National College of Ireland

Effectiveness of Supervised and Unsupervised algorithms in detecting RAPs in Wireless Networks

MSc Research Project
Cyber Security

Ahmed Alazawy
Student ID: x23158352

School of Computing
National College of Ireland

Supervisor: Eugene McLaughlin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ahmed Alazawy		
Student ID:	X23158352		
Programme:	MSc Cybersecurity	**Year:**	2023/2024
Module:	Practicum 2		
Supervisor:	Mr. Eugene Mclaughlin		
Submission Due Date:	12th of August 2024		
Project Title:	Effectiveness of Supervised and Unsupervised algorithms in detecting RAPs in Wireless Networks		
Word Count: 7192	**Page Count**: 20 pages		

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: Ahmed A

Date: 11th of August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□ X
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□X
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.	□ X

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):	

AI Acknowledgement Supplement

Practicum Part 2

Effectiveness of Supervised and Unsupervised algorithms in detecting RAP' in Wireless Networks

Your Name/Student Number	Course	Date
Ahmed Alazawy/x23158352	MSc Cybersecurity	11/08/2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
N/A	N/A	N/A
N/A	N/A	N/A

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
N/A	
N/A	N/A

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

N/A

Additional Evidence:

N/A

Effectiveness of Supervised and Unsupervised algorithms in detecting Rogue Access Points in Wireless Networks

Ahmed Alazawy
X23158352

Abstract

In this paper, the performance of supervised and unsupervised ML Classifiers in identifying RAPs in wireless networks are discussed. Our research focuses on the main research question which fuelled this research project and guided the development which is "How do unsupervised algorithms like Isolation Forests and One-Class SVMs compare in their effectiveness at detecting Rogue Access Point attacks in Wi-Fi networks compared to other Supervised Algorithms?". To answer the above research questions, we utilized the AWID public dataset, which consists of labeled and unlabeled data of Wi-Fi networks. The research shows that the techniques Isolation Forest and One-Class SVM while applying unsupervised learning have the anomaly detection capability but lack accuracy and reliability as compared to supervised learning techniques. However, in term of accuracy, precision, recall and F1 two supervised models of Random Forest and Supervised SVM performed better than the rest models. This study contributes to the future work in cybersecurity by mapping the areas of unsupervised learning in network security and asserting that supervised learning models are effective. Besides, it contributes to the literature on the use of machine learning in network security and lays the groundwork for further research that might examine further hybrid models in detecting rogue access points or even better feature engineering techniques or using a more complex dataset.

1 Introduction

Wireless networks are perhaps one of the key necessities in the present day connected world as interactions between diverse devices across different platforms are necessary. While the use and integration of these network structures has increased and extended across organizational boundaries the exposure of these networks to threats to security has also increased. One of the most dangerous of them is the Rogue Access Points (RAPs), it is access points, created intentionally within a security system to spy on or even tap into the data transmission line. The recognition and elimination of these threats are necessary for preserving network stability and users' confidentiality. It has been shown that various machine learning techniques can be useful in distinguishing network threats including RAPs Juwale (2020) Reyes et al. (2020). These works explain how Decision trees and random forests can help further filter through the analysis and data to detect the various anomalies. However, these approaches always face the difficulties of scalability and real-time processing especially in the large data or high speed data flow. It also draws attention to the need to acquire additional methods of machine learning that may effectively run under such restrictions. Regarding this unsupervised learning models which are Isolation Forests and One-Class SVMs can be useful.

RAPs are involved in different types of negative actions, including information leakage, spam distribution, and malware distribution. However, with many detection methods available, these are dependent on such approaches as supervised learning which require large datasets

that are well labeled. That is, they are can be impractical due to the dynamic and unpredictable nature of the network environments and the frequent lack of enough labeled data. This research is based on the rationale that there is a gap on how unsupervised learning techniques can be used in the independent identification of anomalies in network traffic without the manual tagging of data. In this case, AWID public dataset will be used right from this study due to its intensive collection of both labeled and unlabeled Wi-Fi network data thus offering credible starting point when it comes to performance benchmark of unsupervised machine learning algorithms to detect RAPs.

1.1 Research Question, Research Objectives, Hypothesis and Contribution

"How do unsupervised algorithms like Isolation Forests and One-Class SVMs compare in their effectiveness at detecting Rogue Access Point attacks in Wi-Fi networks, compared to other Supervised Algorithms?"

Research Objectives:

- Evaluate and compare the effectiveness of both Isolation Forest and One-class SVM Algorithms
- To assess the scalability and applicability of the two algorithms
- Evaluate and compare the supervised and unsupervised ML models

Hypothesis:

- One-Class SVM will provide higher precision and Accuracy score in detecting RAPs when from the AWID dataset
- The unsupervised models will prove to be more successful in accuracy of multiple classes.

Contribution to scientific literature:

- Making a comparative analysis of Isolation Forests and One-Class SVMs using AWID dataset which is an established benchmark in network security studiy.
- Improv understanding of how different data extractions affect the detection of machine learning algorithms.
- Offers practical knowledge for network proffesionals on the deployment of these algorithms to help protect against rogue access points.
- Gives new insights on these unsupervised algorithms and how they differ from traditional machine learning algorithms.
- Produces a new framework that allows anyone to continuously monitor and update which will provide reliability in the long term.

1.2 Structure of Report

The report here has multiple different sections that all provide a different view to the whole process of the research and application of research. Firstly, we have the ‘Introduction’ which is used to give context and introduce the reader to the full overview of the project, as well as that, it will provide an idea to reader of what to expect in the coming sections of the report. Next up is the ‘literature review’ section which will ultimately provide an overview of what approaches has been done before in previous research papers, what they do and don’t excel at and how parts of the research papers can be used to give an idea of how to develop my own. Moving on, there is the ‘methodology’ section in which this section will discuss the research

process that was undermined and the approach done in terms of data collection. The next section will be 'Design Specification' which will discuss and layout the framework and architecture of the design of the project including and PC specifications. 'Implementation' section will be where the majority of the implementation of the artefact/product will be discussed and where the main implementation techniques will be discussed and fleshed out to show how the final implementation was achieved. Next up is the 'Evaluation' section which is where I will be providing information, graphs and evidence of comprehensive analysis of the results that were obtained after the implementation. Finally, it ends off on 'Conclusion and Future Work' the final verdict will be discussed here and where it will be stated if the project was successful and explaining how things went wrong or right and how I feel bout it and it is concluded with how I will be further expanding on any work and any potential research projects that could be done based on this project.

2 Literature Review

2.1 Detecting Rogue Access Points in Wireless Networks

The detection of Rogue Access Point (RAP) is important since connected wireless networks are associated with many security threats in a number of industries and sectors including business and public access networks. Any unauthorized access can result in critical security infringement, such as unauthorized data access, eavesdropping, and even further attack possibilities such as the Man-In-The-Middle attack that was explored in detail in some of the research papers that I analysed (MITM) (Das et al. 2022)

Comprehensive Approaches to RAP Detection: (Kolias et al.) (2016) has provided one of the most comprehensive review of multiple ML algorithms that they have tested on AWID dataset, which is specifically designed dataset for intrusion detection in Wireless Networks. Their analysis incorporated extensively classifiers such as AdaBoost and specifically classifiers such as the Random Forest. They were utilized in the analysis and identification of the capacity of a given network in recognizing and blocking the potential threats that originate from RAPs or differentiate between real traffic and a threat. As for the strengths of the study, the adjustability of the ensemble techniques such as the Random Forest, which employs several learning algorithms to enhance the detection precision, and offers protection against various attack types including RAPs, has been presented well.

A different approach of machine learning is used here (Juwale) (2020) by applying GA, SVM, and KNN on RAP detection with the help of RTT data. It is highly commendable that the paper proposed the application of a new optimization algorithm called Ant Colony Optimization to enhance the feature selection aspect of the design. This approach proves to give a very high accuracy rate of 98 percent mean value of fifteen percent, suggesting that algorithms from nature can improve feature selection and thus favour the execution of IDS solutions.

The literature that was reviewed according to the topic of detecting Rogue AP in wireless networks show that, despite the use of machine learning in finding RAP, there is always a need to develop models that have both high accuracy and reasonable computational cost. Micro-studies can be conducted in future to identify real-time lightweight algorithms that can run at the edge of the network hence minimizing latency and bandwidth utilization. Moreover, applying unsupervised types of learning could help in the case of the lack of labeled data which is often typical in real-life situations.

2.2 Comparative Analysis of One-Class SVM, Isolation Forest and Other ML Algorithms for Network Security

One-Class SVM is specifically developed to take in data that define normal operations and view variations from these as abnormal. This aspect is especially helpful in the fields where abnomalities depict threats in the network, for instance, unauthorized entry points. While the specific type of One-Class SVM and their usage in the context of RAP detection isn't addressed in the papers, theoretical applicability of One-Class SVM is similar to other scenarios and techniques related to anomalous activity detection, which have been proposed in the works of Saed et al. (2022) where the authors provided an overview of ML for intrusion detection. They speak about the possibilities of using ML in recognizing many folded patterns that signify security threats, which is similar to how One-Class SVM may be involved in similar applications.

It is evident that Decision Trees and Random Forests are often employed in the papers for network security applications since they provide satisfactory results in the classification tasks. For example, Das et al. (2022) uses the mentioned algorithms to detect patterns that suggest MITM attacks from network traffic. One-Class SVM involves in modeling normal behavior and then identifying the outliers while Decision Tree and Random Forests classify according to the features of normal and abnormal samples or instances. This can be helpful in the situations where multiple threat categories are applicable, and since C-SVM covers more general space it would be safer in its protection compared to OSCM and One-Class SVM.

2.3 Summary

Research paper	**Approach**	**Dataset**	**Algorithm**	**Limitations**	**Advantages**
Othman et al., 2018	Spark-Chi-SVM model was used for intrusion detection.	KDD99 dataset	SVM	Type intrusions are not detected.	The use of the SparkChi-SVM model makes the detection effective.
Kasongo et al. 2019 [1]	Wrapper-based feature extraction that uses Feed Forward Deep Neural Networks (FFDNNs)	UNSW-NB15, AWID	FFDNNs with Extra Trees (ET) algorithm used for feature extraction	Its limited to the datasets used and requires more investigation into detection rates for each classes	The paper has high accuracy for both binary and multiclass classification. Performs better than traditional ML methods.
Vaca et al. 2018 [2]	Uses ensemble learning approach with bunch of ML algorithms combined	UNSW-NB15	Ensemble of multiple ML algorithms	Has potential computational complexity because of ensemble approach	Has Improved detection rates compared to single algorithm approaches and is robust through ensemble technique
Reyes et al. 2020 [3]	Two-stage ML-based Wi-Fi NIDS with feature selection	AWID	ML, XAI, enssemble methods	Complexity in the realtime applications	High accuracy, reduced feature set, explainability using XAI
Ige et al. 2024 [4]	Reviewed state-of-the-art machine learning approaches for cyber atttack detection	Didn’t focus on one particular dataset but used multiple datasets from different studies	Random Forest, SVM, Logistic Regression, , CNN	Issues with dataset availability, also issues with unbalanced datasets and needs wide scenario testing to validate findings	A good detailed comparison of ML alogrthms, insights into strengths and weaknesses for specific attacks

Cetin et al. (2019) [5]	Federated learning approach leverages decentralized data processinng to have mor privacy	AWID	Stacked Autoencoders	Complexity and scalability of federated learning	makes user privacy beter by processing local data, reduces communication overhead
Saed et al. (2022) [6]	Uses various machine learning techniques to identify and classify MITM attacks in wireless networks.	Wi-Fi network benchmark dataset	supervised and semi-supervised learning techniques with deep learning models	Depends a lot on good dataset quality and diversity,has a lot of computational demand.	High detection accuracy indetecting complex MITM attack patterns.
Latha et al. (2022) [7]	ML IDS designed to detect De-authentication attacks	NSL- KDD	KNN, SVM and Logistic Regression	Depends on specific frame features which might not generalize across different network settings.	High detection rate of 89% for De-authentiication attacks
Wang et al. (2019) [8]	Deep learning approAach using Stacked Autoencoders and Deep Neural Networks (DNN) for attack classification	AWID	SAE and DNN	Has high complexity in model training, is limited to the types of attacks available in AWID dataset	Has high classification accuracy for bunch of different attack types
Saxena et al. (2014) [9]	Hybrid approach using SVM which is optimized by Particle Swarm optimization and feature reduction	KDD99	SVM-PSO	Depends on optimal parameter tuning for SVM and effective feature selection	High detection rate and reduced feature dimensionlality through good preprocessing
Kolias et al. (2016) [10]	Empirical evaluation of threats using ML algorithms on a public dataset	AWID Dataset	AdaBoost, Hyperpipes, J48, Naive Bayes, OneR, Random Forest, ZeroR	Has challenge in adapting for new evolving attack techniques	empirical evaluation that was conducted showed that it was effective
Sathya et al. (2020) [11]	ML used to detectmultiple Wi-Fi BSSs in LTE-U CSAT environments done by analyzing energy values	LTE-U	Neural network models, SGD	Needs collection and training on specific data	Is high accuracy and simpler than decoding Wi-Fi packets
Perera et al. (2016) [12]	Feature selection and ML techniques for intrusion detection on apublic Wi-Fi dataset	AWID	OneR, Ada Boost, J48, Random Forest, Random Tree	Depends on really good feature selection for performance	Has an improved processing time and accuracy with feature reduction
Agarwal et al. (2014) [13]	A ML approach for detecting and localizing DOS attack	802.11 networks	Machine learning with AoA based localization	Process isimited to specific types of DoS attacks	high detection accuracy and helps in swift recovery
Das et al. (2022) [14]	Ensemble learning approach todetect MITM attacks using a few ML algorithms	IoT Intrusion Detection Dataset	ANN, SVM, Decision Tree, k-Nearest Neighbors, Random Forest, Logistic Regression,	Relies on high-quality feature selection and preprocessing to be most effective	Effective combination of algorithms makes better detection accuracy
Juwale (2020) [15]	Using multiple ML algorithms to detect unauthorized access points via RTT data	Synthetic RTT dataset	SVM, KNN, Genetic Algorithms, Ant Colony Optimization	Depends on synthetic dataset so might not generalize to real-world conditions	High accuracy (98.15%) was shown by Ant Colony Optimization in detection

Table (1): summary of all the literatures

2.4 Literature Gap

The research that is being conducted has a goal of figuring out the effectiveness of machine learning algorithms in identifying unauthorized access points in wireless networks and evaluating the results that they provide. The problem is that we have a lack of approaches that are available online that incorporate ML algorithms to identify unauthorized access points specifically. There is a notable gap in focused comparative studies of unsupervised ML algorithms, specifically with Isolation Forest and One-Class SVM, which are focused on identifying RAPs using the AWID dataset. Although using unsupervised machine learning

approaches is best for long term effectiveness against network intrusions not many research papers used them. Unsupervised algorithms have a big role in identifying outliers or anomalies in data which is exactly what unauthorized access points actually are in a network environment making it good for this situation and with the AWID dataset it does not have the most comprehensive labeling so unsupervised algorithms would hopefully be suitable.

There is a lack of detailed comparative analysis on the algorithms that I have chosen under realistic network conditions shows a critical literature gap and so I am hoping to trv fill this gap by leveraging the AWID dataset to only compare the effectiveness of Isolation Forests and One-Class SVMs in detecting Rogue access points and I might potentially add other algorithms probably supervised learning algorithms to additionally compare the effectiveness of detecting RAPs even further.

3 Research Methodology

3.1 Approach Taken

A system was setup for the approach of identifying and detecting unauthorized access points in wireless networks utilizing machine learning algorithms with the AWID Dataset. The procedure will be split into two phases, training phase and prediction phase. In the training phase the data from the dataset will be read and pre-processed and further on after that, the ANOVA test is going to be used to get the best features and apply them into both models which will then be trained. After that the preprocess data will be used to load the trained models and make predictions for how successful they are at detecting RAP's and then get the results and compare and evaluate them. The figure below shows that whole process in a high level overview.

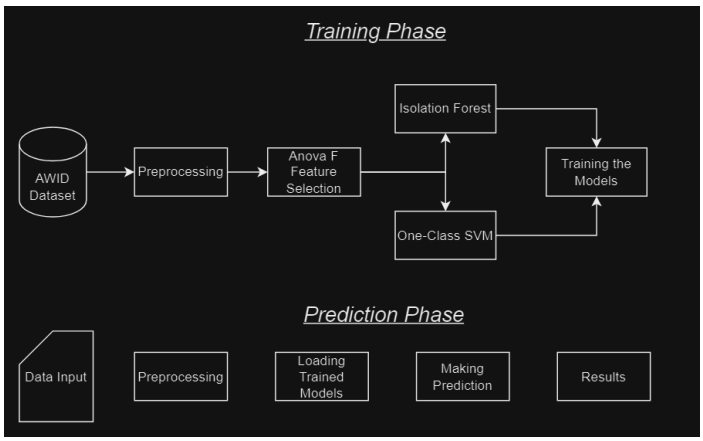

Figure 2 – Workflow Diagram

3.2 Research Procedure

In order for me to actually understand the reasoning behind the uses of every aspect of the project I had to undergo a lot of research online. The procedure was designed carefully in order to ensure that the study that was done was both informative, replicable and with rigorous evaluation. The research that was done was used to ensure that all the tools and models and datasets were chosen correctly and they all work together and that there is enough resources available to complete the study.

3.3 Data Collection

The specific generalised dataset employed in the present study was AWID dataset, which has newly formulated specifically for the research on the Wireless network security. This dataset is a result of frequent monitoring of traffic in the Wi-Fi network; it comprises a number of parameters, which are necessary to examine in case of the network security threats, such as SSIDs(name of wi-fi network), BSSIDs(mac address of wireless access points), MAC addresses(indicator for network devices), a time stamp, the headers of packets, and detailed information related to the payloads. On this note the AWID dataset differs from the regular data that is available in the machine learning, that is, the database includes both labeled and unlabeled data, thus making it the most appropriate database to use in training and testing the ML model under the actual network conditions. I used a CSV file that had a mix of both normal activities and rogue access point simulations to give us a more balanced view of normal networks and also potential security threats. The dataset had various types of attacks which were labeled as class and these included ‘normal’, ‘injection’, ‘flooding’, ‘impersonation’. The normal class means that the network hasn’t been attacked yet and the other three indicate a specific attack that has been committed to the network.

3.4 Data Preprocessing

As part of the pre-processing phase, all data that is not useful or needed for the overall system and model development is filtered out from the dataset. Since there is so much data within the dataset, in order for the performance of the models to be sufficient only the needed data should be used. All rows and columns that have null values will be taken out for further success for model development. Missing data can significantly impact the performance of machine learning models. The approach was used to fill in the missing values with the mean of their respective columns so that no loss of data happens and it keeps statistical integrity. Machine learning models need numerical data so categorical data needs to be changed into numerical. Columns that are non numeric were converted using label encoding, which makes a unique integer to give each category. Binary indicators were also created for potentially rogue SSIDs and BSSIDs that are providing explicit signals to the models when potentially malicious identifiers are detected.

3.5 Feature Engineering

After data preprocessing was done, to enhance the model’s ability to detect rogue access points, I performed feature engineering based on domain knowledge so part of this included making binary indicators for potential rogue SSIDs and BSSIDs which are very closely related to rogue access point interference, and engineering features like signal strength differences and also packet rates which usually indicate of unauthorized network access. The ANOVA test or ANOVA-F Test conducted a test to see and choose the top 10 most significant features so that these 10 features can be used for model evaluation and testing and increase the efficiency of the models.

3.6 ML Model Development

Once the data has been collected, pre-processed and feature engineering has been done, it was time to utilise that data in order to train the ML models. The data that I have was divided into

two sets of data, training and testing. 80% of the data will be used to train the machine learning techniques and the remaining 20% will be prioritised for testing and evaluating the performance of the machine learning techniques.

Two main unsupervised models were chosen because of their suitability for anomaly detection and were utilizing the data that has been prepared for them specifically.

3.6.1 Isolation Forest

This is an algorithm for the anomaly detection that separates the anomalous profiles instead of profiling the normal cases. This obviously makes it effective due to its approach of making observations ‘isolated’ by choosing the feature at random, then choose a split value at random between the maximum and minimum values for the said feature. This random partitioning creates rather noticeable paths in the data structure, while generally, anomalies have shorter paths within the trees of the forest due to the need for fewer additional conditions to filter them from the rest of the data. Using the trained forest, model applied the data points in the test set to the different classification depending on the length of the path that would have to be travelled to isolate them. Even shorter paths were considered as possible RAPs because of their anomaly status.

3.6.2 One-Class SVM

One-Class SVM modification of the support vector machine is applied for the case when the anomaly class is underrepresented or entirely unknown at the training stage. It functions in the same way that it puts a circle around the 'normal' shaped data points placing any point outside this ellipses as an outlier or an anomaly. This boundary is determined by the SVM's aim of maximizing the margin around different classes while adjusted in this case to enclose the largest possible volume of points assumed to be normal.

In this project, the One-Class SVM will be trained only on what the model described as normal activity within the AWID dataset since I am aware of its existence; The kernel that is going to be used is a radial basis function (RBF) kernel in order to deal with the non-linear data set. Thus, the model was learning the boundary of normal behavior using one set of features that was expected to contain typical characteristics of network activities.

3.6.3 Model Evaluation

The performance of each model will be evaluated using a few different metrics and these include: Accuracy, Precision, Recall, and F1 Score and these metrics will give a view of each model's performance. Using confusion matrix will be able to give detailed insight into the models classification accuracy in different categories. Each model's performance will be visualized using plots and various diagrams will be made in order to evaluate the effectiveness of both models.

4 Design Specification

4.1 System Specification

The system that is being used to detect rogue access points in a wireless network is a Windows PC that has:

- 16GB RAM with an 11th gen Intel Core I7-11370H with an RTX 3050 graphics card and 1TB SSD for storage
- Jupyter Notebook which would hold all the work done
- Python programming language
- Visual Studio Code to run the code in a .py file

4.2 System Architecture

The system had an architecture that utilizes many stages which consist of data ingestion from the AWID public dataset and with the data from AWID, preprocessing would be done and feature engineering to mainly only use the main significant features that are helpful for detecting wireless networks which are then used for the purpose of training Machine learning models; Isolation Forest and One-class SVM that I am incorporating into our testing architecture. These models will be used to detect inappropriate access points based on the Wi-Fi data that is in AWID dataset.

5 Implementation

5.1 Final Stage of Implementation

Once pre-processing was completed and only the necessary data is left, then the final stages of implementation were worked on. As you already have learned from this report that ultimately, rogue access points need to be detected so feature engineering was done specifically for features that correspond to the rogue access points so I used lambda function to check if columns ‘ssid’ and ‘bssid’ exist with the term ‘rogue’ and furthermore use feature engineering to check for more rogue access point specific for ‘signal_strength_diff’, ‘num_packets’ and ‘duration’. Moving on to help further with the classifier’s development using StandardScaler() to make sure that all the features contribute equally to the model’s by normalizing the mean and variance and it improves the algorithms accuracy. Used ANOVA F-Test to select the top 10 features which will be used for model training which will identify and keep the top 10 strongest features to give better efficiency and accuracy. The ANOVA F-test scores were then extracted and sorted in ascending order in order with ‘’ to show the importance of some of the features over the others and then plot these scores in a horizontal bar graph to display the importance of the top 10 features as seen in the figure below.

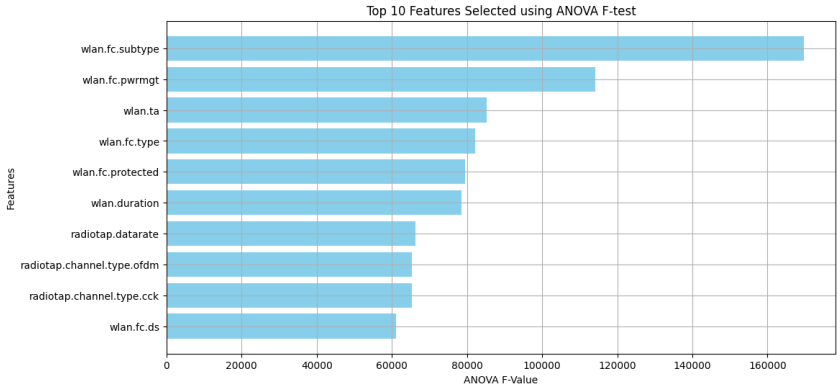

Figure 3 – ANOVA F-test scores sorted in ascending order in a graph

It is then necessary to check if the target variable ‘y’ is of object type ‘O’ which is categorical data and then I used ‘LabelEncoder’ to convert it into integers as the upcoming ML classifiers require full numerical input to work efficiently. Finally, before implementing and evaluating the models to see which detect rogue access points the best, I split the data into 80% train and 20% test sets where the 80% is allows the model to learn and adapt to the patterns in the data and the 20% evaluates the models performance for new data and this is done by using ‘train_test_split()’ from the Scikit-learn library’s ‘model selection’ and with this step out of the way it was time to start implementing the ML Models as you will see in the next sub-chapter.

5.2 Model Implementation

Ultimately, two main models were developed to compare, analyse and compare their effectiveness to detect unauthorized access points and they are Isolation Forest and One-class SVM, and in addition to them, an additional two other ML models were used which are known to be universally effective were used due to their similarities to the main two algorithms and they are Random Forest which is similar to Isolation Forest as they are both built on decision-trees supervised SVM which is similar to One-class SVM but can only really successfully be set on one class so a clear contrast can be made between the two. The following is how these models were developed and their outputs.

Isolation Forest:
Isolation forest was imported in using the ‘sklearn.ensemble’ library from python and is set with the contamination rate of 0. 1 and trained with 'X_train' so that it can identify the anomalies, then predictions for both training and testing sets are produced based on which the final predictions are made and then transformed into the binary scale of the model's natural output. then, the results are analyzed with the accuracy, precision, and recall, and F1-score measures on the train and test data.

One-class SVM:
One-class SVM was imported using the ‘sklearn.svm’ library. The model is initialized with the RBF Kernel which is used to deal with nonlinear data. The ‘nu’ parameter is set to 0.01 so it is more sensitive to outliers and including more support vectors. The ‘gamma’ parameter is used to scale based on the variety of features. The model is then trained with the training data set and predict on the test dataset and finally evaluated using ‘evaluate_model’.

Random Forest:
Isolation forest is an ensemble model used from the ‘sklearn.ensemble’ library. The ‘RandomForestClassifier’ is set up with 100 trees aswell as a fixed ‘random_state’ for consistent results. It is then trained and generates predictions using ‘predict(X_test)’ for the test set which is for unseen data. Finally, the ‘evaluate_model’ function tests the models performance based on a couple of metrics like accuracy, precision, recall and F1-score as well as confusion matrix which is going test the predictions against the actual test label which is ‘y_test’.

SVM:
The Supervised SVM Algorithm is imported from the ‘sklearn.svm’ library as svc. It is also initialized with the RBF kernel to handle complex data and ‘c’ parameter is set to 1.0 which is used to achieve low error but with a better decision boundary. Then ‘gamma’ is set to scale

again. After that, the model is trained on the training set and then predicted based on the test set and finally the 'evaluate_model' function evaluates the predictions.

5.3 Tools and Languages Used

There were various different tools and languages that were used in the implementation stage of the project and some of these include:

- Scikit-learn: The Scikit-learn library was the main library that gave me all the good and robust tools to develop the models and evaluate the metrics
- Pandas: Pandas was used a lot in the code for pre-processing as well as manipulating data to suit our implementation
- Matplotlib and Seaborn: These two were used to visualise the results and feature selection
- Numpy: was used to convert the predictions of both testing and training sets to binary format

6 Evaluation

6.1 Results

There were various sets of results that were gained from evaluating each models performance and accuracy of how well it can detect wireless network attacks specifically with rogue access points and in the upcoming sub chapters below these will be displayed, explained and evaluated. Before I explain what the details in each classification report identify, it is important to know what they indicate:

- Precision: finds the accuracy of all the positive predictions
- Accuracy:
- Recall: finds the ability of a certain model in how they find all the relevant true positives in the dataset and true positive means correct identification
- F1-Score: is the average of both precision and recall and the ideal score is 1 and the worst is 0.
- Support: is the number of times each class is seen in the dataset
- The classes 0 – 3 are the following; 0 = Normal, 1 = Flooding, 2 = Injection and 3 = Impersonation

6.1.1 Isolation Forest

The classification report below for the Isolation Forest model shows various factors of the performance of the model. As you have already seen from the details above, the results of those scores vary between machine learning models as some have achieved great results whereas others not so much which you will view in this sub chapter and the others below. In this classification report for IF model, there is only results for class 0 where precision score is low which indicates a lot of false positives while recall score is high which shows there is good coverage of the actual positive class and F1 score is at 29% meaning there is a moderate imbalance between precision and recall and the rest of the 3 classes are sitting at 0% meaning they are incredibly poor and the reason for this would be that since there is a high false positive rate for class 0, this has the effect of misclassifying the other class as 0 too as seen in figure below. In terms of confusion matrix which is a table which shows the number or true positives, true negatives, false positives, true negatives and false negatives, the one that was

produced by Isolation Forest was not good as it is seen that it fails to correctly identify actual cases for each class.

```
--- Isolation Forest ---
Classification Report:
              precision    recall  f1-score   support

           0       0.18      0.87      0.29      7778
           1       0.00      0.00      0.00      7687
           2       0.00      0.00      0.00      9435
           3       0.00      0.00      0.00     17338

    accuracy                           0.16     42238
   macro avg       0.04      0.22      0.07     42238
weighted avg       0.03      0.16      0.05     42238

Confusion Matrix:
[[ 6783   995     0     0]
 [ 7684     3     0     0]
 [ 9423    12     0     0]
 [14336  3002     0     0]]
Accuracy: 16.07%
Precision: 3.28%
Recall: 16.07%
F1 Score: 5.44%
```

Figure 4 – IF Classification Report

6.1.2 One-Class SVM

The classification report for One-class SVM is displayed below and you can notice that with this unsupervised model, 2 classes have been identified and scores were provided with both and it is noted that results vary in each class and the scores for the first 2 classes are actually pretty valid unlike the other 2 classes which provided a 0 for each respective score showing that it may not be good at predicting more than one class which its name implies whereas it could provide good and reliable score in one class and so the average that it got for accuracy and recall were around 31% which reflect poor detection rates too. One-class SVM is used for anomaly detection which is done in binary so normal or anomaly and in the confusion matrix it showed poor diagonal values so it is not suited for being able to distinguish multiple classes.

```
--- One-Class SVM ---
Classification Report:
              precision    recall  f1-score   support

           0       0.21      0.99      0.35      7778
           1       0.87      0.68      0.76      7687
           2       0.00      0.00      0.00      9435
           3       0.00      0.00      0.00     17338

    accuracy                           0.31     42238
   macro avg       0.27      0.42      0.28     42238
weighted avg       0.20      0.31      0.20     42238

Confusion Matrix:
[[ 7689    89     0     0]
 [ 2456  5231     0     0]
 [ 9423    12     0     0]
 [16634   704     0     0]]
Accuracy: 30.59%
Precision: 19.68%
Recall: 30.59%
F1 Score: 20.31%
```

Figure 5 – OC SVM Classification Report

6.1.3 Random Forest and Supervised SVM

The following two classification reports are shown below and it is clear that these both indicate very strong performance. At 96.51% random forest indicates really good

performance and consistency between the different sets of classes. It also looks like random forest shows a balanced approach between precision and recall scores, not to mention the confusion matrix shows minor areas where you would be able to make better most likely due to misclassification with similar classes but overall seems completely normal and good at detection of rogue access points and that is the same with the Supervised SVM Model which comes at slightly lower accuracy and precision at 94.47% accuracy which similarly also reflects a perfect score in the data in class 2 and since it is only slightly lower than random forest is shows that there is still room for improvement and the confusion matrix once again looks well mapped out and you can notice that class 1 has a good amount of its instances misclassified in class 3. Overall, it seems these 2 supervised classifiers show much more versatility, efficiency and scalability between classes.

```
--- Random Forest ---
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.94      0.97      7778
           1       0.89      0.99      0.94      7687
           2       1.00      1.00      1.00      9435
           3       0.97      0.95      0.96     17338

    accuracy                           0.97     42238
   macro avg       0.96      0.97      0.97     42238
weighted avg       0.97      0.97      0.97     42238

Confusion Matrix:
[[ 7346     0     0   432]
 [    0  7576     0   111]
 [    0     0  9435     0]
 [    7   924     1 16406]]
Accuracy: 96.51%
Precision: 96.69%
Recall: 96.51%
F1 Score: 96.54%
```

Figure 6 – RF Classification Report

```
--- Supervised SVM ---
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.89      0.91      7778
           1       0.93      0.92      0.93      7687
           2       1.00      1.00      1.00      9435
           3       0.92      0.95      0.94     17338

    accuracy                           0.94     42238
   macro avg       0.95      0.94      0.94     42238
weighted avg       0.94      0.94      0.94     42238

Confusion Matrix:
[[ 6893     0     0   885]
 [  109  7110     0   468]
 [    0     0  9435     0]
 [  367   506     1 16464]]
Accuracy: 94.47%
Precision: 94.48%
Recall: 94.47%
F1 Score: 94.46%
```

Figure 7 – Supervised SVM Classification

6.2 Accuracy Scores of Models

A bar graph was produced which would display the four classifiers side by side with distinct colouring to show the clear differences between them in terms of their accuracy scores where the accuracies are the ratios of accurate predictions consisting of both true positives and true negatives based on the total number of predictions. It is very important measurement as it shows how effective the predictions of a certain model is and the graph below also shows us that all the feature engineering I done and pre-processing is still effective when dealing with supervised models hence the high accuracy scores.

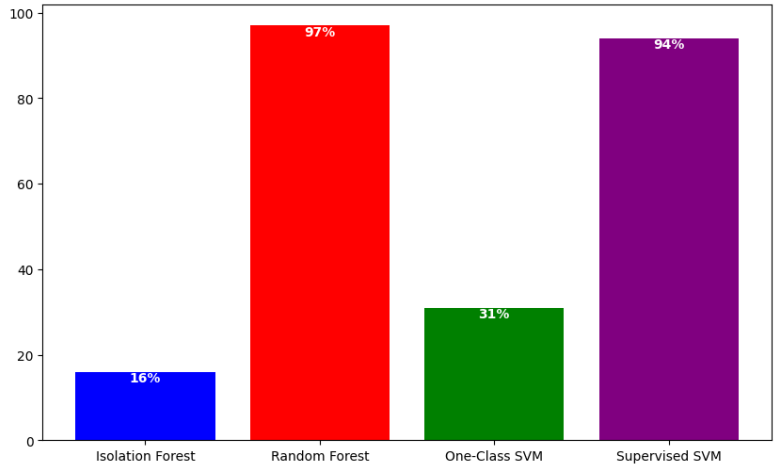

Figure 8 – Bar graph of Accuracy of the four models

6.3 Precision Scores of Models

The dot plotting graph below displays the comparisons of the precision scores of the 4 machine learning algorithms displaying their differences in measuring the proportion of precise predictions that each respective model made, and precision is the ration of true positives over the total of true positives and false positives and once again RF and Supervised SVM were more successful in that as seen in the graph below.

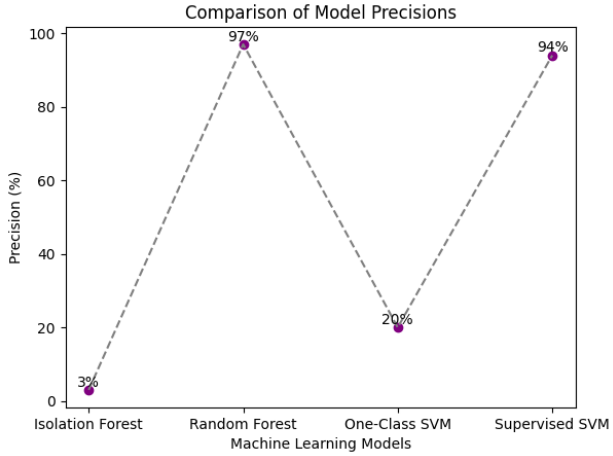

Figure 9 – Dot Plot of the precision of the four models

6.4 Recall Scores of Models

The following heat map displays the recall scores of the four models, to put it simply the higher recall values with random forest and supervised SVM means that rogue access points are more likely to be noticed which is really important for maintaining the safety of wireless networks.

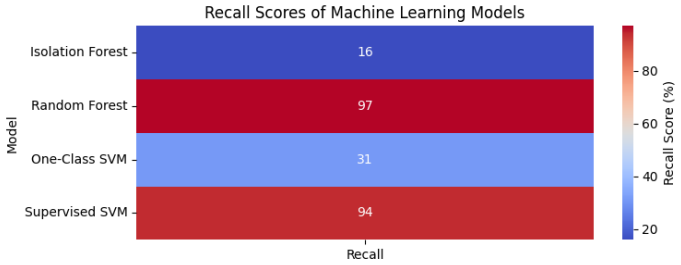

Figure 10 – Heat Map of the recall scores of the four models

6.5 Discussion and Critical Analysis

The low scores for both Isolation Forest and One-class SVM in the context of wireless network security especially in Isolation Forest meant that the model is just not reliable enough to use practically so to accurately identify if the data set as normal or rogue which could lead to risks like misidentifying completely normal access points as dangerous which is a big problem when trying to protect a network, while on the other hand, One-class SVM is a little better as its average accuracy scores are twice the percentage as Isolation forest but both are still blown out of the water compared to the supervised algorithms. At least with my process of evaluating isolation forest I was able to achieve a higher score than another literature that used it similarly but achieved an accuracy score of 16% while a lower score of 11% was achieved by sharma

(2022)[16]. Whereas a similar literature that used one-class SVM could not be found in respect to rogue access point detection so my usage of it is considered novelty but as seen with the flaws that it has similar to isolation forest it seems to be low across the board in this scenario. I felt like the experiments that I carried out were slightly unfair as isolation forest and One class SVM do not seem to work well with large datasets and even though the dataset was decreased in size it was still too much for it so if any improvements could be to the design could be noticed then it will be well worth to take a look at it. It seems like a more rough dataset with unclear values would be more effective to deal with unsupervised models but since we have clear training objectives of what seem to be normal and rogue behaviours supervised algorithms work best.

7 Conclusion and Future Work

7.1 Conclusion

In conclusion, the research question “"How do usupervised algorithms like Isolation Forests and One-Class SVMs compare in their effectiveness at detecting Rogue Access Point attacks in Wi-Fi networks, compared to other Supervised Algorithms?" was attempted and I honestly believe that I was able to explore the research question and answer it quite well. I was able to achieve results for both sides of the research question and compare them while conducting a series of experiments and metrics to really assess the difference in effectiveness between the two sets of machine learning algorithms. I did this by comparing them based on their accuracy, precision, recall and f-1 scores. This research also further reveals some limitations of unsupervised models in complex security contexts while also points out the necessity of having high quality labeled data which contributes to the improvements of model performances. However, the study's shortcomings are clear because they are conducted from the foundation of a single data set and the computational cost of model tuning, and because of this, the research opens the door to future studies that could turn to other models, for example, hybrid ones, or to more sophisticated tools for feature project that are discussed in next chapter.

I had issues like I had previously used a much larger dataset but it would take so long to actually run the models and train based on it so I went with an already compressed version of the AWID dataset that worked much better although the SVM model still took a long time but the time taken was drastically reduced. To summarize the conclusion, I would say that I tried my best to complete the research project and answer the research question quite comfortably.

7.2 Future Work

Looking back on the research project, it is evident that more work would be needed to fully optimise all the models in how well they detect rogue access point attacks. In addition, I could also work on wider attacks and potentially move on to different types of networks. I would also most likely do a follow up research report as I found this one super interesting and expanding onto more specific cybersecurity research outside of machine learning like in potentially the medical field would be interesting and meaningful in this society. I could also explore more machine learning algorithms especially deep learning ones which require a lot more careful analysis. Also I could actually implement a piece of software that would actively detect attacks in networks which would I couldn’t do due to time constraints but

overall I am excited to work on similar projects and expand my knowledge in the world of cybersecurity.

References

[1] Kasongo, S.M. and Sun, Y. (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Computers & Security, 92, p.101752. Kasongo et al. 2019

[2] https://ieeexplore.ieee.org/abstract/document/8548315
An Ensemble Learning-Based Wi-Fi Network Intrusion Detection System (WNIDS).
Vaca et al. 2018

[3] A. Reyes, A., D. Vaca, F., Castro Aguayo, G.A., Niyaz, Q. and Devabhaktuni, V. (2020). A Machine Learning Based Two-Stage Wi-Fi Network Intrusion Detection System. Electronics, 9(10), p.1689. doi:https://doi.org/10.3390/electronics9101689.

[4] Ige, T., Kiekintveld, C. and Piplai, A. (2024). An Investigation into the Performances of the State-of-the-art Machine Learning Approaches for Various Cyber-attack Detection: A Survey. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2402.17045.

[5] Cetin, B., Lazar, A., Kim, J., Sim, A. and Wu, K. (2019). Federated Wireless Network Intrusion Detection. 2019 IEEE International Conference on Big Data (Big Data). doi:https://doi.org/10.1109/bigdata47090.2019.9005507.

[6] Saed, M. and Aljuhani, A. (2022). Detection of Man in The Middle Attack using Machine learning. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICCIT52419.2022.9711555.

[7] ieeexplore.ieee.org. (n.d.). Deauthentication Attack Detection in the Wi-Fi network by Using ML Techniques | IEEE Conference Publication | IEEE Xplore. [online] Available at: https://ieeexplore.ieee.org/abstract/document/10099975.

[8] Wang, S., Li, B., Yang, M. and Yan, Z. (2019). Intrusion Detection for WiFi Network: A Deep Learning Approach. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, [online] pp.95–104. doi:https://doi.org/10.1007/978-3-030-06158-6_10.

[9] Saxena, H. and Richariya, V. (2014). Intrusion Detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain. International Journal of Computer Applications, 98(6), pp.25–29. doi:https://doi.org/10.5120/17188-7369.

[10] Kolias, C., Kambourakis, G., Stavrou, A. and Gritzalis, S. (2016). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. IEEE Communications Surveys & Tutorials, 18(1), pp.184–208. doi:https://doi.org/10.1109/comst.2015.2402161.

[11] https://ieeexplore.ieee.org/abstract/document/9049781 Machine Learning based detection of multiple Wi-Fi BSSs for LTE-U CSAT
(Sathya et al., 2020)

[12] Thanthrige, U.S.K.P.M., Samarabandu, J. and Wang, X. (2016). Machine learning techniques for intrusion detection on public dataset. [online] IEEE Xplore. doi:https://doi.org/10.1109/CCECE.2016.7726677.

[13] Agarwal, Mayank & Pasumarthi, Dileep & Biswas, Santosh & Nandi, Sukumar. (2014). Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization. Int. J. Mach. Learn. & Cyber.. 7. 1-17. 10.1007/s13042-014-0309-2.

[14] https://ieeexplore.ieee.org/document/9984365 Man-In-The-Middle Attack Detection Using Ensemble Learning (Das et al., 2022)

[15] Juwale, A.M. (2020). Analysis and Detection of Unauthorized Access Points Using various Machine Learning Algorithms. [online] norma.ncirl.ie. Available at: https://norma.ncirl.ie/4498/

[16] Sharma (2022) Research, M., Cybersecurity, K., Sharma, N. (n.d.). *Providing Network-Centric Data Security Using Machine Learning and Intrusion Detection*. [online] Available at: https://norma.ncirl.ie/6058/1/komalsharma.pdf