International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

A FUSION Features Selection for 802.11 Wireless Intruder Detection System (WIDS)

Norzaidi Baharudin[1], Fakariah Hani Mohd Ali[2,3]

[1,2] *School of Computing Sciences, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia*
[3] *Digital Forensics Research Initiative Group (RIG), College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia*

E-mail address: mail@matnet.my, fakariah@fskm.uitm.edu.my

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: In this paper, we introduce FUSION (Feature Unification via Selection, Integration, and Optimization in Networks), an innovative approach amalgamating various methods for optimal feature selection in wireless intruder detection systems. Incorporating techniques based on filters, wrappers, embedded methods, and domain knowledge, FUSION is designed to effectively pinpoint significant features in wireless networks, thereby enhancing the efficiency of intrusion detection. Our methodology initiates with a comprehensive pre-processing stage. This stage focuses on normalizing and balancing the dataset, managing missing data, and discarding irrelevant features. Beyond these pre-processing techniques, FUSION embraces a hybrid feature selection method, harnessing the advantages of filter methods, suitable for initial feature screening, wrapper methods, proficient in interaction-based selection, and embedded methods, which integrate feature selection within the model training process. A critical aspect of our evaluation includes measuring the time taken for training for each feature selection method, providing insights into the computational efficiency of the different techniques. To ensure the context's relevance throughout the selection process, we consider domain knowledge. Decision-making within FUSION is influenced by a polling weight system, aggregating the selections made by different classifiers, and prioritizing them accordingly. To verify the efficacy of our FUSION framework, we performed empirical evaluation. The results underscored a significant enhancement in intrusion detection accuracy and provided a detailed analysis of the training time, thus positioning FUSION as a promising approach to fortify network security within wireless systems.

Keywords: Wireless, 802.11, IDS, WIDS, Features Selection, Machine Learning

1. INTRODUCTION

Wireless networks, specifically 802.11 networks, are an integral component of our interconnected digital ecosystem, supporting everything from mobile devices to Internet of Things (IoT) systems. These networks utilize radio frequencies to transmit and receive data through the air, primarily by exchanging three types of frames: management frames, control frames, and data frames [1]. These frames are typically sent unencrypted and are exposed as plain text on any sniffer applications such as Wireshark, which presents potential security vulnerabilities[1] [2].

The networks interconnectivity and ease of data transmission across diverse platforms come with inherent risks. Among the most prevalent threats are unauthorized intrusions, which attempt to compromise the integrity, confidentiality, and availability of network resources. Consequently, Intrusion Detection Systems (IDS) have emerged as a critical line of defense, providing real-time monitoring and detection of potential threats within wireless networks [3].

Despite the importance of IDS, ensuring their efficiency and accuracy remains a complex task. A significant aspect of this challenge lies in the selection of pertinent features capable of accurately distinguishing between normal network behavior and potential intrusions. However, given the vast volumes of data and inherently multidimensional nature of network features, feature selection has become a challenging endeavor [4][5].

Traditional methods often lead to overfitting or underfitting of models due to the inclusion of irrelevant features or the exclusion of significant ones, underscoring the urgent need for robust and efficient feature selection techniques [6].

In this paper, we introduce FUSION (Feature Unification via Selection, Integration, and Optimization in Networks), an innovative approach designed to tackle this challenge in Wireless Intruder Detection Systems. FUSION amalgamates filter, wrapper, embedded methods, and domain knowledge-based techniques to achieve optimal feature selection. A critical aspect of our evaluation includes measuring the time taken for training for each feature selection method, providing insights into the computational efficiency of the different techniques [7] . Our methodology initiates with a meticulous pre-processing stage, recognizing that data quality significantly impacts IDS effectiveness. This stage focuses on normalizing and balancing the dataset, managing missing data, and discarding irrelevant features.

We validate the effectiveness of FUSION using the Aegean Wi-Fi Intrusion Dataset (AWID), an expansive dataset that represents a wide spectrum of wireless intrusion scenarios [8]. AWID's comprehensive and complex structure provides an ideal testing ground for FUSION, enabling its optimization under realistic conditions.

The key contributions of this paper lie in the extraction of a new subset of 10 features, resulting in high accuracy with minimum false positives for future 802.11 IDS studies. We also present a detailed analysis of the training time, thus positioning FUSION as a promising approach to fortify network security within wireless systems.

The rest of the paper is organized as follows: Section II provides an in-depth review of relevant literature. Section III introduces the AWID dataset used in our study. Our proposed FUSION methodology, detailing pre-processing, feature selection, and performance evaluation metrics, is outlined in Section IV. Section V presents the results and discusses our findings. Lastly, Section VI concludes our work, summarizing the key insights and implications of our research.

2. LITERATURE REVIEW

The field of Intrusion Detection Systems (IDS) has seen significant advancements over the years, with a plethora of studies exploring various feature selection methodologies. The challenge of high dimensionality, the need for efficient reduction techniques, and the integration of domain knowledge have emerged as central themes in the literature.

High dimensionality is a recurring concern across various authors' work [9]–[11]. Efficient reduction of dimensionality, a process that involves the removal of irrelevant or less significant features, is crucial in feature selection. This process enhances computational efficiency and accelerates the overall procedure. However, many studies, while recognizing the importance of dimensionality reduction, often fall short in providing an efficient method for its execution. This shortfall represents a significant gap in the existing body of literature, one that our proposed FUSION method aims to address by employing powerful machine learning algorithms renowned for their feature selection capabilities.

Traditional feature selection methodologies, such as filter and wrapper methods, often lack sufficient context sensitivity and fail to comprehend the intrinsic relationships among features. This can lead to suboptimal IDS performance. For example, Bhandari et al. [12] acknowledge the issue of covariate shift, a phenomenon that can drastically impede a model's predictability, yet their work does not suggest a comprehensive solution to this problem. The introduction of embedded methods, as seen in FUSION, provides a more nuanced approach, integrating feature selection within the model training process.

The work of Abdulhammed et al. [4] underscores the importance of adopting a holistic suite of evaluation metrics. While accuracy is a common choice for classifier performance evaluation, it might not provide a complete understanding, especially in scenarios involving imbalanced classes. Considering additional metrics such as precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) can provide a more in-depth evaluation. Our research aligns with this perspective, emphasizing a comprehensive evaluation that includes measuring the time taken for training for each feature selection method.

In line with our research, Aminato et al. [13] also employ weighted-feature selection for identifying impersonation attacks. They utilize methods such as ANN, SVM, and C4.5, which discern crucial features based on their respective weights, thereby enhancing the efficiency of the training process. This approach resonates with FUSION's methodology, which leverages a polling weight system to aggregate selections made by different classifiers.

Moreover, many studies tend to employ a 'blind feature selection' approach, choosing features without understanding their relevance to network intrusions fully. Such an approach can increase the noise in the dataset, obscuring meaningful intrusion-related patterns. FUSION's integration of domain knowledge aims to overcome this limitation, ensuring context relevance throughout the selection process.

On a more promising note, Zhou et al.[11] propose a novel intrusion detection system that combines feature selection and ensemble learning techniques. They introduce a heuristic algorithm, CFS-BA, for

dimensionality reduction, and an ensemble approach that synergizes C4.5, Random Forest, and Forest PA algorithms. This method exhibits superior performance in terms of accuracy, F-Measure, and Attack Detection Rate, all while maintaining a low False Alarm Rate. Despite this progress, the systematic incorporation of domain knowledge in feature selection remains largely unexplored, even though insights from domain experts can enhance the process by identifying feature relevance and importance.

The existing literature on feature selection in IDS reveals significant gaps, including effective reduction of dimensionality, handling of covariate shift, the selection of appropriate evaluation metrics, and the systematic inclusion of domain knowledge. Our proposed FUSION feature selection method aims to address these gaps, enhancing the accuracy and efficiency of intrusion detection and making a valuable contribution to the field. By amalgamating filter, wrapper, embedded methods, and domain knowledge, FUSION represents an innovative approach to fortify network security within wireless systems, reflecting the latest advancements in the field.

3. AWID DATASET

The Aegean WiFi Intrusion Dataset (AWID) is a widely recognized and publicly accessible repository of datasets that contain both normal and malicious network flows [8]. This tabular dataset encompasses 154 features, including the class feature. Four classes are represented within the dataset: "normal," and three types of network attacks - "injection," "impersonation," and "flooding."

The AWID dataset primarily includes information from the MAC layer, collected from network traces using Wireshark. The data is conveniently split into two distinct datasets, one labeled as training data (AWID-CLS-R-Trn) and the other as testing data (AWID-CLS-R-Tst). As stated in [14], the two datasets were collected at different points in time, underscoring the need for robust algorithms that can perform well despite the likely covariate shift.

The distribution of the four classes in the training and testing datasets is illustrated in Table 1. There is an evident imbalance in the data distribution: the ratio between the number of "normal" instances and "attack" instances stands at 10:1 in the training dataset and escalates to 12:1 in the testing dataset. Such imbalances can pose challenges for classification tasks, highlighting the need for the careful selection and evaluation of machine learning algorithms[14].

TABLE I. DATA DISTRIBUTION IN TERM OF TYPE OF ATTACKS

	Normal	Injection	Impers.	Flooding
AWID-R-Trn	1,633,190	65,379	48,522	48,484
AWID-R-Tst	530,785	16,682	20,079	8,097
Total	2,371,281	82,061	68,601	56,581

A. Data Preprocessing

Data preprocessing plays a pivotal role in preparing data for machine learning algorithms, ensuring the data is clean, consistent, and suitable for analysis [15]. In this research paper, we discuss various data preprocessing techniques employed on the AWID dataset. The dataset undergoes a series of steps to handle missing values, convert non-numeric values, and eliminate columns with low mean values. These techniques contribute to improving the quality and suitability of the dataset for subsequent analysis and machine learning tasks.

Initially, missing values are addressed by checking the presence of missing data using the isna().sum() function. This function calculates the sum of missing values for each column in the AWID dataset, it help us to identified columns with incomplete information. Subsequently, the replace() function is utilized to replace any "?" values with none, serving to standardize the representation of missing values across the dataset. This step ensures consistency and uniformity in handling missing data, facilitating subsequent analysis.

To further mitigate the impact of missing values, columns with a high percentage of missing values are identified. This is accomplished by computing the mean of missing values using the isnull().mean() function, which quantifies the proportion of missing values for each column. Columns with a mean exceeding 0.5 are selected as candidates for removal and stored in the null_column variable. By removing columns with a substantial amount of missing values, researchers reduce the potential bias and noise in the dataset, leading to more reliable and accurate results.

Subsequently, the drop() function is employed to eliminate the identified columns with high missing values from the AWID dataset. By specifying axis=1, columns are dropped along the horizontal axis, resulting in a more streamlined dataset for subsequent analysis. Furthermore, to ensure the dataset is devoid of any remaining missing values, the dropna() function is applied. This function removes rows containing any missing values, providing researchers with a complete and robust dataset to work with.

Moving forward, non-numeric values present in the dataset are addressed by applying the pd.to_numeric() function. This function iterates over each column in the Awid dataset, converting the values to numeric type. By specifying errors='ignore', the function disregards any errors that may arise during the conversion process. Consequently, non-numeric values remain unchanged, preserving the integrity of the dataset while transforming numeric values to a consistent format suitable for subsequent analysis.

Before the data preprocessing steps were applied, the dataset exhibited a class distribution with the following counts: 1,633,189 instances in the normal class, 48,522

instances in the impersonation class, and the injection class with the same count as the impersonation class. However, after the preprocessing techniques were implemented, the counts of instances for each class underwent changes.

For the normal class, the count decreased from 1,633,189 instances to 775,634 instances. This reduction in the count of instances belonging to the normal class suggests that the preprocessing techniques might have identified and removed instances that were potentially irrelevant or noisy. This decrease in the normal class count reflects the improved data quality resulting from the preprocessing steps, as the remaining instances are expected to be more representative and indicative of the normal class.

Similarly, the count of instances belonging to the impersonation class decreased from 48,522 to 44,731 after the preprocessing steps were applied. This reduction indicates that the preprocessing techniques were successful in identifying and eliminating instances that might have been mislabeled or incorrectly categorized as impersonation. By reducing the count of instances in the impersonation class, the preprocessing steps contribute to a more accurate representation of this specific class, enhancing the reliability of subsequent analyses and models.

Interestingly, the injection class maintained the same count as the impersonation class after the preprocessing steps. This observation suggests that the preprocessing techniques did not have a significant impact on the instances belonging to the injection class. It is possible that the preprocessing steps focused more on addressing missing values, non-numeric values, and low mean values, which may have been more prevalent in the other classes. As a result, the injection class remained relatively unchanged in terms of its count.

From this processes, we had divided the dataset into two groups. The first group consists of the last 76 features remaining after the preprocessing steps. This reduction in the number of features, from the original 154, highlights the significance of feature selection and extraction in streamlining the dataset. By eliminating 78 features, the preprocessing steps effectively removed potentially irrelevant or redundant information, enhancing the dataset's quality and efficiency.

In addition to the group of 76 features, we further refined the dataset by removing columns with a mean value less than or equal to 1. This additional step resulted in the creation of a second group, containing the last 12 features. This selection criterion, based on the mean value, allowed for the removal of features that exhibited low variation or had limited impact on the dataset's overall representation.

These two distinct groups of features provide a foundation for subsequent analysis and modeling. The dataset can be explored and analyzed using the 76 features selected through initial preprocessing. Additionally, we will investigate the impact of further dimensionality reduction by examining the last 12 features identified based on the mean value criterion. These groups of features will serve as the foundation for subsequent analyses and modelling tasks. The selection and refinement of features through the preprocessing steps are essential for ensuring the dataset's quality, reducing noise, and improving the representation of each class [16]. By considering both the importance of feature selection and the elimination of low-mean features, researchers can effectively process and analyze the dataset to extract meaningful insights and build reliable models.

Later, after we have the two groups of datasets consisting of 76 features and 12 features, we proceed to apply the SMOTE algorithm to ensure the data is balanced and reliable. By using SMOTE on both groups of features, we aim to address any remaining class imbalance issues that may persist after the preprocessing steps. SMOTE generates synthetic instances of the minority class to match the count of the majority class, thereby achieving a more balanced representation of the dataset [17]. This step further enhances the reliability of the dataset and ensures that subsequent analysis and modeling tasks are performed on a more representative and equitable dataset. The whole preprocessing process are depicted in Diagram 1 – Preprocessing Framework.

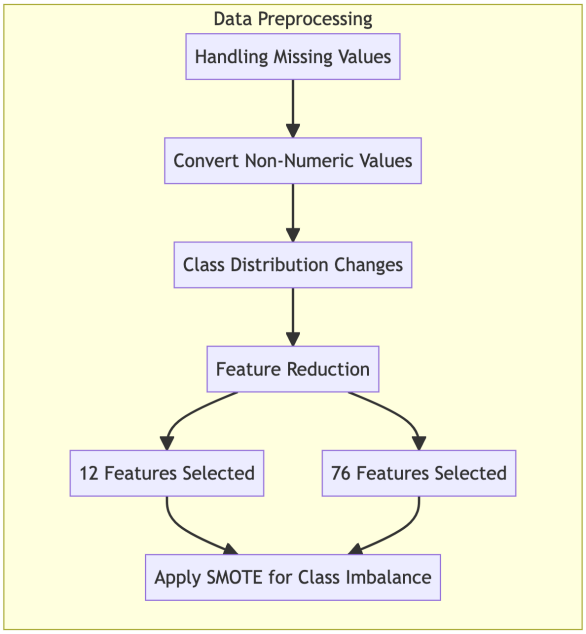


Diagram 1 – Preprocessing Framework

4. METHODOLOGY

The methodology adopted for this study consists of five key steps - preprocessing, feature selection, decision making by polling weight system, review using domain knowledge, and evaluation of performance. Initially, we

divide our dataset into two feature sets: Feature Set A with 76 features and Feature Set B with 12 features. Each set undergoes separate preprocessing. For these two groups, we apply a multifaceted approach for feature selection, encompassing filter, wrapper, and embedded methods. The covariate shift problem, as described by Glauner et al. [18], refers to the change in data distribution present in the training and the test data. This issue can significantly impact the model's performance, as the distribution that the model is trained on may differ from the distribution it is tested on.

As stated by Bhandari et al. [12], a simple solution to the covariate shift is to mix the train and test files, and based on the full dataset, generate new train or test sets that could be classified with reasonable accuracy. In our study, we adopted this approach as well. Combining the training and test datasets resulted in a total number of 2,371,281 instances. We used 20% of the entire dataset as the test set, with the remaining 80% of the data for training. To address the large imbalance problem of a 10:1 ratio, we selected a number of "normal" instances equal to the number of all "attacks." This method allowed us to create a balanced dataset, enhancing the robustness of our feature selection and intrusion detection processes.

The filter methods are applied using correlation analysis and mutual information, while the wrapper method employs Recursive Feature Elimination (RFE)[10]. Embedded methods utilize multiple machine learning algorithms, including XGBoost, CatBoost, LightGBM, Gradient Boosting, Decision Trees, Random Forest, and AdaBoost, along with various algorithms from the sklearn library [19]. Renowned for their feature selection capabilities, these algorithms assist in efficient dimensionality reduction and help select the 10 most important features from each group. This combination of methods forms an integral part of our methodology, ensuring the accuracy and efficiency of the Intrusion Detection System (IDS).

The data is split into training and test sets, and a dictionary of models is created, including DecisionTree, XGBoost, CatBoost, AdaBoost, GradientBoosting, RandomForest, and LightGBM. Each model is trained and evaluated, with measurements taken for training time, prediction time, accuracy, precision, recall, F1 score, and others. Confusion matrices are generated and visualized using heatmaps. Once the feature selection process is complete for both sets, we use a polling weight system for decision-making. This process involves aggregating the selections made by different classifiers and prioritizing them based on their weighted votes. Subsequently, we conduct a review using domain knowledge to ensure that our feature importance list is comprehensive and precise.

Once the feature selection process is complete for both sets, we use a polling weight system for decision-making. This process involves aggregating the selections made by different classifiers and prioritizing them based on their weighted votes [20]. This helps us arrive at a more robust and comprehensive list of features. The methodology of FUSION framework depicted in Diagram 2.

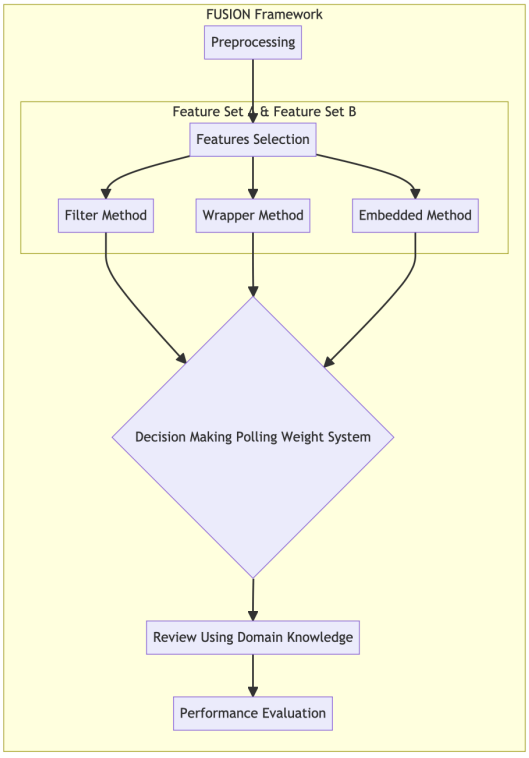


Diagram 2 – FUSION Framework

We would like to obtained 10 feature importance results for each of the 8 machine learning algorithms applied to both 12-feature set (FS) and 76-feature set. The objective is to identify the top 10 features from these combined results. The formula to calculate the total weighted importance for each feature would be:

$$S_j = \sum_i (W_i * F_{ij})$$

- F_{ij} as the importance of the j-th feature as calculated by the i-th machine learning algorithm
- W_i as the weight of the i-th machine learning algorithm
- S_j as the total weighted importance of the j^{th} feature

Subsequently, we conduct a review using domain knowledge to ensure that our feature importance list is comprehensive and precise. During this review, we examine the features that did not make it into the list of the top 10 important features from each group. For instance, we consider the feature selection mentioned by Amoordon et al. [1], where features such as frame interval, Received Signal Strength, Sequence number gap, and subtype were identified as significant for detecting impersonation attacks. If any feature, based on our domain knowledge, appears to be important for our IDS, we manually include it in our list. This additional step helps us ensure that our

feature selection process is not only data-driven but also guided by expert knowledge.

Finally, we arrive at our list of final features. Using these features, we then evaluate our IDS. We apply the machine learning algorithms used earlier for feature selection and measure the performance using a variety of metrics, including accuracy, precision, recall, F1 score, and others. Additionally, we also measure the time taken for training during each feature selection process, providing a comprehensive understanding of computational efficiency [7]. This final step enables us to determine how effectively our feature selection method contributes to the performance of our IDS and provides insights into its practical applications in wireless network security.

5. Result And Discussions

Prior to conducting our experiments, we applied the SMOTE (Synthetic Minority Over-sampling Technique) algorithm to address class imbalance in our dataset. The initial class distribution revealed a significant disparity, with "normal" having a count of 775,634, while "injection" and "impersonation" had much lower counts of 65,379 and 44,731, respectively. This imbalance could potentially introduce bias in our machine learning models during training.

After implementing SMOTE, the class distribution underwent a notable transformation, resulting in a balanced dataset. The count for the majority class, "normal," remained unchanged at 775,634. Meanwhile, both "injection" and "impersonation" were adjusted to match the count of the majority class, resulting in a count of 775,634 for each class. The primary objective was to alleviate the bias stemming from imbalanced data and improve the model's capacity to effectively classify instances from the minority classes.

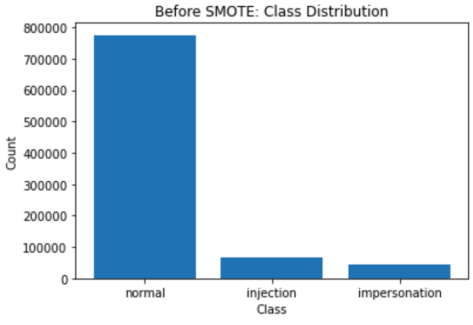


Figure 1 – Before SMOTE

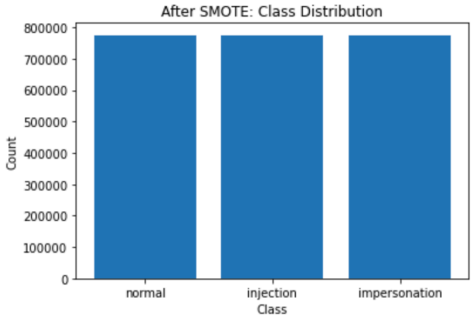


Figure 2: After SMOTE

The process of applying SMOTE introduced synthetic samples to augment the representation of the minority classes. While this technique helps prevent favouritism towards the majority class, it is important to consider the implications of these synthetic samples. They may not fully encapsulate the true distribution and characteristics of the minority classes, necessitating a careful evaluation of their impact on the overall performance of our models.

By attaining a balanced class distribution, our dataset becomes more conducive to training machine learning models. The enhanced representation of the minority classes enhances the model's ability to discern and generalize patterns from these classes, ultimately leading to more accurate predictions. Nonetheless, it remains crucial to evaluate the performance of the models using appropriate metrics and validate their efficacy in real-world scenarios to ensure robust and reliable outcomes.

In this study, we conducted a series of several experiments aimed at optimizing the feature selection for an Intrusion Detection System (IDS). The first series of experiments include the following three experiments:

i. Experiment 1: Running Set A (76-feature set) through filter, wrapper and embedded method using five Machine Learning (ML) classifiers both before and after applying Synthetic Minority Over-sampling Technique (SMOTE).
ii. Experiment 2: Running Set B (12-feature set) with eight ML classifiers, similar to Experiment 1, both pre- and post-SMOTE.
iii. Experiment 3: After determine the final feature selection through a polling weight system, we ran the selected features through the eight ML classifiers again to measure various performance metrics, including accuracy, precision, F1 score, recall, and training time taken.

Tables 2 through 6 display the results derived from the application of various machine learning algorithms for feature selection within Set A and Set B, post the Synthetic Minority Over-sampling Technique (SMOTE) process:

TABLE II. FEATURES IMPORTANCE USING CATBOOST

	Features Set A	Features Set B
1	wlan.duration	wlan.duration
2	wlan.seq	wlan.fc.subtype
3	frame.len	wlan.seq
4	frame.time_relative	radiotap.mactime
5	frame.cap_len	frame.time_epoch
6	radiotap.mactime	frame.time_relative
7	frame.time_epoch	frame.len
8	data.len	frame.cap_len
9	wlan.fc.pwrmgt	wlan.fc.type
10	frame.time_delta	radiotap.datarate

TABLE III. FEATURES IMPORTANT USING LIGHTGBM

	Features Set A	Features Set B
1	frame.time_epoch	frame.time_epoch
2	wlan.seq	frame.len
3	frame.len	wlan.seq
4	frame.time_delta	wlan.fc.subtype
5	wlan.duration	wlan.duration
6	radiotap.dbm_antsignal	wlan.fc.type
7	radiotap.datarate	frame.time_relative
8	wlan.wep.key	radiotap.channel.freq
9	radiotap.channel.freq	radiotap.datarate
10	wlan.frag	radiotap.mactime

TABLE IV. FEATURES IMPORTANT USING XGB

	Features Set A	Features Set B
1	wlan.duration	wlan.fc.type
2	frame.len	wlan.duration
3	frame.time_delta	frame.len
4	frame.time_relative	wlan.sequence
5	wlan.frag	radiotap.datarate
6	wlan.fc.frag	wlan.fc.subtype
7	frame.time_epoch	frame.time_epoch
8	wlan.fc.moredata	radiotap.mactime
9	wlan.fc.subtype	radiotap.channel.freq
10	wlan.wep.key	frame.time_relative

TABLE V. FEATURES IMPORTANT USING GRADIENTBOOSTING

	Features Set A	Features Set B
1	wlan.duration	wlan.seq
2	frame.cap_len	frame.len
3	frame.len	frame.cap_len
4	frame.time_relative	wlan.duration
5	frame.time_delta_displayed	wlan.fc.type
6	radiotap.mactime	frame.time_epoch
7	frame.time_delta	wlan.fc.subtype
8	wlan.frag	frame.time_relative
9	data.len	radiotap.datarate
10	wlan.seq	radiotap.mactime

TABLE VI. FEATURES IMPORTANT USING DECISION TREE

	Features Set A	Features Set B
1	wlan.duration	wlan.duration
2	frame.cap_len	wlan.fc.type
3	frame.time_relative	frame.len
4	wlan.fc.subtype	radiotap.mactime
5	frame.time_epoch	wlan.seq
6	radiotap.channel.freq	wlan.fc.type
7	data.len	wlan.fc.subtype
8	wlan.seq	frame.cap_len
9	radiotap.channel.type.2ghz	frame.time_relative
10	radiotap.channel.type.gsm	radiotap.datarate

TABLE VII. FEATURES IMPORTANT USING RANDOM FOREST

	Features Set A	Features Set B
1	wlan.duration	wlan.duration
2	wlan.seq	frame.cap_len
3	frame.cap_len	wlan.fc.subtype
4	frame.len	frame.len
5	radiotap.datarate	wlan.seq
6	radiotap.mactime	frame.time_relative
7	data.len	wlan.fc.type
8	frame.time_relative	frame.time_epoch
9	frame.time_epoch	radiotap.datarate
10	wlan.fc.subtype	radiotap.mactime

TABLE VIII. FEATURES IMPORTANT USING SELECTKBEST

	Features Set A	Features Set B
1	frame.time_epoch	frame.time_epoch
2	frame.time_delta	frame.time_relative
3	frame.time_delta_displayed	radiotap.mactime
4	frame.time_relative	wlan.seq
5	frame.len	frame.len
6	frame.cap_len	frame.cap_len
7	radiotap.mactime	wlan.fc.subtype
8	radiotap.datarate	wlan.duration
9	radiotap.channel.type.cck	wlan.fc.type
10	wlan.duration	radiotap.datarate

The results from the three experiments can be summarized as follows:

- Experiment 1: There were only slight variations in the results when the 76-feature set was run through the eight ML classifiers before and after SMOTE. The top 10 features identified as most significant in this experiment are: wlan.duration, frame.len, frame.time_relative, frame.cap_len, wlan.seq,frame.time_epoch,radiotap.mactime, data.len, frame.time_delta, wlan.fc.subtype.

- Experiment 2: In a similar fashion to Experiment 1, the 12-feature set was run through the eight ML classifiers both before and after SMOTE. The results were largely identical with some minor differences. The top 10 features determined from this experiment include: wlan.duration, wlan.fc.subtype, frame.len, wlan.seq, frame.cap_len, frame.time_epoch, wlan.fc.type,frame.time_relative,radiotap. mactime, radiotap.datarate.

- Experiment 3: The final feature selection was determined using a weighted voting system combined with domain expertise. The top features identified through this approach are: wlan.duration, frame.len, frame.time_relative, frame.cap_len, wlan.seq, frame.time_epoch, radiotap.mactime, wlan.fc.type, wlan.fc.subtype, and radiotap.datarate.

Comparing these lists, we see that there is a significant overlap in the features selected from both sets. Namely, the features **'wlan.duration', 'frame.len', 'frame.time_relative', 'frame.cap_len', 'wlan.seq', 'frame.time_epoch', and 'radiotap.mactime'** appear in both lists. This suggests that these features carry significant importance across different configurations of features and machine learning algorithms.

However, the selection process didn't stop there. Our next step was to bring in domain knowledge to supplement and enrich our initial feature list. In light of the importance of some features demonstrated in previous research and their observed significance within our dataset, we decided to add 'data.len', 'wlan.fc.retry', 'wlan.fc.subtype', 'wlan.fc.type', and 'radiotap.datarate' to our list of top features.

The 'data.len' feature was of particular Interest due to Its frequent appearance within our Set A feature list, and its previous citation in Abdulhammeed et al. [21] research as a key feature for effective feature selection. An exploration of our dataset revealed that 'data.len' has a diverse range of 1309 distinct values. However, it's worth noting the presence of a large number of NaN or "?" values, which appear 903,020 times in the dataset.

The features 'wlan.fc.type' and 'wlan.fc.subtype' represent specific packet configurations and thus have high relevance for intrusion detection. The 'wlan.fc.type' feature indicates whether a packet belongs to management frames, data frames, or control frames. The 'wlan.fc.subtype' goes further in detail, distinguishing between different subtypes within these categories, such as association requests, probe requests, beacon messages, and so on, as mentioned by Abdulhameed et al. [4]. These categorizations allow a more refined analysis of the traffic and significantly aid in identifying unusual patterns.

While the 'wlan.fc.retry' feature was pointed out by Kolias et al. [9] as a potential indicator of flooding attacks, our examination of the dataset indicated that it appeared only once in the non-normal class, which would not offer much contribution to our intrusion detection system. Finally, the 'radiotap.datarate' feature also made it to our list. It was highlighted ten times in our Set A feature list, and our data analysis confirmed its richness for further examination.

Consequently, after careful consideration and the incorporation of domain knowledge, our final list of top features includes 'wlan.fc.type', 'wlan.fc.subtype', and 'radiotap.datarate' in addition to the previously identified seven.

6. EVALUATION AND PERFORMANCE

Upon deriving our top 10 features – 'wlan.duration', 'frame.len', 'frame.time_relative', 'frame.cap_len', 'wlan.seq', 'frame.time_epoch', 'radiotap.mactime', 'wlan.fc.type', 'wlan.fc.subtype', and 'radiotap.datarate', we advance to the performance evaluation and assessment phase of our methodology.

In this critical phase, we utilize our selected features and various machine learning algorithms to assess the effectiveness of these features in intrusion detection. The algorithms implemented for feature selection, which include but are not limited to XGBoost, CatBoost, LightGBM, Gradient Boosting, Decision Trees, Random Forest, and AdaBoost, will also be put to use in this stage. To gauge the performance of our Intrusion Detection System (IDS), we rely on a range of evaluation metrics that includes accuracy, precision, recall, F1 score, and training time, among others. Each of these metrics lends a unique perspective to the IDS's performance [9]:

- **Accuracy** gauges the fraction of total predictions that are accurate, factoring in both positive and negative predictions. This metric is essential for understanding the overall effectiveness of the IDS in classifying network traffic correctly.
- **Precision** evaluates the fraction of positive predictions that are indeed correct. This metric reflects the IDS's competence in minimizing false alarms, thereby reducing the administrative overhead associated with investigating false positives.
- **Recall (or Sensitivity)** measures the system's prowess at correctly identifying positive (intrusive) instances. This is pivotal in averting potential security threats, as a high recall rate ensures that fewer actual intrusions go undetected.
- **F1 Score** offers a balanced measure that takes both precision and recall into account. This metric is especially valuable when the classes namely, intrusive and non-intrusive activities are unequally distributed, as it provides a more holistic view of the system's performance.
- **Training Time** assesses the duration required for the IDS to become operational. In the context of rapidly evolving cyber threats, a shorter training time is advantageous as it allows the IDS to adapt more quickly to new types of attacks. This metric is particularly crucial in environments with limited computational resources, where efficiency is a key concern.

By including training time as a key metric in our evaluation criteria, our objective is to offer a fuller picture of the IDS's functional efficiency and its ability to adapt, alongside its skill in classifying network activities.

Furthermore, we employ a Confusion Matrix as a tool to visualize the classifier's performance. This matrix outlines the results in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), thereby providing an all-encompassing view of the classifier's effectiveness in accurate identification and categorization [22].

TABLE IX. TRAINING OF TIME WITH VARIOUS MODEL AND FEATURES SELECTION

Model	Without FS (Feature Selection) in (s)	FUSION FS (s)	RF FS (s)	Lasso FS (s)	RFE FS(s)
DT	0.85	**0.61**	0.77	1.07	0.45
Xgboost	21.89	**9.88**	9.92	32.15	21.95
CatBoost	20.69	**17.9**	19.43	28.08	24.23
AdaBoost	16.25	**9.11**	12.42	12.22	9.91
GradientBoosting	179.45	**116.6**	169.05	211.71	105.05
RF	13.48	**13.74**	15.75	26.14	12.19
LightGBM	1.73	**1.18**	1.35	8.49	1.11

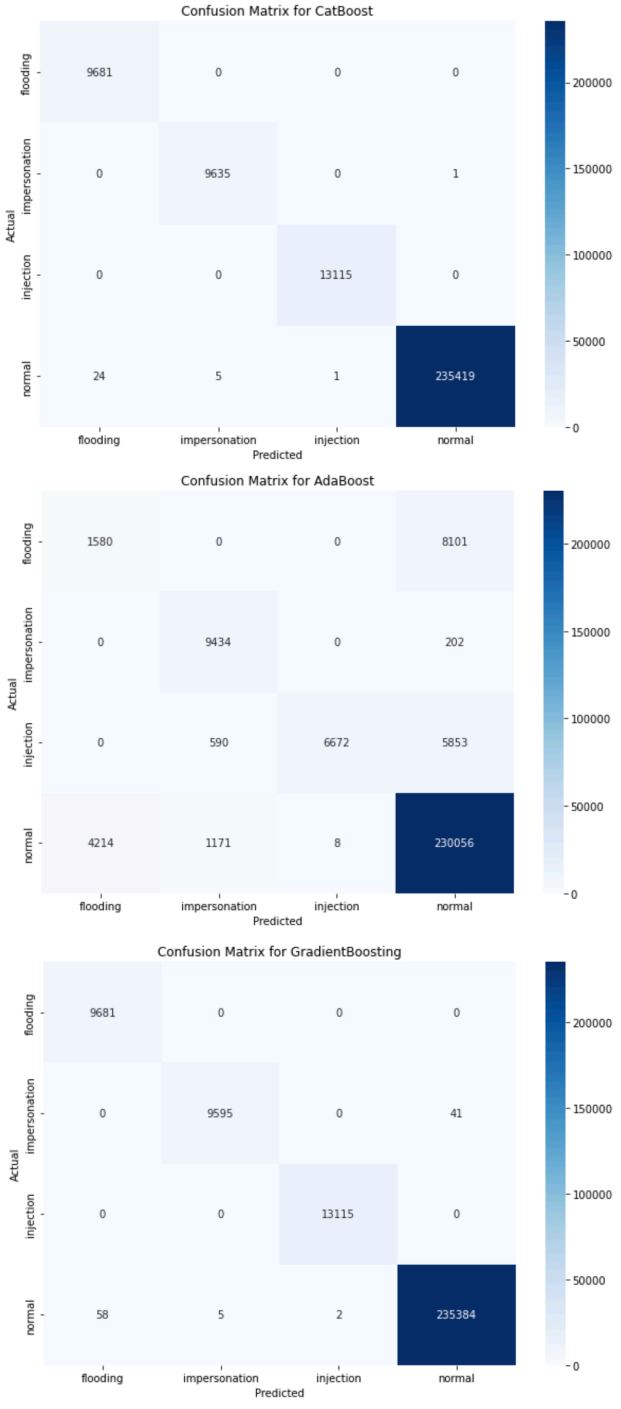
Through this extensive evaluation, our objective extends beyond merely determining the performance of our IDS. We aim to comprehend how our feature selection methodology contributes to this performance effectively. This understanding will shed light on the prospective practical applications of our feature selection process within the realm of wireless network security. This performance and evaluation phase symbolizes the final stage in our FUSION Framework, bringing our research on effective feature selection for intrusion detection to a conclusion.

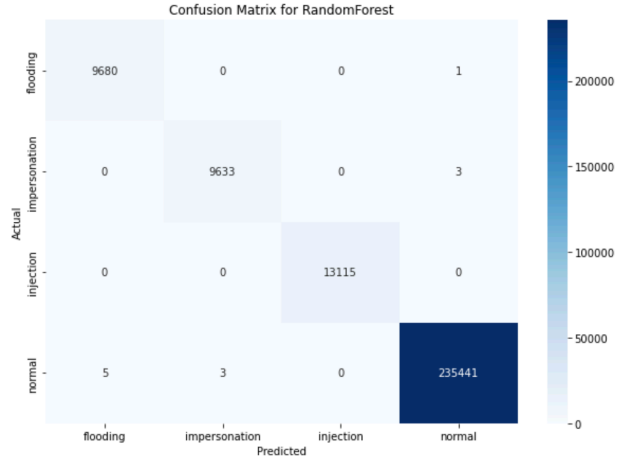
The outcomes of our experiments Indicate outstanding performance in all primary metrics, including accuracy, precision, F1 score, and recall, each surpassing a value of 0.9999. In light of these near-flawless results, our subsequent evaluation will emphasize the comparison of training times across different algorithms at Table IX. This will enable us to discern which algorithm is not only superior in terms of classification but also most efficient in training time. Additionally, for a more nuanced understanding of the classifier's performance, we plan to visualize the results using a heat map of the Confusion Matrix. This multi-faceted approach aims to provide a thorough assessment of the algorithm's applicability in real-world IDS scenarios.

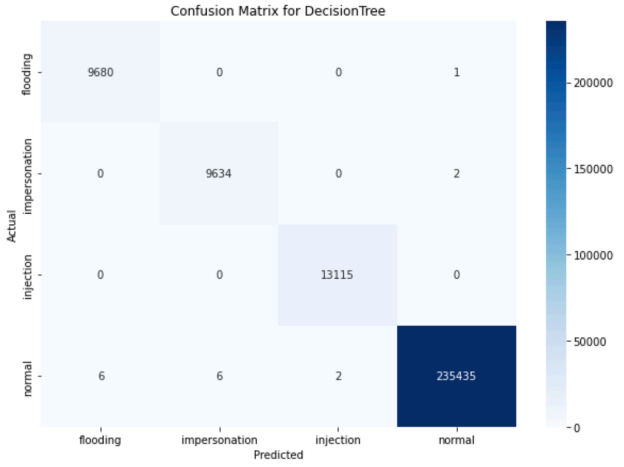
From the Table IX, the evaluation encompasses a range of machine learning algorithms such as Decision Trees (DT), XGBoost, CatBoost, AdaBoost, Gradient Boosting, Random Forest (RF), and LightGBM. These algorithms were tested with and without feature selection methods, including FUSION, Random Forest-based Feature Selection (RF FS), Lasso-based Feature Selection (Lasso FS) and Recursive Feature Elimination (RFE) Feature Selection. Our novelty FUSION feature selection method generally demonstrates a substantial reduction in training time across a variety of algorithms, with a notable exception in RFE which is with Decision Tree Model. As traditional performance metrics like accuracy and precision reach saturation, training time emerges as a pivotal metric for evaluating IDS performance. The FUSION method, therefore, represents a promising avenue for enhancing IDS efficiency, particularly in scenarios requiring rapid adaptation to emerging threats.

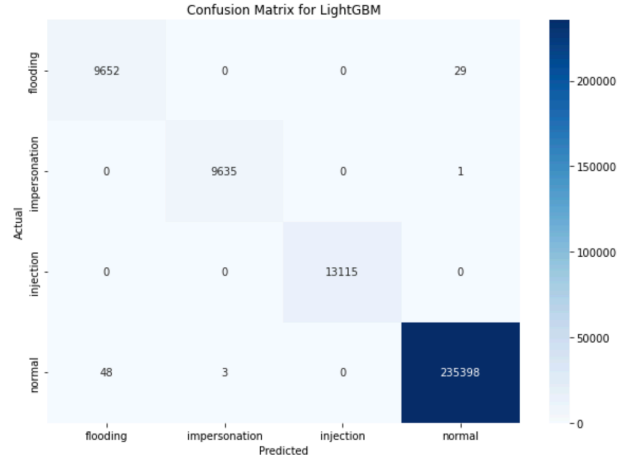
Figures 3-7 display the Confusion Matrix visualizations for all evaluated models, including Decision Trees (DT), CatBoost, XGBoost, Gradient Boosting, AdaBoost, Random Forest, and LightGBM. These matrices reveal consistently high True Positive (TP) values across all models, underscoring their effectiveness in correctly identifying intrusive activities.

Figure 3-7: Confusion Matrix









7. CONCLUSION AND FUTURE DIRECTION

In this research, we introduced FUSION (Feature Unification via Selection, Integration, and Optimization in Networks), a novel feature selection framework specifically tailored for wireless intrusion detection systems. FUSION amalgamates a variety of techniques, including filters, wrappers, embedded methods, and domain-specific knowledge, to identify the most pertinent features for intrusion detection. The framework begins with an exhaustive pre-processing stage that normalizes and balances the dataset, manages missing data, and eliminates irrelevant features.

Subsequently, FUSION employs a hybrid approach to feature selection, leveraging the strengths of various methods to optimize both the feature set and the computational efficiency of the model. Our empirical evaluation revealed that FUSION not only significantly improves intrusion detection accuracy but also reduces training time across multiple machine learning algorithms.

This is particularly noteworthy given that traditional performance metrics like accuracy, precision, and recall have reached near-perfect scores, making training time an increasingly critical metric for real-world deployments. The only exception was observed in the RFE algorithm, where FUSION slightly increased the training time. This anomaly suggests that the effectiveness of FUSION may vary depending on the specific characteristics of the machine learning algorithm employed, warranting further investigation.

As we move forward, several avenues for future research emerge. First, the adaptability of FUSION to other types of networks beyond wireless systems could be explored. Second, the framework could be extended to incorporate

real-time feature selection, which would be invaluable in dynamic environments where threats evolve rapidly. Third, the integration of more advanced machine learning techniques or ensemble methods could further optimize both the feature selection process and the overall performance of the intrusion detection system.

In summary, FUSION represents a significant advancement in the field of intrusion detection, offering a comprehensive, efficient, and adaptable feature selection method. Its ability to reduce training time while maintaining high accuracy positions it as a promising solution for enhancing network security, especially in scenarios that demand quick adaptability to emerging threats.

ACKNOWLEDGMENT

Authors acknowledge the College of Computing, Informatics and Mathematics for funding the publication.

REFERENCES

[1] A. Amoordon, V. Deniau, A. Fleury, and C. Gransart, "A single supervised learning model to detect fake access points, frequency sweeping jamming and deauthentication attacks in IEEE 802.11 networks," *Machine Learning with Applications*, vol. 10, p. 100389, Dec. 2022, doi: 10.1016/j.mlwa.2022.100389.

[2] N. Baharudin, F. H. M. Ali, M. Y. Darus, and N. Awang, "Wireless intruder detection system (WIDS) in detecting de-authentication and disassociation attacks in IEEE 802.11," in *2015 5th International Conference on IT Convergence and Security, ICITCS 2015 - Proceedings*, 2015. doi: 10.1109/ICITCS.2015.7293037.

[3] P. Satam and S. Hariri, "WIDS: An Anomaly Based Intrusion Detection System for Wi-Fi (IEEE 802.11) Protocol," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 1077–1091, Mar. 2021, doi: 10.1109/TNSM.2020.3036138.

[4] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Alessa, "Effective Features Selection and Machine Learning Classifiers for Improved Wireless Intrusion Detection," *2018 International Symposium on Networks, Computers and Communications, ISNCC 2018*, pp. 1–6, 2018, doi: 10.1109/ISNCC.2018.8530969.

[5] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Comput Secur*, vol. 92, May 2020, doi: 10.1016/j.cose.2020.101752.

[6] Md. A. Talukder *et al.*, "A Dependable Hybrid Machine Learning Model for Network Intrusion Detection," Dec. 2022, doi: 10.1016/j.jisa.2022.103405.

[7] A. A. Reyes, F. D. Vaca, G. A. C. Aguayo, Q. Niyaz, and V. Devabhaktuni, "A machine learning based two-stage wi-fi network intrusion detection system," *Electronics (Switzerland)*, vol. 9, no. 10, pp. 1–18, Oct. 2020, doi: 10.3390/electronics9101689.

[8] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 184–208, 2016, doi: 10.1109/COMST.2015.2402161.

[9] E. Chatzoglou, G. Kambourakis, C. Kolias, and C. Smiliotopoulos, "Pick Quality Over Quantity: Expert Feature Selection and Data Preprocessing for 802.11 Intrusion Detection Systems," *IEEE Access*, vol. 10, pp. 64761–64784, 2022, doi: 10.1109/ACCESS.2022.3183597.

[10] T. Gaber, A. El-Ghamry, and A. E. Hassanien, "Injection attack detection using machine learning for smart IoT applications," *Physical Communication*, vol. 52, Jun. 2022, doi: 10.1016/j.phycom.2022.101685.

[11] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.

[12] S. Bhandari, A. K. Kukreja, A. Lazar, A. Sim, and K. Wu, "Feature Selection Improves Tree-based Classification for Wireless Intrusion Detection," in *SNTA 2020 - Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, Association for Computing Machinery, Inc, Jun. 2020, pp. 19–26. doi: 10.1145/3391812.3396274.

[13] M. Erza Aminanto, P. D. Yoo, H. Chandra Tanuwidjaja, and K. Kim, "Weighted Feature Selection Techniques for Detecting Impersonation Attack in Wi-Fi Networks."

[14] Z. G. R and K. C, "Survey on Machine Learning Approaches for Intrusion Detection System," European Alliance for Innovation n.o., Jan. 2022. doi: 10.4108/eai.7-12-2021.2315107.

[15] M. Natkaniec and M. Bednarz, "Wireless Local Area Networks Threat Detection Using 1D-CNN," *Sensors*, vol. 23, no. 12, p. 5507, Jun. 2023, doi: 10.3390/s23125507.

[16] J. Liu and S. S. Chung, "Automatic feature extraction and selection for machine learning based intrusion detection," in *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, Institute of Electrical and Electronics Engineers Inc., Aug. 2019, pp. 1400–1405. doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00254.

[17] J. Chen, T. Yang, B. He, and L. He, "An analysis and research on wireless network security dataset," in *Proceedings - 2021 International Conference on Big Data Analysis and Computer Science, BDACS 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021, pp. 80–83. doi: 10.1109/BDACS53596.2021.00025.

[18] P. Glauner, P. Valtchev, and R. State, "Impact of Biases in Big Data," Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.00897

[19] G. Granato, A. Martino, L. Baldini, and A. Rizzi, "Intrusion detection in Wi-Fi networks by modular and optimized ensemble of classifiers," in *IJCCI 2020 - Proceedings of the 12th International Joint Conference on Computational Intelligence*, SciTePress, 2020, pp. 412–422. doi: 10.5220/0010109604120422.

[20] M. Erza Aminanto, P. D. Yoo, H. Chandra Tanuwidjaja, and K. Kim, "Weighted Feature Selection Techniques for Detecting Impersonation Attack in Wi-Fi Networks."

[21] R. Abdulhammed, "Intrusion Detection: Embedded Software Machine Learning and Hardware Rules Based Co-Designs," p. 176, 2019.

[22] J. W. Mikhail, J. M. Fossaceca, and R. Iammartino, "A semi-boosted nested model with sensitivity-based weighted binarization for multi-domain network intrusion detection," *ACM Trans Intell Syst Technol*, vol. 10, no. 3, 2019, doi: 10.1145/3313778.

Norzaidi Baharudin is an Enforcement Officer in Malaysia, where he plays an essential role in maintaining law and order. His professional interests are deeply anchored in cybersecurity, with a specialized focus on wireless 802.11 technologies. Currently, Norzaidi is a Ph.D. candidate at Universiti Teknologi Mara. His research centers on 802.11 Management Frames for Intrusion Detection Systems (IDS), aiming to address the pressing security challenges in wireless networks. This work contributes to the development of more secure and resilient systems. In addition to his research endeavors, Norzaidi serves as a part-time lecturer, covering a broad spectrum of topics such as cybersecurity, networking, and Artificial Intelligence.

Dr. Fakariah Hani Hj Mohd Ali is a distinguished Senior Lecturer at Universiti Teknologi MARA (UiTM), specializing in the fields of cryptography, digital forensics, and network security. She holds a Ph.D. in Security in Computing and an MSc in Networking, both from Universiti Putra Malaysia (UPM). Additionally, she earned her BSc (Hons) and Diploma in Computer Science from UiTM. With a research portfolio that includes 21 published works and 237 citations, Dr. Fakariah is a recognized authority in cybersecurity. Her research has garnered attention for its focus on machine learning applications in malware detection.