and thresholds, risk of false positives/negatives.	Automated, adaptable to various networks, independent of specific wireless technologies
Wu, W. et al. (2018)	RSS-based practical rogue AP detection considering missing RSS values in collected vectors.	Practical environment with RSS vectors.	k-medoid algorithm for clustering analysis on RSS vectors	Focus on RSS data, challenges with complex datasets.	Effective in reducing false alarm rate while ensuring high detection rate.
Vaidya, T. S. (2023)	System for identifying unauthorized APs in wireless networks	AWID 2 dataset	KNN, SVM, Random Forest, Genetic Algorithm	Not identifying the type of unauthorized AP attack, only detects the simple pattern by using	Random forest shows the highest accuracy

	using ML classifiers.			ML, Potential overfitting	
Baby, B. S. et al. (2023)	Application of machine learning techniques to enhance NIDS accuracy	NSL-KDD dataset	Random Forest, Decision Tree, SVM	Dataset specificity, algorithmic limitations, evaluation constraints	High accuracy (97.38%) and low error rate (0.25%)
Hossain, Z. et al. (2021)	Examining effectiveness of machine learning classifiers in network intrusions	UNSW-NB15 intrusion detection dataset	Decision Tree, Random Forest, SVM, Naïve Bayes	Binary classification, lack of feature selection, dataset specificity	Improved performance with popular classifiers
Dsouza, A. et al. (2022)	Developing a real-time IDS using machine learning algorithms	NSL-KDD dataset	Naïve Bayes, SVM, Random Forest, Decision Tree, Logistic Regression	Dataset limitations, real-time processing challenges, model generalization	Feasible approach for real-time detection with high accuracy
Shone, N. et al. (2018)	Novel deep learning technique for intrusion detection using NDAE and Random Forest.	KDD Cup '99 and NSL-KDD datasets.	Non-symmetric Deep Auto-Encoder (NDAE) and Random Forest.	Potential overfitting, old dataset	Improved detection accuracy, promising for modern NIDS.
Wei, P. et al. (2019)	Deep Belief Network (DBN) optimization for intrusion detection classification model.	NSL-KDD	Deep Belief Network (DBN) with optimization algorithms.	Performance overhead, real-life application challenges	Promising classification accuracy in binary classification.
Alsughayyir, B. et al. (2019)	Deep Learning approach for Intrusion Detection System using Auto-encoders.	NSL-KDD Dataset	Deep Auto-encoders for feature learning and classification.	Potential reduction in accuracy when tested on unseen data.	High accuracy in training and testing.

The main focus of the research is to develop an advanced and efficient method for detecting rogue access points in wireless networks. The research identifies the limitations of traditional machine learning models. The research introduces the CART model with sophisticated deep learning model FCNN and Ensemble model. By utilizing these 3 algorithms, the research addresses the key challenges observed in previous studies such as over reliance on specific data features like RTT values, limitations in handling complex data, and the need for robustness against the noise and diverse attack scenario. Moreover, the utilization of AWID 3 dataset, incorporates modern attack scenarios, which enables our model to be more adaptable and effective against contemporary network threats.

3 Research Methodology

The research methodology is emphasizing on the properly planned experimental design which uses the AWID 3 dataset to assess the machine learning models which detects the inappropriate access points in wireless network.

Dataset Acquisition:
The AWID 3 dataset was chosen because of its progression in wireless network security and research area. The goal of this dataset was to capture a wide range of attacks which occur in an IEEE 802.1X EAP environment. These range of attacks includes Deauth, Disas, (Re)Assoc, Rogue AP, Krack, SSH, Botnet, Evil_Twin and SSDP attack. To mimic the real-life corporate network, consist of various kinds of client devices and servers, this data was produced in controlled environment. This dataset is an excellent resource for training and evaluating the machine learning models because dataset accurately mimics the actual traffic data and attack patterns.

Dataset Pre-processing:
After the data acquisition, the dataset went under pre-processing phase to get the clear idea about the content and the structure of the dataset. In pre-processing, first cleaning performed in which, the null and missing values were removed. After dataset clean-up, label encoding was performed on the target feature in order to make the data numerical.

Feature Extraction & Selection:
In order to enhance the dataset for the model training, ANOVA test was carried out in order to identify the important features from the dataset, which would impact the target variable. This methodology is based on the assumption that not all features have the equal significance on the predictive ability of the model. The ANOVA test played a crucial role in determining the features which has the highest significant scores. The features with less significant scores were removed from the dataset so that only selected group of the most relevant features will be used

for model training

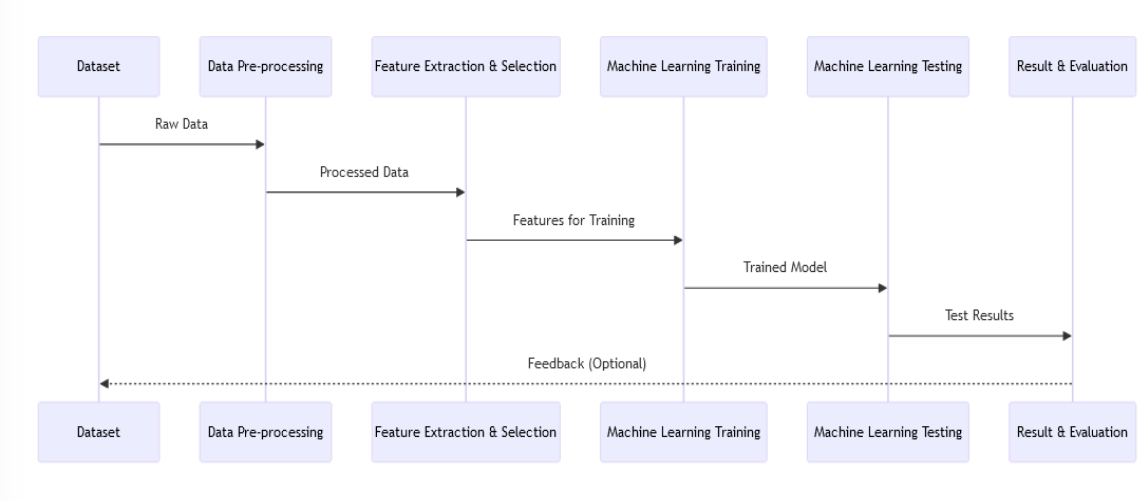


Figure 3: Research Methodology

Model Development:

Machine Learning Model: The study implemented a comprehensive strategy by using different models. Each model has different capabilities in data processing and analysis.

1. CART: The CART (Classification And Regression Tree) model was chosen because of its exceptional interpretability (Thapa, N. et al., 2020), which is really needed for learning the decision-making procedures. CART is comparatively cheap in computational cost which makes it the best choice where speedy response time is essential.

2. FCNN: The deep learning model FCNN (Fully Connected Neural Network), was selected for its capacity to identify the complicated connections and patterns within the dataset which may not be seen by other simpler models (Chen, C. et al., 2022). Its layered structure helps to interpret the nuanced network traffic.

3. Ensemble model (FCNN + XGBoost): The combined model was created by merging the FCNN and XGBoost model to take advantage of both machine learning and deep learning approach. XGBoost is known for the high performance and accuracy in analyzing the structured data. On the other hand, FCNN shows excellent performance in detecting the detailed, non-linear trends in data, which makes it particularly suitable for the dataset.

Training Phase: Each model went through training stage, in which a specific subset of the dataset which is also known as training data, was used to teach the models in identifying and categorized the network traffic. The training data comprises of both regular and malicious network activity, which helps the models with important information that needed to detect the distinctive characteristics.

Testing Phase: In order to assess the performance of the model on unseen data, this phase is critical. The dataset was partitioned into training and testing data, this testing data is exclusively reserved for the testing purpose. Each model, was tested using the testing dataset in order to prevent the data leakage from the training phase and ensure proper evaluation of the model. The focus during the testing stage was on just the accuracy of the models, which gives the quick and easy measure of the model performance.

Evaluation:
The evaluation phase is a in detail analysis of the models' performance using different metrics to understand their strengths and limitations.

- Cross-validation: K-fold cross validation was utilized in development phase, but the results are analysed in this phase. This method confirms the accuracy of the model and also detects any possible differences in the performance among the different sets of data.
- Confusion Matrix: The confusion matrix is tested in the evaluation phase to get the insights into the types of errors that model has made. It helps to find the difference between the false positives and false negatives. This gives a more complete picture of how model is performing than just how accurate it is.
- Precision and Recall: The confusion matrix is used to find the precision and recall. Precision is the percentage of correct positive prediction and real positive prediction percentage is known as recall. These numbers are used to evaluate the model's quality of prediction.
- F1 Score: F1 score, which also known as harmonic mean of precision and recall is very important metric. F1 score plays crucial role when dealing with imbalanced class, as it provides the balance between the both precision and recall measurements.
- Evaluation Result: The results from the confusion matrix and cross validation are used to compare the models against each other.

Final Model Selection:
The model which has the best overall performance across all the metrics, mainly confusion matrix analysis, is chosen for real world web application. This application will be used to detect the inappropriate access point.

The above-described methodology combines the experimental design with secondary data analysis, which helps to answer the research question of balancing new model implementation against the accuracy of evolving detection system for wireless access points.

4 Design Specification

4.1 Architectural Foundations

The research uses the robust machine learning architecture which are specifically tailor for the sophisticated categorization of the network data as either normal or malicious.

CART Model Architecture:

- Construction: The CART model works using the binary tree structure, in which the data is divided into nodes that are based on the features which segregates the attack classes from the normal behavior. Each and every node in the tree acts as a decision point, which ultimately leading to leaves that signify the ultimate classification decision.

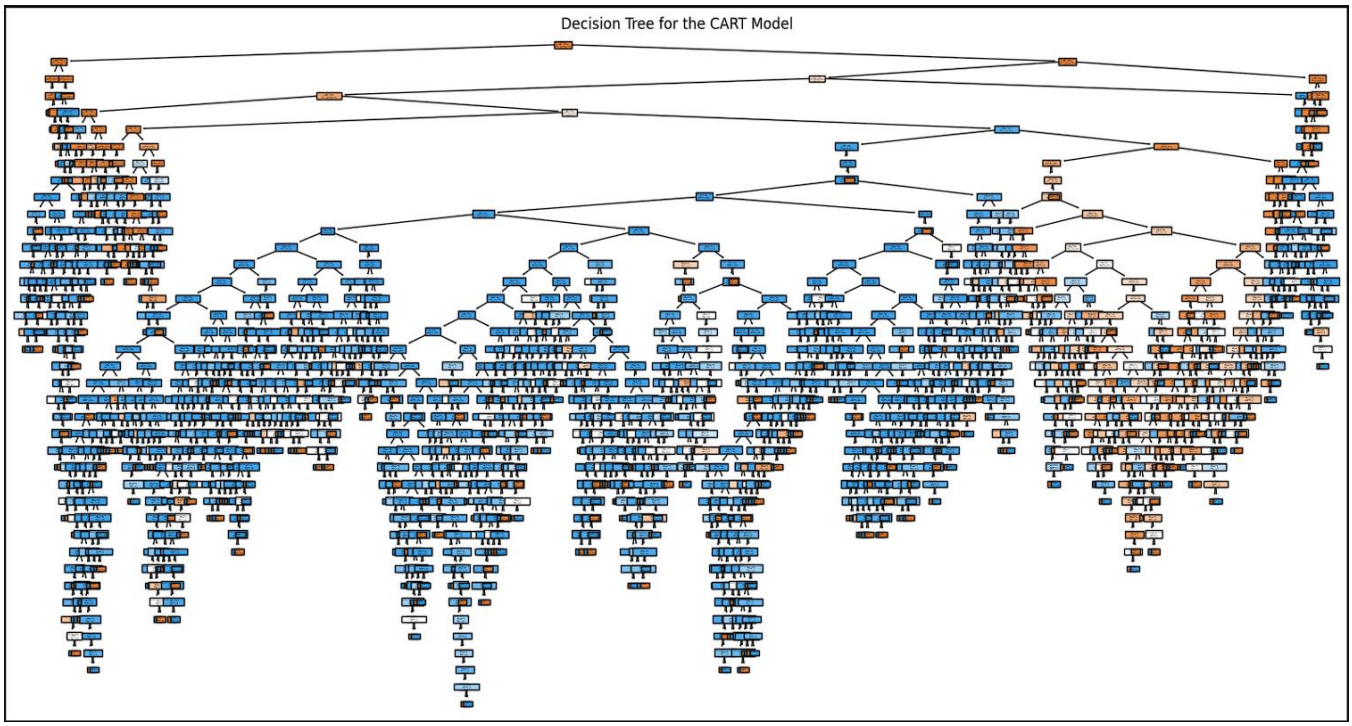


Figure 4: Tree Architecture of CART algorithm

- Pruning Strategy: In order to prevent from overfitting, the tree is trimmed by eliminating the branches which have low impact on the prediction accuracy. This helps to enhance the model's capacity to generalize.

FCNN Architecture:

- Layer Configuration: The FCNN architecture is made up of an input layer which accept specified number of features, Two hidden layer with 128 neurons each, and an output layer with a single neuron. The hidden layers use the Rectified Linear Unit (ReLU) activation function in order to include non-linearity, which is needed to model the complex relationship.
- Output Activation: The output layer employs a sigmoid activation function, which is very useful for binary classification applications, such as identifying between the normal network traffic and network intrusions.
- Optimization and Loss: The network is mase up using the Adam optimizer and binary cross-entropy loss function, which is useful for binary classification problem.

Ensemble Model Architecture (Parallel Ensemble):

- XGBoost Independence: The XGBoost model was train independently on the same dataset. The model's ability to sequentially focus on and improve upon the misclassification made it the perfect choice.
- Meta Model Evaluation: In parallel to the FCNN, the prediction of the XGBoost model were combined to evaluate the meta model's performance. This ensured the strength of both models were used effectively.
- Performance synthesis of Meta model: rather than training the meta-learner, the parallel ensemble technique focused on synthesizing the predictions of the independently trained FCNN and XGBoost models. The performance of the ensemble's model was evaluated based on the consensus between the two predictive models.

4.2 Technical and Operational Requirements

- *Data Integrity*: The dataset has gone through comprehensive preprocessing which ensured the data integrity. The AWID 3 dataset was chosen for because it offers the extensive coverage and wide range of features.
- *Computational Requiremen*t: The data pre-processing and feature extraction performed on the windows laptop with Ryzen 7 processor and 16 GB RAM. The machine and deep learning model revealed the necessity for more significant computational resources, so using Google Cloud Platform instance with 16 cores and 64 GB RAM greatly improved the training process. This approach made the model more precise and training process efficient.
- *Software and Libraries*: Python was chosen for its wide range library support. Libraries such as Pandas for its exceptional data handling capabilities, Numpy for its comprehensive support for large, multidimensional arrays and matrices. Matplotlib and Seaborn were used for visualizing the dataset and showing the results of the models. To train models, library such as Scikit-learn for developing the CART model, TensorFlow and Keras to build and train the FCNN model and lastly XGBoost library were used to train the model.

4.3 Functionality of Proposed Models:

- *CART Model*: The main goal of the CART model is to deliver the fast and easily interpretable results. The decision tree's structure provides clear insights for the decision making which very useful for the research problem.
- *FCNN Model*: The design of the FCNN model, fully connected layers allow it to identify the deep relationships within the network traffic data.
- *Ensemble Model*: This model utilizes the parallel integration approach and improve accuracy by maximizing on the unique capabilities of each model. This synthesis is carefully combined to cover wider range of intrusion detection scenarios which ensures the comprehensive coverage.

The design specification outlines the exact technical as well as operational requirements for the models, addressing the challenges of managing the large dataset, and the flexible computational framework. The proposed models were described with their unique characteristics, and combined contribution to the detection system.

5 Implementation

The implementation stage of the study involves the practical execution of the theoretical and design elements that were previously described. This also entails the final model development and deployment of the model into the web application for real-time Rogue access point detection.

5.1 Tools and Languages:

- *Programming language*: Python Language was used throughout the study because of its extensive support for data science and machine learning.
- *Libraries and Frameworks:*
    - Pandas and Numpy: For data manipulation and numerical computations.
    - Scikit-learn: For implementing the CART model and data preprocessing tools.

    - TensorFlow and Keras: These tools were used to train the deep learning FCNN model.
    - XGBoost: For developing the XGBoost model for ensemble approach.
    - Matplotlib and Seaborn: For visualizing the data and the results.
- *Computational Resources*: The project was implemented on powerful computing environment GCP (Google Cloud Platform). The instance was created with 16 core CPU and 64 GB RAM. This allowed the efficient processing and training of the models, especially intensive deep learning model.

5.2 Dataset Collection and Preprocessing:

- *Dataset Selection*: The AWID 3 dataset was selected for this research because of its comprehensive array of network traffic data, this dataset is particularly relevant to contemporary network security challenges (AWID-3, Aegean.gr., 2021). It has a variety of network behaviors, capturing both standard operational data and multiple attack scenario, which provides the realistic environment for testing and training the detection model.
- *Dataset Curation*: From the range of the attack files, focus was made on 4 specific types of network attacks. These attacks are Deauth, Evil twin, ReAssoc, and Rogue AP attacks. These attacks were chosen because of their relevance to the study problem as these attacks also plays roles to create the inappropriate access point. Other attack data were removed from the dataset, so that research critically concentrate on the most impactful security issues.
- *Data Cleanup and Merging*: The AWID 3 dataset undergone the thorough cleanup. For each selected attack types, corresponding CSV files were cleaned of any irrelevant and null values. After cleanup, the data pertaining to each attack type was merged into a single CSV file to make the training process simpler and easier. This merged dataset formed the backbone of the subsequent model training and analysis. After cleanup and merging the size of the CSV file was 1.91 GB with 9,072,280 rows and 34 columns.
- *Label Encoding*: A crucial step in preprocessing after the merger, was label encoding. The network traffic data were encoded into binary format. The Normal traffic was labeled as ‘0’, on the other hand, attack data was denoted as ‘1’. This binary classification scheme played an important role which allowed the machine learning models to effectively distinguish between the normal and malicious network traffic.

5.3 Feature Selection & Reduction:

- *ANOVA Test*: The ANOVA (Analysis of Variance) test played the significant role in the feature selection process. (Surendiran, B. et al., 2011). This method helped to identify the features that has the higher variance in the dataset. The features with the most higher variance value have the impact on the target variable. The ‘Label’ column indicates the behavior of the traffic (Normal or Attack).
- *Feature Reduction*: The results from the ANOVA test led to the identification of 12 key features that has impact on the predictability of the traffic data. The dataset was then reduced to the 12 features and 1 label variable. This approach allowed that the machine learning models get trained only on the important features and enhancing the accuracy and the efficiency of the model to predict the rogue access points. After dropping the irrelevant features from the dataset, the size of dataset reduced to 700 MB with 9,072,280 rows and 13 columns.

5.4 Model Development and Training:

The model development and training of the models were executed systematically, each model went through rigorous training and evaluation to ensure the optimum performance.

5.4.1 CART Model:

- Development and Training: The CART model was trained on the GCP instance to ensure the smooth training procedure. The dataset with 9,072,280 rows and 13 columns were divided into 80% training set and 20% testing set. The 12 features were trained to detect the label variable.
- Accuracy Concern: The model gets trained in very short time and with exceptionally high accuracy rate of 99.98%. However, the high accuracy of the model also raised the concern of the possibility of the overfitting, which is very common issue in machine learning models, where the model learns the training data so well but cannot generalize the new data.
- Validation Technique (Cross-validation): In order to find the reliability of the model, cross-validation technique was also used. This process divided the data into various subsets, the model was trained on some sets and tested on other sets. This provided the comprehensive evaluation of the performance across different data subsets.
- Validation Technique (Confusion Matrix): The confusion matrix was also performed. This test helped to get the insights into the capabilities of the model. It also provided the details of the precise classification of various types of network traffic (Normal & Malicious).

5.4.2 FCNN Model:

- Initial Training and Analysis: The Fully Connected Neural Network was trained after the training the CART model. The FCNN model has 3 layers. The first input layer is designed to receive data with 12 features. The other layers with 128 neurons and ‘ReLu’ activation functions, which are standard for hidden layers as they implement non-linearity in the model to make it learn more complex pattern. The final layer is a single neuron with a ‘sigmoid’ activation function, which makes it suitable for binary classification. At the first, this model showcased the high accuracy of 98%. With the help of confusion matrix, it was discovered that there was a tendency to mostly classify the normal traffic data. This suggested that the model is having issue with the imbalanced dataset, where the one class (normal traffic) was over-represented compared to the other class.

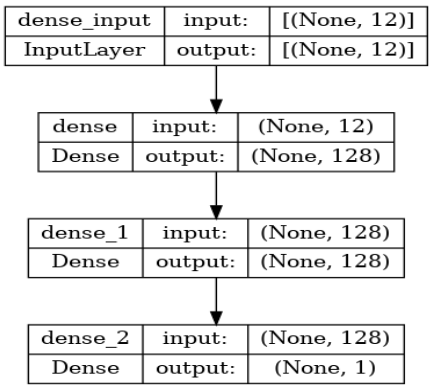


Figure 5. Model architecture of FCNN

- Addressing imbalance: The Synthetic Minority Over-Sampling Technique (SMOTE) was used to deal with the imbalanced dataset. SMOTE allowed the synthetic data generation to balance the under-represented class of the dataset. This enabled the FCNN model to learn from the uniform dataset. The accuracy reduced to 68% after training the model on the balanced dataset.

5.4.3 Ensemble Model (FCNN + XGBoost):

- Model Combination: The parallel ensemble model combined the strength of both model FCNN and XGBoost. This model was designed to leverage the strength of both FCNN model to understand the complex pattern and XGBoost for its precision.
- Training Approach: The ensemble model was developed by training the both model FCNN and XGBoost independently and subsequently merging them for their predictions. The main goal was to build the more precise and resilient detection model.

All in all, the implementation of each model entailed a detailed process of the development, training and evaluation, The use of cross-validation, confusion matrix and techniques like SMOTE ensured that each model was not only accurate but also reliable but also capable of effectively identifying the intrusion.

5.5 Web Application Development and Deployment:

After the model development and validation, the next phase focused on implementing the model into a practical, user-friendly format. This led to the development and deployment of the web application. This application was designed to utilized the CART model due to its high accuracy and efficiency.

- Web Application Design: The application was built to provide an interface for user to interact with CART model. It was designed to allow users to input network data and receive the real time classification for potential network intrusion
- Development Tools: The application was built using a combination of the front-end and back-end technologies. HTML, CSS and JavaScript were used to develop the user interface design of the application. For back-end, Python with flask framework was used to integrate the CART model into the application.
- Model Integration: The CART model was embedded into the application back-end, where it processed the input data and provided the classification results. This integration allowed users to use model's capabilities.
- Deployment on GCP: The final application was deployed on the GCP application. This platform was chosen for its scalability, reliability and ability to handle the computational demand. The app engine service of GCP was utilized to deploy the application. The deployment allowed the access of the application from various location.

5.6 Outputs:

- Transformed Data: The preprocessed and feature-selected dataset, which is ready for model training and testing.
- Developed Models: The finalized CART, FCNN and Ensemble models. Each model was tailored to accurately classify the rogue access point.

- Performance Metrics: Detailed reports on each model's performance, including its accuracy, precision, recall and F1 scores. The confusion matrices were also generated for each model to evaluate the effectiveness.
- Web application: The user-friendly web application was also developed with CART model. This allowed normal user to check the potential rouge access point.

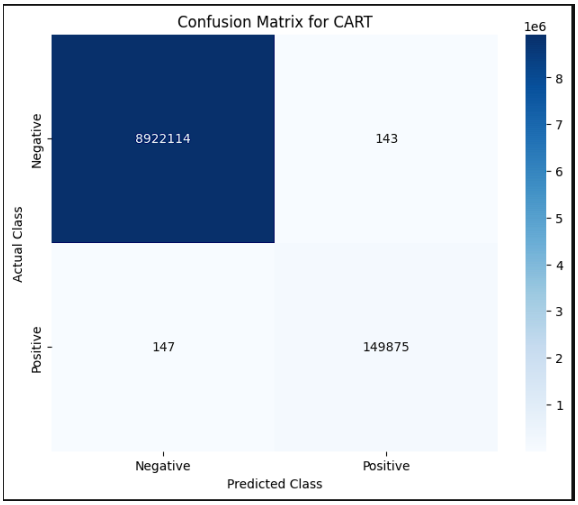
In summary, the implementation of the proposed solution included systematic approach to data preparation, model development, performance evaluation and web-application development using the range of tools and technologies which resulted in the accuracy and efficiency of the Rouge access point detection system.

6 Evaluation

The evaluation section in depth analyses the performance of each model. This analysis very critical to understand the effectiveness of these models in detecting rogue access points, and their implication from both an academic and practical perspective.

6.1 Case study 1: CART Model Evaluation:

- Objective: To assess the effectiveness of CART model in accurately identifying the network traffic, with focus of detecting false positives and negatives.
- Experiments Performed:
    - Performance Metrics: Accuracy, precision, Recall and F1-score were measured to evaluate the overall performance of the model
    - Confusion Matrix analysis: To gain insight into the model's ability to correctly classify traffic and identify the types and rates of classification errors.
- Performance Metrics:
    - Accuracy: 99.97%
    - Precision: 98.44%
    - Recall: 98.56%
    - F1-Score: 98.38%
- Confusion Matrix:



True Negatives (TN): 8,922,114
False Positives (FP): 143
False Negatives (FN): 147
True Positives (TP): 149,875

Figure 6. Confusion Matrix for CART

- Analysis: The CART model has outstanding accuracy, showing its robustness in detecting network traffic. The model's excellent precision and recall shows that model is identifying both normal and attack cases, while minimizing the occurrence of false positives and false negatives. The slight variance in precision and recall suggests the balanced approach for both types of classification errors.

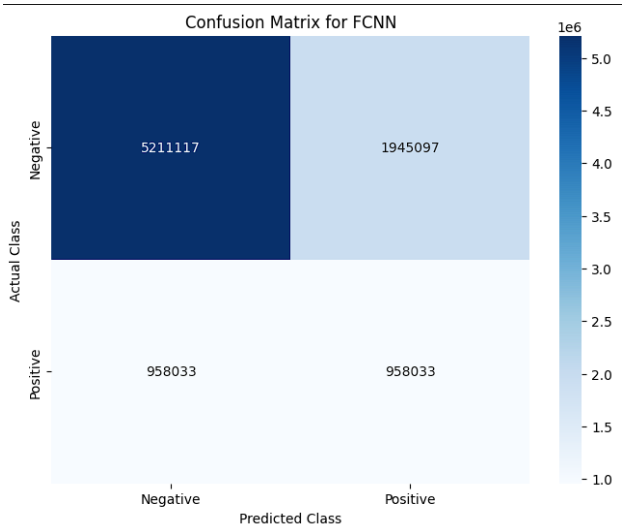
6.2 Case study 2: FCNN Model Evaluation:

- Objective: Evaluate the FCNN deep learning model's ability to classify the complex pattern of the network traffic and classify into normal or malicious category.

- Experiments performed:
    - Performance metrics: Focusing on accuracy, precision, Recall and F1-score to assess the balanced performance of the model.
    - Confusion Matrix: To understand how well the model can distinguish between normal data and malicious traffic, specifically after applying the SMOTE.

- Performance Metrics:
    - Accuracy: 68.21%
    - Precision: 33.98%
    - Recall: 50.58%
    - F1-Score: 40.86%

- Confusion Matrix:



True Negatives (TN): 5,211,117
False Positives (FP): 1,945,097
False Negatives (FN): 958,033
True Positives (TP): 958,033

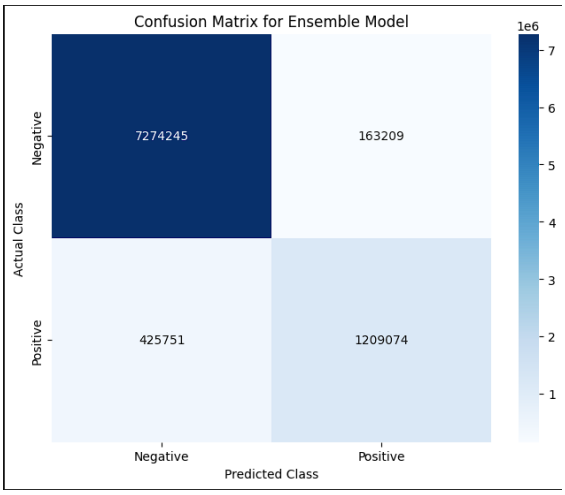
Figure 7. Confusion Matrix for FCNN

6.3 Case study 3: Ensemble Model Evaluation:

- Objective: To evaluate the efficacy of the Ensemble model in achieving a well-balanced and precise classification of the network traffic.
- Experiments Performed:

- Comparative Performance Metrics: Evaluation accuracy, precision, recall and F1-score and comparing it with the individual FCNN and CART model.
- Confusion Matrix: To investigate the efficacy of t=combined model in minimizing both false positives and false negatives.

- Performance Metrics:
  - Accuracy: 81.98%
  - Precision: 88.11%
  - Recall: 73.96%
  - F1-Score: 80.41%
- Confusion Matrix:



True Negatives (TN): 7,274,245
False Positives (FP): 163,209
False Negatives (FN): 425,751
True Positives (TP): 1,209,074

Figure 8. Confusion Matrix for Ensemble

- Analysis: The Ensemble model shows enhanced accuracy compared to the individual model FCNN. The excellent precision score indicates the substantial decrease in the false positives, while recall demonstrates a remarkable capability to accurately identify real attacks. The F1-score, significantly raised which demonstrates the improved accuracy of the model in categorizing the network traffic.

6.4 Discussion

The machine learning models are assessed using the bar graph, which provides a comparative analysis of their performance across different measures. The discuss will be focused on the interpretation of these findings and their implication for the practical implementation of these models in detection system

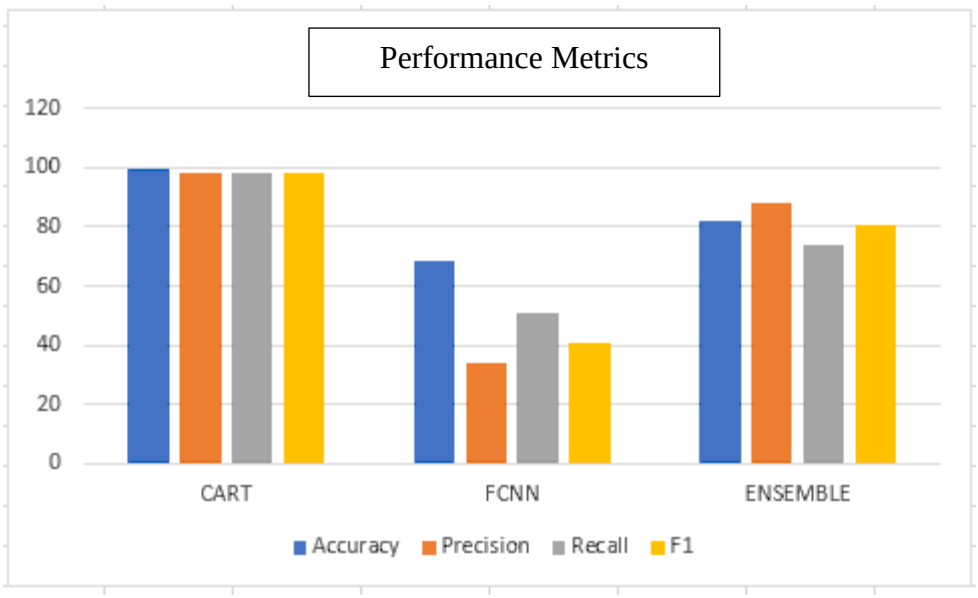


Figure 9. Performance Metrics

CART Model Performance:

- The CART model shows the near-perfect scores in all measures outperforming the other models in precision and F1 score. This suggests that the CART model is highly effective in identifying both normal and attack classes with few errors.
- The outstanding performance across all metrics also raise the concerns about overfitting. However, the consistent score in all metrics could also indicate that the model is working very well.
- Given these results, the CART model stands out as highly reliable choice for deployment in environments where high precision is required to minimize the false positives.

FCNN Model Performance:

- The FCNN model's performance is the lowest among the three. The accuracy score with 68% and even lower precision, recall and F1 score indicates that the model struggles to balance the detection of positive and negative classes.
- The moderate performance of the FCNN model points to a need for further tune up in the model architecture, additional feature engineering, and further refinement.

Ensemble Model Performance:

- The Ensemble model shows the significant enhancement compared to the FCNN model, particularly in terms of precision and F1 score. This suggests that ensemble model offer more reliable performance in detection of true positives and avoiding false positives.
- While the ensemble model does not reach the accuracy levels of the CART model, but its balanced metrics may offer reliable performance in varied scenario, where CART model might be too sensitive.
- The enhanced performance of ensemble over the FCNN model shows that combining models can leverage the strength of individual models and achieve overall good performance.

The bar graph demonstrates that the CART model consistently achieves the greatest metrics. it's the balance of precision and recall that often determines the practical use of a model in real-world application. The ensemble model's balanced performance suggests it might be the best model for

deployment in environments with different type of network traffic. The FCNN model showed lower performance because of simple tabular dataset. FCNN may outperform the machine learning models with more complex dataset.

7 Conclusion and Future Work

7.1 Conclusion:

This study effectively developed and evaluated machine learning model CART, deep learning model FCNN, and ensemble model (FCNN + XGBoost) for detecting rouge access point in wireless network. The research tackled significant issues identified in previous studies such extensive dependence on specific data characteristic, old and significantly small dataset that does not seem applicable for modern networks, and difficulties in managing complex dataset, by using the AWID 3 dataset. The CART model demonstrated the highest accuracy with very low false positives, which makes it particularly effective for real world application. The FCNN model showed the lower performance and potential of refinement for this study. Lastly, ensemble model demonstrated balanced between precision and recall.

7.2 Future Work:

The future research should prioritize the enhancement in FCNN model to optimize the accuracy in complex networks. Investigating other deep learning model could enhance the ability to recognize more complex pattern and identify the type of attack. Moreover, it is important to implement these models into real-world networks to validate their efficacy on wider scale. Finally, with the regular updating of the dataset, model training to detect new attack patterns is also essential.

References

Alsughayyir, B., Qamar, A. M. and Khan, R. (2019) "Developing a network attack detection system using deep learning," in 2019 International Conference on Computer and Information Sciences (ICCIS). IEEE.

Arisandia, D., Ahmad, N. M. and Kannan, S. (2022) "A detection technique using dual authentication stages framework for Rogue Access Point identification," IOP conference series. Earth and environmental science, 1083(1), p. 012091. doi: 10.1088/1755-1315/1083/1/012091.

AWID - 3 (2021) Aegean.gr. Available at: https://icsdweb.aegean.gr/awid/awid.

Babu, B. S. et al. (2023) "Network intrusion detection using machine learning algorithms," in 2023 3rd International Conference on Smart Data Intelligence (ICSMDI). IEEE.

Chen, C. et al. (2022) "FCNN-SE: An intrusion detection model based on a fusion CNN and stacked ensemble," Applied sciences (Basel, Switzerland), 12(17), p. 8601. doi: 10.3390/app12178601.

Dsouza, A. et al. (2022) "Real time network intrusion detection using machine learning technique," in 2022 IEEE Pune Section International Conference (PuneCon). IEEE.

Hossain, Z. et al. (2021) "Network intrusion detection using machine learning approaches," in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE.

Jin, J., Lian, C. and Xu, M. (2019) “Rogue base station detection using A machine learning approach,” in 2019 28th Wireless and Optical Communications Conference (WOCC). IEEE.

Khan, K. et al. (2020) “A survey on intrusion detection and prevention in wireless ad-hoc networks,” Journal of systems architecture, 105(101701), p. 101701. doi: 10.1016/j.sysarc.2019.101701.

Kim, D., Shin, Dongil and Shin, Dongkyoo (2018) “Unauthorized access point detection using machine learning algorithms for information protection,” in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE.

Shetty, S., Song, M. and Ma, L. (2007) “Rogue access point detection by analyzing network traffic characteristics,” in MILCOM 2007 - IEEE Military Communications Conference. IEEE.

Shone, N. et al. (2018) “A deep learning approach to network intrusion detection,” IEEE transactions on emerging topics in computational intelligence, 2(1), pp. 41–50. doi: 10.1109/tetci.2017.2772792.

Sumaiya Thaseen, I., Poorva, B. and Ushasree, P. S. (2020) “Network intrusion detection using machine learning techniques,” in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE.

Surendiran, B. and Vadivel, A. (2011) Feature selection using stepwise ANOVA discriminant analysis for mammogram mass classification, Psu.edu. Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e0f88e7f396e952c667eed360fb8b03a1ace7695.

Taher, K. A., Mohammed Yasin Jisan, B. and Rahman, M. M. (2019) “Network intrusion detection using supervised machine learning technique with feature selection,” in 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST). IEEE.

Thapa, N. et al. (2020) “Comparison of machine learning and deep learning models for network intrusion detection systems,” Future internet, 12(10), p. 167. doi: 10.3390/fi12100167.

Vaidya, T. S. (2023) Identifying inappropriate access points using machine learning algorithms RandomForest and KNN. Dublin, National College of Ireland.

Wei, P. et al. (2019) “An optimization method for intrusion detection classification model based on deep belief network,” IEEE access: practical innovations, open solutions, 7, pp. 87593–87605. doi: 10.1109/access.2019.2925828.

Wu, W. et al. (2018) “PRAPD: A novel received signal strength–based approach for practical rogue access point detection,” International journal of distributed sensor networks, 14(8), p. 155014771879583. doi: 10.1177/1550147718795838.