Bulletin of Business and Economics, 13(2), 391-403
https://bbejournal.com
https://doi.org/10.61506/01.00343

Analysis of Machine Learning Methods for Intrusion Detection Systems in Wireless Networks

Muhammad Faseeh Sultan[1], Sammia Hira[2], Sohail Masood[3], Allah Rakha[4]

Abstract

Wireless networks have become integral to modern communication systems, making them vulnerable to various security threats. Intrusion detection systems (IDSs) are essential for detecting and mitigating these threats and can also powerfully screen network traffic for pernicious activities that are planned to abuse the classification, honesty, realness, and accessibility of the network. Machine learning (ML) and Deep learning (DL) techniques are effective in identifying and classifying network attacks. This study proposes a novel intrusion detection system that employs ML and DL models to classify and distinguish network attacks in wireless networks. The proposed system enhanced detection accuracy and efficiency in IDS, the scalability of IDS systems, and estimated the performance of IDS in wireless networks. It also investigates IDS techniques using machine learning, designs and implements IDS in wireless networks using machine learning, and trains several IDS models regarding wireless networks that are fitted. It contrasts the exhibition of proposed models and existing procedures. The suggested system can therefore be utilized as an effective IDS for wireless networks, providing real-time detection and classification of network attacks.

Keywords: wireless networks, machine learning, deep learning, classification, intrusion detection system

1. Introduction

Wireless networks have turned into a basic part of modern communication systems, allowing people to connect and access information from virtually anywhere, without the need for physical cables or wires. These networks are built on the principle of wireless communication, using radio waves to transmit data between devices. Wireless networks have been persuaded in the last many years, to figure out how to agitate the strength of the wired ones (Kolias, C et al., 2016a). The development of wireless networks began in the early 1990s, with the introduction of the first wireless standard, IEEE 802.11. This standard, usually known as Wi-Fi, provided a way for devices to communicate wirelessly within a limited range, typically within a single room or building. As wireless technology evolved, new standards and protocols were introduced to support faster and more reliable wireless communication. For example, the introduction of 802.11a and 802.11g standards allowed for faster data transmission rates and wider coverage areas. The Wi-Fi network standards are considered to be the major traffic in internet traffic (Kolias, C et al., 2016). The updated standard, 802.11ax (Wi-Fi 6), offers even higher data transfer speeds, lower latency, and increased network capacity. Wireless networks have revolutionized how people communicate and access information, making it possible for businesses to operate more efficiently, for people to work remotely, and for individuals to stay connected with friends and family worldwide. However, wireless networks also present unique security challenges, including the risk of eavesdropping, data theft, intrusion, and unauthorized access.

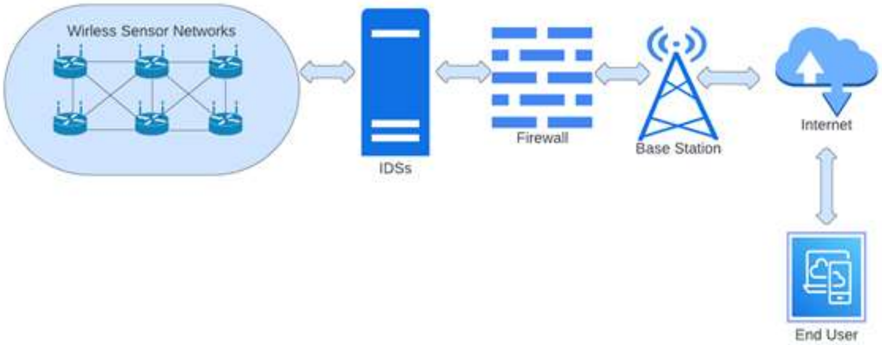


Figure 1: Intrusion detection system (IDS) architecture

To address these concerns, security protocols such as WPA2 and WPA3 have been developed and implemented in wireless networks (Sagers, G. 2021)(V. Srikanth, & I. Reddy, 2019), along with the use of Intrusion Detection Systems (IDS) and other security measures. As wireless networking became more popular in the 1990s, IDSs were adapted to work with these types of networks as well. In wireless networks, Intrusion detection frameworks are one of the ongoing research topics in recent times. Wireless IDSs (WIDSs) were first developed in the late 1990s, around the same time as the first wireless networks were being deployed. These early WIDSs were based on the same principles as traditional IDSs but were adapted to work with the unique characteristics of wireless networks (Kumar, Sandeep, & Spafford, Eugene, 1999). An interruption is essentially any sort of illegal behavior performed by assailants to damage network assets or sensor hubs. For intrusion detection, huge research has been conducted such as intrusion detection in wireless networks utilizing clustering methods (T. M. Khoshgoftaar, et al., 2005) and developed many wireless intrusion prevention systems (Luo, Q, 2010). However, these systems have limitations such as a lack of excellent architecture, a high false positive rate, and problems in the detection of positive nostrils in the network. Therefore, an intrusion detection system is a structure for detecting such illegal or harmful undertakings (Djenouri, D.et al., 2005). The main function of IDS is to observe user behavior and network behavior in wireless networks. Wireless IDSs work by analyzing network traffic and looking for patterns that match

[1] Corresponding Author, Faculty of Computing, Department of Computer Science and Information Technology, The Superior University, Lahore, 54000, Pakistan, faseehsultan70@gmail.com
[2] Faculty of Computing, Department of Computer Science and Information Technology, The Superior University, Lahore, 54000, Pakistan
[3] Faculty of Computing, Department of Computer Science and Information Technology, The Superior University, Lahore, 54000, Pakistan
[4] Faculty of Computing, Department of Computer Science and Information Technology, The Superior University, Lahore, 54000, Pakistan

known attack signatures (Mukherjee, B, et al., 1994). They can also use behavior-based analysis to detect abnormal network behavior, such as a sudden surge in traffic or an unusual number of failed login attempts. IDSs can be implemented at various locations within a wireless network, including at the network perimeter, on individual wireless access points, or individual devices. The development of IDSs for wireless networks has been driven by the need for improved security and the desire to prevent attacks before they can cause damage (Loo, C. E, et al, 2006).

1.1. Motivation of Study

Over time, WIDSs became more sophisticated and were able to detect a wider range of security attacks and threats. Security attacks can come in various forms, including network-based attacks, application-based, and host-based attacks. Network-based attacks include port scanning, denial-of-service (DoS) attacks, and network reconnaissance. Application-based attacks include SQL injection, cross-site scripting (XSS), and buffer overflow attacks. Host-based attacks include malware infections, rootkit installation, and unauthorized changes to system files or configuration settings. In addition, as evidenced by the DDoS (Distributed Denial-of-Services) attacks that been sent off against large web destinations where safety efforts are set up, the protocols and frameworks that are intended to offer types of assistance (to general society) are intrinsically exposed to attacks like DDoS (Khan, S., et al., 2008). Intrusion detection could utilized as a second wall to secure network frameworks (Bace, R., & Mell, P. 2001) as once a breach is exposed, such as in the previous stages of DDoS assault, feedback could initiated to decrease damage, accumulate proof for indictment, and, surprisingly counter-send-off assaults. Intrusion detection systems can help prevent security attacks by identifying and alerting security personnel to suspicious or anomalous activity. Once an IDS detects a potential security attack, it can trigger a response mechanism, such as shutting down a network segment, blocking traffic, or alerting security personnel. This can help reduce the risk of a successful security attack and minimize the impact of a security breach. Moreover, intrusion detection systems are an essential component of any organization's security architecture, as they provide a critical layer of defense against a wide range of security attacks (Mudzingwa, D., & Agrawal, R. 2012). By continuously monitoring network traffic and system activity, IDSs can help guarantee the probity, confidentiality, and accessibility of critical information and systems. WIDSs are a critical part of many wireless network security systems. They help to ensure the integrity of wireless networks and protect against a wide range of security threats. As wireless networking continues to evolve, WIDSs will continue to assume a significant part in ensuring the security and reliability of these networks. Lately, there has been a growing emphasis on the utilization of machine learning and artificial intelligence techniques to enhance IDS performance and accuracy, allowing for more efficient detection and prevention of attacks. Huge research on intrusion detection systems in ad-hoc networks, wireless sensor networks (Roman, R., et al, 2006), WLANS, WMN, and IOT-WSN etc. has been conducted utilizing different machine learning and deep learning models (Abdulhammed, R., et al., 2019)(Javaid, A., et al., 2016)(Chen, A. C. H., et al., 2022).

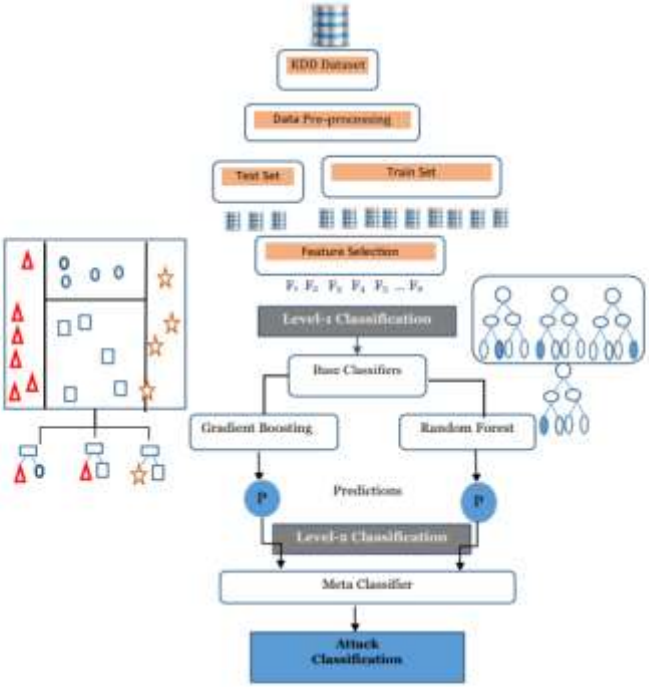


Figure 2: Proposed system architecture by (Chen, A. C. H., et al., 2022)

A number of concerns and difficulties with using traditional machine learning techniques in wireless networks (Pushpender Sarao. 2019). It is difficult to design the machine learning foundation routing method in wireless links. Preparing a data set and training it is a particularly challenging operation because of the dynamic network circumstances in many ad-hoc networks. Moreover, the advantages of machine learning-based IDS offers several advantages over traditional rule-based IDS, including handling complex and dynamic data, detecting unknown and zero-day attacks, and reducing false positives and negatives (Yalin E. Sagduyu, et al 2020). However, it also presents some challenges, such as the need for large and diverse datasets, the selection of appropriate ML algorithms, and the potential for adversarial attacks (Zamani, M., & Movahedi, M. 2013).

Overall, IDS using machine learning is an energizing and quickly developing area of exploration that holds extraordinary commitment for improving the security of computer systems and networks.

1.2. Contribution of Study

In wireless networks, Intrusion detection utilizing machine learning can benefit from several contributions. Some of the main contributions are:

Feature selection: One of the primary tasks of intrusion detection in wireless networks is identifying relevant features that can accurately represent the network traffic. Machine learning algorithms rely on features to learn patterns and identify anomalies. Therefore, developing effective include determination techniques can altogether work on the precision of intrusion detection systems.

Anomaly detection: Wireless networks are susceptible to different kinds of attacks, such as denial-of-service attacks, man-in-the-middle attacks, and rogue access point attacks. Anomaly detection using machine learning algorithms can help identify these attacks by detecting deviations from normal network behavior.

Real-time monitoring: Intrusion detection systems should be capable of detecting attacks in real-time. Machine learning algorithms that can handle huge volumes of information and provide real-time predictions can help to enhance the responsiveness of intrusion detection frameworks.

Model explainability: Machine learning models used in intrusion detection systems should be interpretable, which means that they should be able to explain why particular network traffic is classified as anomalous. This is important for understanding the rationale behind the intrusion detection system's decisions and for identifying false positives.

Transfer learning: Transfer learning methods can be utilized to work on the presentation of interruption discovery frameworks in wireless organizations. Transfer learning enables models trained on one dataset to be applied to another dataset with similar characteristics, thus reducing the need for large amounts of labeled data. By focusing on these contributions, it can develop more effective intrusion detection systems that can accurately identify and mitigate attacks in wireless networks.

2. Literature Review

Intrusion detection systems (IDS) are essential components of wireless network security. These systems are anticipated to detect unauthorized access to the network by monitoring and analyzing network traffic. One approach to developing an IDS for wireless networks is to use machine learning algorithms. ML calculations can dissect a lot of information and recognize designs that may indicate a potential security threat (Kumar, G., & Alqahtani, H. 2023). It will offer a thorough analysis of the field's current research. Intrusion detection, ML approaches, and their various forms are then introduced. It will offer a thorough analysis of prior works, talk about progressing research issues regarding energetic execution of ML-based intrusion detection in wireless networks, and explore interesting new developments in this area. Recently, a number of methods for building an effective IDS have been put out (Chen, A. C. H., et al., 2022). However, due to conventional networks' scattered characteristics, ML approaches have restricted access to the data for analysis. Switches and other network devices have a limited perspective of the data that makes up a little part of the whole network. Subsequently, ML models trained on a specific network segment cannot be used to identify an intrusion across the entire network. (Granato, G., et al 2020) proposed measured ensemble of classifiers for spotting vindictive assaults to tackle measurable, computational and authentic issues in intrusion detection system because of the overload of data in wireless networks. The network security has turned into a significant focal point of PC security exploration. The assault on the network framework is a risk to the network and data insurance. In this research paper, the creators describe the design and implementation of the ensemble system, including the selection and configuration of individual classifiers, the integration of their outputs, and the decision-making process. The optimization aspect implies that the ensemble of classifiers may undergone a process of parameter tuning, feature selection, or other optimization techniques to enhance its performance in accurately identifying and classifying intrusions while minimizing false alarms.

The system introduced is tested on well-known AWID dataset. (Hochin, T., et al., 2017) proposed framework with two approaches such as k-means and Random-forest for network insurance. Network security is the main detached necessities. Network assault or interruption is a last divulgence, handicap, steal or endeavor to acquire unapproved access. Hackers could compose unapproved approach to a solitary PC or network associated PC to harm. The issues looked by network interruption is the cutting-edge insurance discovery industry. The dataset used in this research is KDD CUP'99, a most widely utilized dataset for intrusion detection. The examination is executed utilizing K-means and Random Forest techniques. The authors utilized the K-means algorithm to create diverse data set to almost assorted data set. Their future work includes using data mining approaches for real network data. (Kolias, C., et al., 2016) introduced TermID to effectively detect and mitigate security threats in wireless networks. The approach is based on swarm intelligence, which is inspired by the collective behavior of social insects. The streamlining has been utilized on a pre-handled AWID dataset, conglomerating elements to protect security in a focal hub and track down better approaches to recognize assaults. The authors perform disconnected traffic examination by embracing a Macintosh total measurements approach concentrating on outlines in pre-characterized time windows. viability of the Introduced technique is exhibited specifically by the fascinating On IF-THEN standard based interpretability point of view, for which has been depicted how rich of data could be edges' time windows to find vindictive edges. The dataset was in rough condition and having over fitting. The pre-processing techniques were applied on the dataset to make it used to train the model. (Chang, Y., Li, W., & Yang, Z. 2017) developed different machine learning approach for network intrusion detection by using random forest and support vector machine. The growing speed of internet utilization in present time has been dynamically unstable. There is a requirement for interruption recognition of networks due to the over load of data on internet. In this research, the analysts proposed a new machine learning approach having SVM as classification algorithm and RF as feature selector. The authors utilize KDD 99 dataset having five categories such as DOS, Normal, R2L, U2R, and Probe. (Universitas Indonesia. Fakultas Ilmu Komputer, 2017) for decreasing data complexity, feature selection process assumes a huge part in enhancing the learning ability of any machine learner.

In present time, Wi-Fi networks since inescapable IoT gadgets make enormous traffic and helpless simultaneously. Recognizing known and obscure assaults in Wi-Fi networks is a complicated challenge. In this research, the analysts test and approve the achievability of the chosen features utilizing a typical neural network. The research approach consists of feature selection and classification task. ANN is used for this approach. Also, the feature selection method C4.5 is applied on data set and then compares the results of ANN and C4.5. the evaluation metric detection rate(DR), false alarm rate(FAR) are utilized to estimate the execution of methods. High-layered unique elements are inspected involving a weighted-feature selection technique to dispense with excess and insignificant features. Their future work includes using deep learning methods to incorporate large dataset. (Abedin, M. Z., et al., 2018) proposed a system by using multiple classification algorithms such as naïve base, J48, multilayer perceptron, random forest and naïve base tree on three subsets of selected features from main dataset NSL KDD. The features are selected by doing pre-processing of the dataset which includes data cleaning, transformation, feature selection, and normalization. Attacks which are tackled in this research paper are DOS, U2R, P2L, and Probe. The introduced techniques are estimated utilizing accuracy, AUC, precision, recall, and F1-score. Optimal feature selection is very important for best intrusion detection system using machine learning models. As the precision of ML models rely upon the elements, the determinations of huge features are exceptionally essential. (Kasongo, S. M., & Sun, Y. 2020) presents a deep learning method feed forward DNN that incorporates wrapper-based feature extraction for wireless intrusion detection systems. Extra Tree classifier is used to build this method. The introduced method is used on AWID and UNSW-NB15. The approach utilizes convolutional neural networks to classify network traffic and demonstrates improved performance compared to traditional machine learning techniques. This suggests that the combination of deep learning and feature selection techniques can effectively enhance the detection capabilities of wireless intrusion detection systems. The findings suggest the potential of deep learning in strengthening the security of wireless networks by accurately identifying and mitigating intrusion attempts. Another research by (Aminanto, M. E., & Kim, K. 2017a) introduced DL method with feature extraction for WIDS. The authors worked on AWID dataset to make improvements and to classify impersonation attack. They utilized feature selection for their research. They preprocessed the dataset by excluding integer values and unnecessary attributes. They did feature selection by using ANN. Then to validate the execution of their selected features, they used DL model Auto encoder. The dataset consisted both balanced and unbalanced information. They evaluated their models using detection rate and false positive rate. They obtained 85% DR and 2.36 FPR. They selected only one attack to test and this was the major limitation and their future work include testing their model for other attacks. Therefore, the unauthorized access of information from malicious users should be controlled and the detection of attacks too. To solve these issues, (Wang, S., et al., 2019a) suggested a deep learning-based ID approach for wireless networks. ID frameworks recognize the interlopers and secure the association and data by forestalling access to illegal clients. The potency of intrusion detection is being split into three phases including monitoring, analysis, and detection. They tested their method on well know AWID dataset, which consisted on two distributions such as AWID-CLS-R-TRN, and AWID-CLS-R-TST. They pre-processed their dataset and removed unnecessary attributes. They utilized stacked encoder and DNN for implementation and examined 4 categories of attacks such as normal, injection, flooding, and impersonation and obtained 73.12% accuracy.

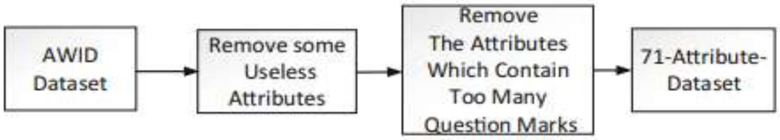

AWID Dataset
Remove some Useless Attributes
Remove The Attributes Which Contain Too Many Question Marks
71-Attribute-Dataset

Figure 3: Data pre-processing by (Wang, S., et al., 2019a)

(Yang, H., & Wang, F. 2019) proposed an enhanced CNN based model for the network intrusion detection with high detection rate and low FPR. NSL KDD-CUP (L.Dhanabal, & Dr. S.P. Shantharajah. 2015) dataset is utilized for this research. The classification preparing and test experiments are completed in IBWNIDM utilizing the pre-handled preparing set and test set information. The analysts proposed a reliable CNN-based wireless network interruption detection mechanism (IBWNIDM). The evaluation of this model is done using accuracy, FPR, and TPR. Three experiments are conducted having KDDTest + dataset to IBWNIDM in first, NSL KDD-CUP to RNN, DBN, LeNet-5 in second, and NSL KDD-CUP to IDABCNN, and NIDM-BCNN in third experiment. Their proposed model needed improvement having loss function and will use SGD for intrusion detection. Another research by ((ICICES, 2014) in which they introduced IDC in adhoc networks utilizing ML techniques. Interruption recognition on MANET is a difficult undertaking because of its special qualities like open medium, dynamic geography, absence of concentrated administration and profoundly asset compelled hubs. The authors utilized ML models such as RST and SVM. They utilized methodology with following steps such as data collection, feature extraction, model implementation, and then detection rate of intrusion. Another research by (Simon, J., et al., 2022) introduced hybrid intrusion identification framework in wireless networks using DL. Because the quantity of internet consumers are increasing day by day and traffic network is also increasing. They utilized NSL-KDD dataset and decision tree algorithm. The curiosity of the exploration work is available in the mix of classifier models utilized in the profound learning organization. Table 1 illustrates some previous studies.

Table 1: Technical Background

Ref	Methodology	Detection technique	Algorithm	Dataset	Attack Classes	Results	Limitations
(Granato, G., et al., 2020)	ID in wifi networks using ensemble of classifiers	Intrusion detection	BSAS and K-NN driven by GA	AWID	14 attack classes	ACC=90 % FAR=5.9	Lower Complexity tools. Low reliability of classifiers.
(Hochin, T., et ak., 2017)	Network intrusion recognition system using RF and K-means	DOS attack detection	RF, Kmeans	KDD CUP'99	Normal Attack	ACC=75%	Less efficient time. Large dataset.
(Kolias, C., et al., 2017)	Swarm intelligence based system	Anomaly based ID	Ant Colony-based	AWID	Normal Attack	ACC=77.8%	Low accuracy. Identification of novel attacks.
(Chang, Y., Li, W., & Yang, Z. 2017)	Intrusion detection using machine learning approach	Intrusion detection	RF, SVM	KDD 99	Normal Attack	Detection rate= 88.2%	All the features of data have not utilized. Optimization of hyperplanes is not determined.
(IWBIS, 2017)	Weighted feature selection for ANN classifier	Intrusion detection	ANN	AWID	Normal Attack	ACC=82.47%, FAR =0.41%	Unbalanced dataset.
Abedin, M. Z., et al, 2018)	Anomaly based intrusion detection using ML classification models	Anomaly based ID	NB, RF, J48, MLP	NSL-KDD	Multi Attacks: DOS U2R R2L Probe	ACC= 80%, Recall= 76% AUC=91%	Crucial selection of features. ML models to be retrained.
(Kasongo, S. M., & Sun, Y. 2020)	Wrapper based FFDNN	Intrusion detection	WFEU-FFDNN	UNSW-NB15, AWID	Normal Injection Impersonation flooding	ACC=87.7% DR= 77.1%	Limited Literature.
(Aminanto, M. E., & Kim, K. 2017a)	Impersonation attack detection using feature selection	Impersonation based intrusion detection	ANN, AE	AWID	impersonation	85% detection rate 2.36 FPR	Only one attack selected.
(Wang, S., Li, B., et al., 2019)	DL based intrusion detection	Intrusion detection	DNN	AWID	Normal Injection Impersonation flooding	ACC= 73.12%	Less accuracy obtained.
L.Dhanabal, & Dr. S.P. Shantharajah. 2015)	Improved CNN for WIDS	Intrusion detection	CNN, RNN, DBN	NSL KDD-CUP	DOS U2R R2L Probe Normal	ACC=86.54% TPR=91.31% FPR=2.03%	Less generalization capability Low TPR
(ICICES, 2014)	ID in MANET using ML	Intrusion, Anomaly Detection	RST, SVM	Traffic data	Normal Attack	Detection rate=77.5%	Data Reduction

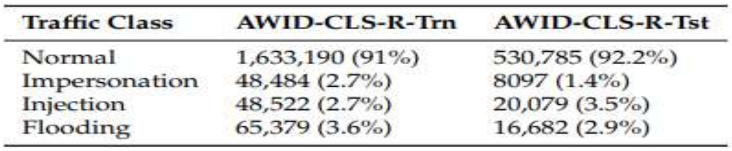
(Simon, J., et al, 2022)	Hybrid ID in wireless networks	Intrusion detection	DT, CNN	NSL-KDD	Normal Attack	Detection rate=75%	NSLKDD dataset was utilized which does not reflect present-day assaults

3. Methodology and Techniques

In this research work, it has utilized publicly accessible dataset AWID Aegean Wi-Fi Intrusion Dataset. The AWID data set consists of network traffic captured from a wireless sensor network that was set up for research purposes. The dataset includes both normal and abnormal traffic, with the abnormal traffic representing different types of attacks. The dataset was specifically designed to evaluate the effectiveness of intrusion detection systems in wireless sensor networks (Kolias, C., et al., 2016b). The dataset includes over 200,000 network packets and covers a range of wireless protocols like ZigBee, 6LoWPAN, and IEEE 802.15.4. It also contains various kinds of attacks, including denial-of-service attacks, flooding attacks, and injection assaults. The dataset is labeled with ground truth information, which makes it ideal for supervised learning tasks.

It is involving AWID-CLS-R variant in this research task, a diminished data set regarding size and quantity of records. AWID-CLS-R incorporates separate sets for preparing (AWID-CLS-R-Trn) and test (AWID-CLS-R-Tst) basis. There are f significant classes in the data set: (i) Flooding, (ii) Infusion, (iii) Normal, and (iv) impersonation. The initial three classes have a place with interruption assaults and the last one addresses typical traffic. Further details about the records of the dataset are mentioned in Table 2.

Table 2: Dataset Record Distribution



Traffic Class	AWID-CLS-R-Trn	AWID-CLS-R-Tst
Normal	1,633,190 (91%)	530,785 (92.2%)
Impersonation	48,484 (2.7%)	8097 (1.4%)
Injection	48,522 (2.7%)	20,079 (3.5%)
Flooding	65,379 (3.6%)	16,682 (2.9%)

Intrusion Detection Systems (IDSs) are designed to identify and respond to potential security breaches in wireless networks. An IDS is typically composed of two components: a sensor that collects data about the network traffic and an analyzer that processes the collected data to detect anomalies or attacks. The methodology of building an IDS with feature selection in wireless networks can be broken down into several steps:

3.1. Data Collection

The first step in building an IDS is to gather data about the network traffic. This data can be collected by sniffing packets on the network or by using network flow data. The data collected should include data such as the source and destination IP addresses, the source and destination port numbers, the packet size, and the protocol type. In this thesis work, it is going to use the already collected dataset AWID3 which is publicly accessible.

3.2. Data cleaning

In this second step, the data cleaning is performed. Exploratory analysis is performed to see the labels in this step. Data cleaning is done by performing Removing Rows with nan as labels, removing un necessary columns, replacing mac addresses with 1 or 0 etc.

3.3. Feature Extraction

The following stage is to extricate the pertinent highlights from the information. Highlight extraction includes transforming the crude information into a bunch of elements that can be utilized by the IDS. There are many techniques for feature extraction, it is using cross-validation, chi-square, correlation variance etc.

3.4. Model Training

After the features have been extracted, the next step is to train a model to detect potential attacks. The model can be trained using a supervised or unsupervised learning approach. In supervised learning, the model is prepared utilizing labeled information, where the labels indicate whether the data represents a normal or anomalous behavior. In unsupervised learning, the model is trained utilizing unlabeled information, and the model is expected to identify anomalies based on the underlying patterns in the data. Here it is using machine learning supervised algorithms and deep learning technique.

3.5. Model Evaluation

Once the model has been trained, it is assessed utilizing a bunch of testing information. The testing data should be independent of the training data and should include both normal and anomalous behavior. The execution of the trained models are assessed based on metrics like accuracy, precision, recall, and F1 score.

In conclusion, building an IDS with feature selection in wireless networks involves collecting data, extracting features, training a model, and evaluating the model. The IDS can be used to detect potential security breaches in wireless networks and to trigger an appropriate response to those breaches. The IDS research methodology is shown in figure 4.

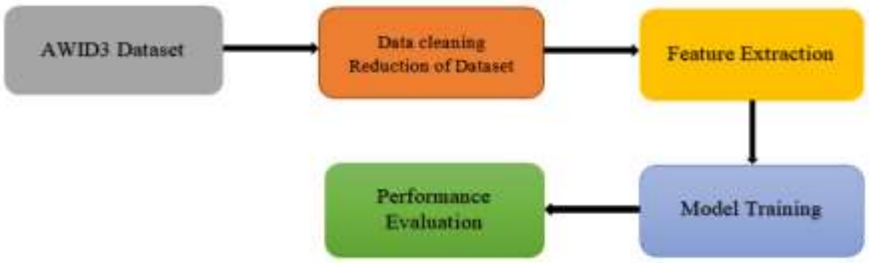


Figure 4: Our purposed IDSs Research Methodology

4. Experimental Results and Discussion

4.1. Machine Learning models using Chi-square technique:

4.1.1. Working of Chi square technique:

As that is mentioned before that chi square is a statistical method. It works with expected frequencies. The expected frequencies are calculated based on the assumption that there is no huge distinction between the noticed and anticipated frequencies. The formula for calculating expected frequencies varies depending on the type of data you are analyzing.

The Chi-Square statistic is determined by contrasting the observed and expected frequencies. The formula for calculating the Chi-Square statistic is:

Chi-Square = ∑ ((Observed Frequency - Expected Frequency)^2 / Expected Frequency)   (1)

4.1.2. Working of chi square using Machine Learning models

The Chi-Square test can be utilized as a feature selection method in Machine Learning models. Here, the essential thought is to utilize the Chi-Square statistic to measure the dependence between each feature and the target variable, and determine the top k features with the highest Chi-Square values.

Here's how the Chi-Square test is applied with ML models:

1. **Calculate Chi-Square statistic:** it is calculated the Chi-Square statistic between each feature and the target variable. This has done by creating a contingency table between the feature and the target variable, and then calculating the Chi-Square value using the formula described before.
2. **Select features:** it is selected the top k features with the highest Chi-Square values. This has done by ranking the features relayed on their Chi-Square values and select the top k features.
3. **Train Machine Learning model:** Then it has trained all the models using selected features such as AdaBoost, Bagging classifier, Extra tree, random forest, decision tree, XGBoost, and Guissian NB.

To better the performance of all the models, it has utilized cross-validation with a specified number of folds. It has used value for the number of folds is 5, which means that the accessible information is being split into 5 equal parts, and the training and evaluation process is repeated 5 times. In each iteration, 4 of the parts are utilized for training, and the endure part is utilized for evaluation. The final execution metric is the average of the performance metrics obtained in the each iteration.

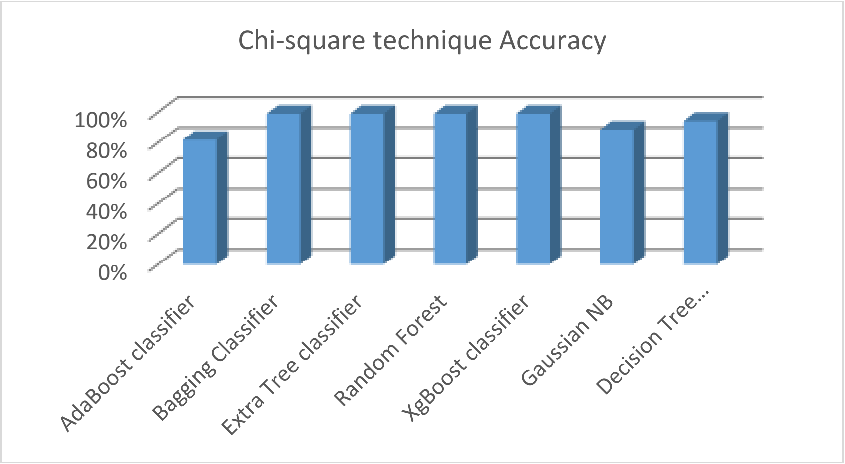


Figure 5: ML model's accuracy using Chi-square technique

4.2. Machine Learning models using Feature importance

Feature importance is a technique used in machine learning to distinguish the significant features or variables that come up with the accuracy of a model. It has selected features using feature importance value greater than 1.0 and then the models are trained using the selected features to increase all model performances. It has calculated the significance of each feature by analyzing how much it comes with the model's performance. The models that are trained using selected features by feature importance are AdaBoost, XGBoost, Extra Tree, Bagging, Random Forest, Gaussian NB, and Decision Tree classifier.

In this implementation, it has again utilized cross-validation to make better the performance of all models. It has used value for the number of folds is 5, which means that the available data is divided into 5 equal parts, and the training and evaluation process is repeated 5 times. In each iteration, 4 of the parts are utilized for training, and the endure part is utilized for evaluation. The final execution metric is the average of the performance metrics obtained in the each iteration.

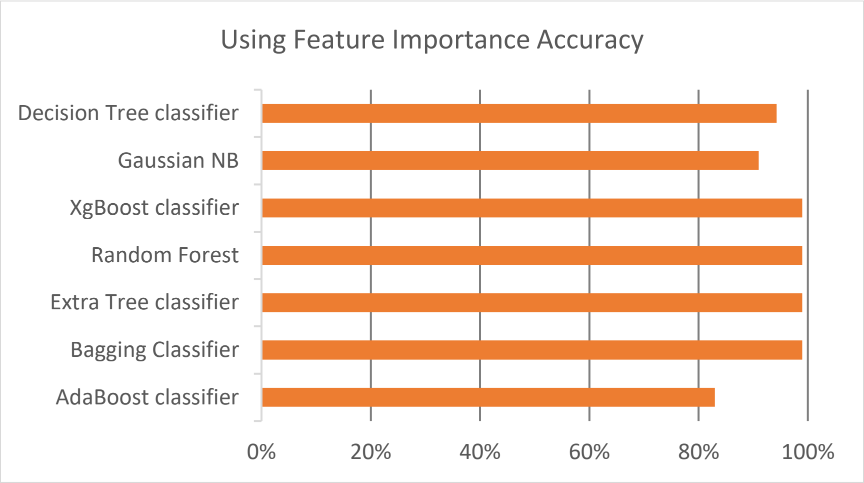


Figure 6: ML model's accuracy using feature importance

4.3. Machine Learning correlation variance

As that is mentioned before correlation variance refers to the degree of features in a dataset that are related or vary together. By using correlation variance, it has selected most informative and non-redundant features. Then it has trained models on informative features extracted using correlation variance. The models are AdaBoost, XGBoost, Extra Tree, Bagging, Random Forest, Gaussian NB, and Decision Tree classifier.

In this implementation, it has again utilized cross-validation to make better the performance of models. It has used value for the number of folds is 5, which means that the accessible information is divided into 5 equal parts, and the training and evaluation process is repeated 5 times. In each iteration, 4 of the parts are utilized for training, and the endure part is utilized for evaluation. The final execution metric is the average of the performance metrics obtained in the each iteration.

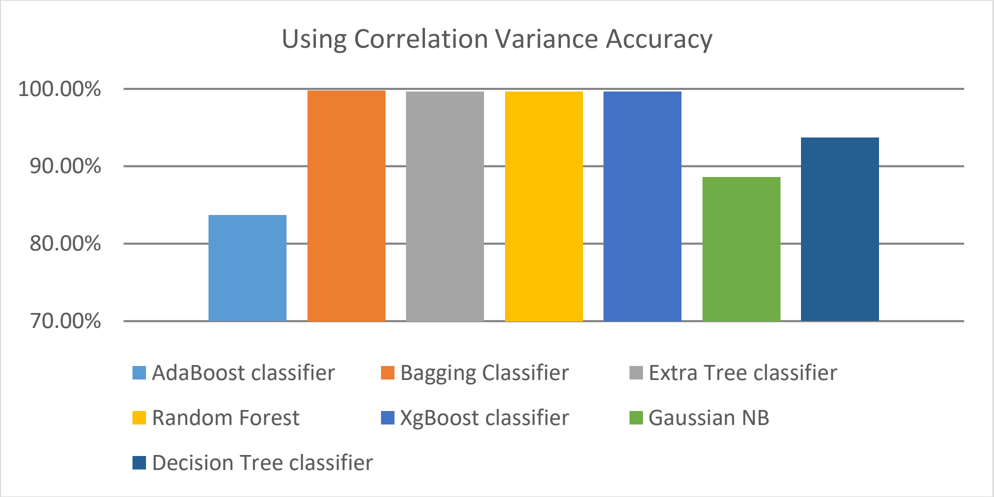


Figure 7: ML model's accuracy using correlation variance

4.4. Deep Learning model implementation

It has trained DCNN model on the dataset to see which model fits best for intrusion detection in wireless networks.

4.4.1. Deep Convolutional Neural Network (DCNN) implementation

The implementation of DCNN involves one hot encoding of labels and then uses the dataset for training the model.

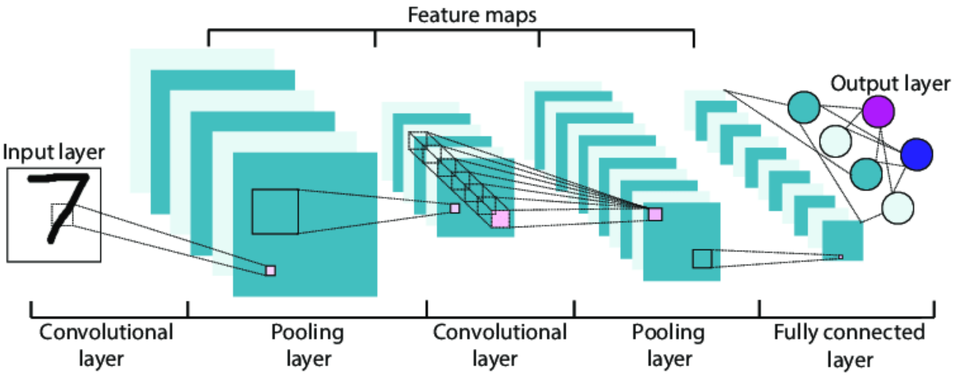


Figure 8: DCNN architecture

4.4.1.1. One hot encoding

Hot encoding is a technique utilized in data analysis to transform categorical information into numerical values. In the Aegean WiFi intrusion detection dataset, there are several categorical features such as protocol type, service, and flag. Hot encoding can be applied to these features to convert them into numerical values, which could utilize as input for machine learning algorithms.

The procedure of hot encoding includes making another twofold segment for every extraordinary class in an all-out feature. For example, the protocol type feature in the Aegean WiFi intrusion detection data set has three unique categories: TCP, UDP, and ICMP. To hot encode this feature, three new binary columns are created: one for TCP, one for UDP, and one for ICMP. The binary column for each category will have a value of 1 if the original categorical feature value matches the category and 0 otherwise.

In this dataset, it has used one hot encoding to convert the categorical features of training dataset to numerical values, and then the resulting dataset has numerical values that are used as input for training of DCNN.

4.4.2. Constructing Model

This step includes the architecture of the model. The choice of model architecture will base on elements including the size of the dataset, the complexity of the features. A DCNN (Deep Convolutional Neural Network) model is a type of neural network that is commonly utilized for the analysis of WiFi traffic data. Here is an overview of the different layers used in the implementation of DCNN model for Aegean WiFi intrusion detection. It has used multiple layers of CNN such as dense, max pooling, dropout, flattern, and Conv1D.

4.4.2.1. Conv1D layer

The Conv1D layer is a type of convolutional layer that applies a one-dimensional convolution operation to the input information. For dataset, the Conv1D layer is used to extract features from the time-series WiFi traffic data. This layer applies a bunch of channels to the information and produces a bunch of result highlight maps.

4.4.2.2. MaxPooling1D layer

It has used MaxPooling1D layer to decrease the dimensionality of the feature maps made by the Conv1D layer. This layer applies a one-dimensional max pooling operation to the input feature maps and produces a set of smaller output feature maps. This reduces the computational complexity of the model while preserving important features.

4.4.2.3. Dense layer

The Dense layer is a fully connected layer that put in a linear transformation to the input data. It has used the Dense layer to perform classification on the extracted features. This layer took the flattened feature maps as input and applies a bunch of weights to produce a bunch of output values.

4.4.2.4. Flatten layer

It has used flatten layer to transform the output feature maps made by the Conv1D layer and MaxPooling1D layer into a one-dimensional array. This is necessary because the Dense layer requires a 1-dimensional input. The Flatten layer flattens the output of the previous layer so that it can be fed into the Dense layer. The trainable parameters are 10,674 out of all parameters. It has used Adam optimizer for improving performance of DCNN. The Adam optimizer calculates individual adaptive learning rates for each parameter in the network based on the past gradients and the running averages of both the inclinations and the squared angles. This means that it adapts the learning rate regarding all parameters in the network based on its past behavior and the behavior of the other parameters.

4.4.2.5. Optimizer

For training the DCNN model for Aegean WiFi intrusion detection, it has used Adam optimizer to renovate the weights of the network during the back-propagation process. The objective is to limit the misfortune capability that actions the contrast between the anticipated result and the genuine result. The Adam optimizer maintains two dramatically rotting moving midpoints of past inclinations: the first moment estimate (mean) and the second moment estimate (variance). These are updated during training as follows:

Initialize the first moment estimate (m) and second moment estimate (v) vectors to 0. At each iteration t, compute the angle of the loss capability with deference to the parameters of the model, denoted as gt. Update the first moment estimate (m) and second moment estimate (v) using the following equations:

$m\_t = \beta1 * m\_t\text{-}1 + (1 - \beta1) * g\_t$ $v\_t = \beta2 * v\_t\text{-}1 + (1 - \beta2) * g\_t\text{^}2$ where β1 and β2 are the decay rates for first and second moment. It has taken decay rate=5e-6 and momentum=0.9. Compute the bias-corrected estimates of the first and second moments:

m_t_hat = m_t / (1 - β1^t) v_t_hat = v_t / (1 - β2^t) where t is the current iteration.
Update the parameters of the model utilizing the followed equation:
θ_t = θ_t-1 - α * m_t_hat / (√v_t_hat + ε)
where α is the learning rate it has taken learning rate = 0.001, ε is a little steady to forestall division by nothing, and θ is the vector of model parameters.

4.4.2.6. Activation functions

This function perform a critical role in deep learning models, as it introduce non-linearity into the network and help it to learn complex and non-linear relationships between inputs and outputs (Sharma, A. 2020). It has used two activation functions such as Softmax and Relu.

The ReLU activation function is a simple non-linear function that is widely utilized in DL models, especially in Convolutional Neural Networks (CNNs). It is defined as:

f(x) = max(0, x) (1)

The ReLU function is a piecewise direct capability that yields the information esteem on the off chance that it is positive, and zero in any case. It has used this function because it is computationally efficient, easy to implement, and has been demonstrated to be viable in working on the preparation of profound brain organizations.

It has utilized Softmax activation function in the output layer. It is defined as:

f_j(x) = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} (2)

where x_j is the input to the j-th neuron in the output layer, K is the number of neurons in the output layer, and e is the Euler's number. The Softmax capability changes over the result of the last layer of the neural network into a likelihood dispersion over the classes, ensuring that the sum of the probabilities of all classes is equal to 1. This makes it useful for multi-class classification problems.

4.4.2.7. Loss function

It has used cross entropy loss function to measure the model loss. Cross-entropy misfortune estimates the contrast between the anticipated likelihood dissemination and the genuine likelihood circulation of the objective classes (Koech, K. E. 2023).

The cross-entropy loss function is defined as:

J(\theta) = -\frac{1}{N}\sum_{i=1}^N\sum_{j=1}^M y_{ij}log(\hat{y}_{ij}) (3)

where \theta are the parameters of the network, N is the quantity of training examples, M is the number of classes, y_{ij} is the true label (binary 0/1) of the i-th example for the j-th class, and \hat{y}_{ij} is the predicted probability of the j-th class for the i-th example. The log function is applied to the predicted probability to ensure that the loss is larger for larger differences between the predicted and true labels. Table 3 showing DCNN architecture.

4.4.3. Proposed DCNN architecture

Table 3: Proposed DCNN Architecture

Classification	Model	Optimizer	Loss function	epochs	Activation function	Metric
Multi-class	DCNN	Adam optimizer	Cross entropy	100	RELU SOFTMAX	Accuracy

The Table 3 shows the summary of DCNN model architecture, in which it has describe all important elements of DCNN. The description of optimizer, loss function, activation function are described above. The training of DCNN model consists 100 epochs to train the dataset having 10% validation dataset. This validation data set is withheld before training model to provide an unbiased gauge of model execution. Because the batch size is 128, the model will be refreshed once it has processed 128 samples. It is having large batch size and high number of epochs because it has large dataset to train.

4.5. Deep Learning model using feature extraction techniques

It has also trained model by using feature extraction techniques to get more accurate results.

In this context, it will explore the use of DCNN on the Aegean intrusion dataset using several statistical techniques, including chi-square, correlation variance, and feature importance. One such technique is chi-square analysis, which measures the association between the extracted features and the intrusion/non-intrusion classes. By calculating the chi-square value for each feature, it has identified those with the highest degree of association with the intrusion class. These features have considered the most important ones for detecting the presence of a volcanic intrusion.

Another technique it has used is correlation variance analysis, which measures the degree of correlation between the features and the intrusion class. By calculating the correlation coefficient for each feature, it has identified those with the highest degree of correlation with the intrusion class. These features are considered the most relevant ones for intrusion detection.

Finally, it has used feature importance analysis to determine the contribution of each feature to the overall execution of the DCNN model. This has done by training the model with and without each feature and measuring the change in performance. The features that result in the most significant changes in performance can be considered the main ones for interruption identification.

In conclusion, using DCNN on the AWID and analyzing the extracted features using statistical techniques such as chi-square, correlation variance, and feature importance have helped to recognize the important features for detecting the presence of a submarine volcanic intrusion accurately.

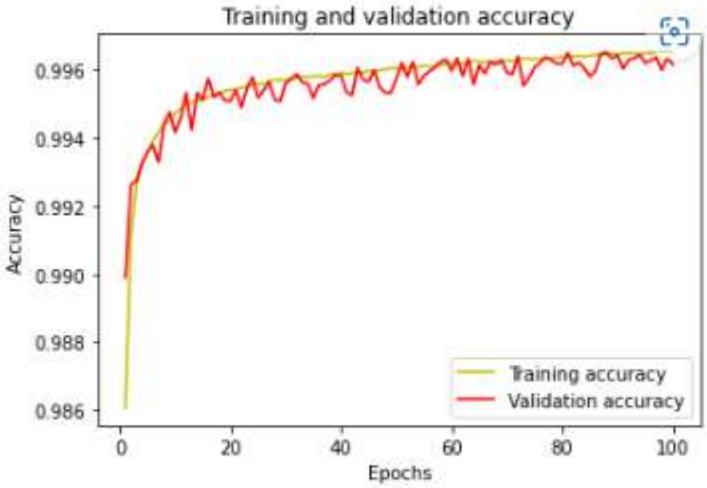


Figure 9: DCNN Model training and validation accuracy results

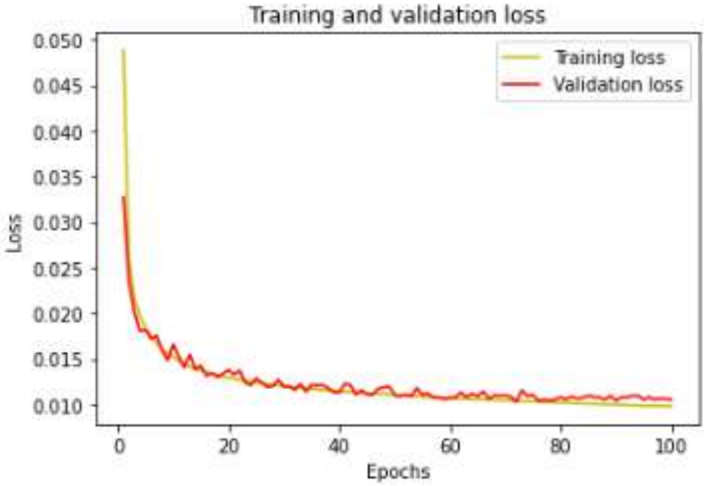


Figure 10: DCNN training and validation loss

The DL model DCNN also performed well and the model's execution is estimated using accuracy, f1-score, confusion matrix, and recall.

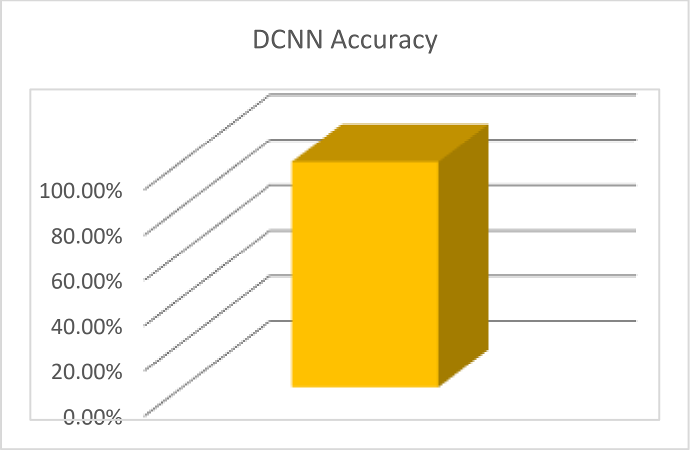


Figure 11: DCNN Model's Summary

5. Conclusion and Future work

Intrusion detection in wireless networks is a critical area of research that has gained significant attention in recent years. In this research, it has trained both ML and DL models on AWID dataset. The Aegean WiFi Intrusion dataset is a popular benchmark data set for estimating the execution of intrusion detection systems based on Machine Learning (ML) and Deep Learning (DL) models.

Several studies been conducted on the Aegean WiFi Intrusion dataset utilizing different ML and DL models, such as Random Forest, Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). The outcomes of these studies have demonstrated the effectiveness of ML and DL models in accurately detecting and preventing network intrusions. It has utilized ML models including AdaBoost, Bagging, Extra Tree, decision tree, random forest, XGBoost, and Gaussian NB classifier. Also it has obtained results using DL model DCNN. It has excluded unnecessary features from the dataset by using different pre-processing techniques. It has cleaned the dataset by removing constant and redundant rows from the dataset. It has got necessary 28 features from 254 features. Then it trained out models on the dataset with selected features. It has also trained the models using feature extraction techniques such as feature importance, correlation variance, and chi-square technique to get better results. It has obtained best results from these models and get that these models can fit best for intrusion detection in wireless networks.
However, further research is required to optimize these models and techniques for real-world scenarios and to ensure their reliability and scalability. With the continued advancements in ML and DL, it is expected that these models will become more sophisticated and effective in intrusion detection, ultimately leading to safer and more secure wireless networks.

References

Abdulhammed, R., Faezipour, M., Abuzneid, A., & AbuMallouh, A. (2019). Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic. *IEEE Sensors Letters*, *3*(1), 1–4.

Abedin, M. Z., Siddiquee, K. N. E. A., Bhuyan, M. S., Karim, R., Hossein, M. S., & Andersson, K. (2018). *Performance Analysis of Anomaly Based Network Intrusion Detection Systems*.

Aburomman, A. A., & Reaz, M. B. I. (2016). *Survey of learning methods in intrusion detection systems*.

Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1).

Alrajeh, N. A., Khan, S., & Shams, B. (2013). Intrusion Detection Systems in Wireless Sensor Networks: A Review. *International Journal of Distributed Sensor Networks*, *9*(5), 167575.

Aminanto, M. E., & Kim, K. (2017a). Detecting Impersonation Attack in WiFi Networks Using Deep Learning Approach. In *Lecture notes in computer science* (pp. 136–147).

Aminanto, M. E., & Kim, K. (2017b). Detecting Impersonation Attack in WiFi Networks Using Deep Learning Approach. In *Lecture notes in computer science* (pp. 136–147).

Bace, R., & Mell, P. (2001). *NIST Special Publication on Intrusion Detection Systems*.

Chandrabala P. Kothari, & Vaishali Kulkarni. (2014). Intrusion Detection System using Wireless Sensor Networks: a Review. *IJERT, Volume 03, Issue 12 (December 2014)*(2278–0181), IJERTV3IS120537.

Chang, Y., Li, W., & Yang, Z. (2017). *Network Intrusion Detection Based on Random Forest and Support Vector Machine*.

Chen, A. C. H., Jia, W. K., Hwang, F. J., Liu, G., Song, F., & Pu, L. (2022). Machine learning and deep learning methods for wireless network applications. *EURASIP Journal on Wireless Communications and Networking*, *2022*(1).

Denning, D. (1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, *SE-13*(2), 222–232.

Djenouri, D., Khelladi, L., & Badache, A. (2005). A survey of security issues in mobile ad hoc and sensor networks. *IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials*, *7*(4), 2–28.

Gajawada, S. K. (2023, August 16). Chi-Square Test for Feature Selection in Machine learning. *Medium*.

GeeksforGeeks. (2023, February 6). *XGBoost*. GeeksforGeeks. https://www.geeksforgeeks.org/xgboost

Granato, G., Martino, A., Baldini, L., & Rizzi, A. (2020). *Intrusion Detection in Wi-Fi Networks by Modular and Optimized Ensemble of Classifiers*.

Gusarova, M. (2023, February 20). Decision Trees Explained. graphviz and shap - Maria Gusarova - Medium. *Medium*.

Hochin, T., Hirata, H., & Nomiya, H. (2017). *18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017)*.

Hodo, E., Bellekens, X. J. A., Hamilton, A. W., Tachtatzis, C., & Atkinson, R. C. (2017). Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. *arXiv (Cornell University)*.

International Conference on Information Communication and Embedded Systems (ICICES), 2014. (2014).

Islam, Md & Rehman, & Syed Ashiqur. (2011). Anomaly Intrusion Detection System in Wireless Sensor Networks: Security Threats and Existing Approaches. *International Journal of Advanced Science and Technology*, *36*.

Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). *A Deep Learning Approach for Network Intrusion Detection System*.

Jayveer Singh, & Manisha J. Nene. (2013). A Survey on Machine Learning Techniques for Intrusion Detection Systems. *International Journal of Advanced Research in Computer and Communication Engineering*, *2*(11), 7.

Kasongo, S. M., & Sun, Y. (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Computers & Security*, *92*, 101752.

Kégl, B. (2013). The return of AdaBoost.MH: multi-class Hamming trees. *arXiv (Cornell University)*.

Khan, S., Mast, N., Loo, K., & Silahuddin, A. (2008). *Cloned Access Point Detection and Point Detection and Prevention Mechanism in IEEE 802.11 Wireless Mesh Networks*.

Koech, K. E. (2023, August 6). Cross-Entropy Loss Function - Towards Data Science. *Medium*.

Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2016a). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials*, *18*(1), 184–208.

Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2016b). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials*, *18*(1), 184–208.

Kolias, C., Kolias, V., & Kambourakis, G. (2016). TermID: a distributed swarm intelligence-based approach for wireless intrusion detection. *International Journal of Information Security*, *16*(4), 401–416.

Kumar, G., & Alqahtani, H. (2023). Machine Learning Techniques for Intrusion Detection Systems in SDN-Recent Advances, Challenges and Future Directions. *Computer Modeling in Engineering & Sciences*, *134*(1), 89–119.

Kumar, Sandeep, & Spafford, Eugene. (1999). A Software Architecture to Support Misuse Intrusion Detection. *Purdue University*, 17.

L.Dhanabal, & Dr. S.P. Shantharajah. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 7.

Loo, C. E., Ng, M. Y., Leckie, C., & Palaniswami, M. (2006). Intrusion Detection for Routing Attacks in Sensor Networks. *International Journal of Distributed Sensor Networks*, *2*(4), 313–332.

Luo, Q., Engineers, I. O. E. a. E., Association, I. I. T. a. R., & Wuhan-Gongcheng-Daxue. (2010). *2010 Second International Conference on Communication Systems, Networks and Applications (ICCSNA 2010)*.

Martins, C. (2023, November 2). *Gaussian Naive Bayes Explained With Scikit-Learn*. Built In.

Mohammadpour, L., Ling, T. C., Liew, C. S., & Aryanfar, A. (2022). A Survey of CNN-Based Network Intrusion Detection. *Applied Sciences*, *12*(16), 8162.

Mudzingwa, D., & Agrawal, R. (2012). *A study of methodologies used in intrusion detection and prevention systems (IDPS)*.

Mukherjee, B., Heberlein, L., & Levitt, K. (1994). Network intrusion detection. *IEEE Network*, *8*(3), 26–41.

Pushpender Sarao. (2019). Machine Learning and Deep Learning Techniques on Wireless Networks. *International Journal of Engineering Research and Technology*, *12*(3), 10.

Rezvy, S., Luo, Y., Petridis, M., Lasebae, A., & Zebin, T. (2019). *An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks*.

Roman, R., Zhou, N. J., & Lopez, J. (2006). *Applying intrusion detection systems to wireless sensor networks*.

Sagers, G. (2021). WPA3: The Greatest Security Protocol That May Never Be. *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*.

Shaikh, R. (2018, October 28). Feature Selection Techniques in Machine Learning with Python. *Medium*.

Sharma, A. (2020, June 10). Understanding Activation Functions in Neural Networks. *Medium*.

Simon, J., Kapileswar, N., Polasi, P. K., & Elaveini, M. A. (2022). Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm. *Computers & Electrical Engineering*, *102*, 108190.

SouravKumarDas. (2020, May 10). *Random Forest Classification explained in detail and developed in R*. Data Science Central.

T. M. Khoshgoftaar, S. V. Nath, Shi Zhong, & N. Seliya. (2005). Intrusion detection in wireless networks using clustering techniques with expert analysis. *Fourth International Conference on Machine Learning and Applications (ICMLA'05)*, 6.

V. Srikanth, & I. Reddy. (2019). Wireless Security Protocols (ITP,WPA,WPA2 & WPA3). *International Journal of Scientific Research in Computer Science Engineering and Information Technology*.

Wang, S., Li, B., Yang, M., & Yan, Z. (2019a). Intrusion Detection for WiFi Network: A Deep Learning Approach. In *Springer eBooks* (pp. 95–104).

Wang, S., Li, B., Yang, M., & Yan, Z. (2019b). Intrusion Detection for WiFi Network: A Deep Learning Approach. In *Springer eBooks* (pp. 95–104).

Yalin E. Sagduyu, Yi Shi, Tugba Erpek1, William Headley, Bryse Flowers, George Stantchev, & Zhuo Lu. (2020). When Wireless Security Meets Machine Learning: Motivation, Challenges, and Research Directions. *arXiv*, arXiv:2001.08883v1 [cs.NI] 24 Jan 2020.

Yang, H., & Wang, F. (2019). Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network. *IEEE Access*, *7*, 64366–64374.

Zamani, M., & Movahedi, M. (2013a, December 8). *Machine Learning Techniques for Intrusion Detection*.

Zamani, M., & Movahedi, M. (2013b, December 8). *Machine Learning Techniques for Intrusion Detection*.