National College of Ireland

Providing Network-Centric Data Security Using Machine Learning and Intrusion Detection

MSc Research Project
MSc Cybersecurity

Komal Sharma
Student ID: 20248890

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Komal Sharma		
Student ID:	20248890		
Programme:	MSc Cybersecurity	**Year:**	2021-2022
Module:	MSc Research Project		
Supervisor:	Niall Heffernan		
Submission Due Date:	15-08-2022		
Project Title:	Providing Network-Centric Data Security Using Machine Learning and Intrusion Detection		
Word Count:	……6179…… **Page Count**……21……..		

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

Signature: Komal Sharma

Date: 15-08-2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Providing Network-Centric Data Security Using Machine Learning and Intrusion Detection

Komal Sharma
Student ID: 20248890

Abstract

In past few years, sophisticated cyber-attacks are growing day by day because the advancement of computer-based technologies and uses of computer-based applications in human life. Intrusion detection system are used to help in identifying the network layer attacks and provide security. But traditional Intrusion detection systems are not able to detect the novel attacks. Therefore, it become necessary to improve and deploy the intrusion detection system with the help of machine learning (ML) models. This can be done by analysing the intrusive data set.

In this research ensemble-based machine learning models along with intrusion detection system are proposed. The aim of this research is to identify the network-based intrusions by evaluating and comparing the performance of different Ensemble ML models. The performance of these models is measured based on accuracy, F-1 score and Cross-validation Score. In this research Aegean Wi-Fi Intrusion Dataset 3 (AWID3) is used which is newest version of AWID dataset having the data of many types of attacks. The dataset of 8 types of attacks is chosen for this research with 8 different Ensemble based ML models such as Bagging, Adaboost, Random Forest (RF), Extra tree, Gradient Boosting, Isolation Forest (IF), Stacking, and Voting Classifier. These ML models are trained by using the AWID3 dataset and performance of these models are evaluated. The results of models show that among all of models Random Forest, Extra tree classifier, and Voting classifier performed very well on AWID3 datset with a 100% accuracy, 100% F-1 score and 100% cross-validation score. That means these three models are most accurate to identify the network-based intrusion attacks.

Keywords: AWID3 dataset, Intrusion detection system, Ensemble machine learning

1 Introduction

An intrusion detection system can be referred to as a software application in order to detect network intrusions if any with the help of various machine learning algorithms. In the modern era, the advent of technology has led to various complex mechanisms of security due to the protection of security-sensitive data and in this aspect, network intrusion detection systems can play a major role. It is a type of network security technology that has been built for the detection of vulnerabilities by exploiting various computer applications (Fang et al., 2020). Machine learning and Artificial Intelligence can be two main branches of this security detection system in the form of intrusion detection as an important part of the Internet of

Things. It is an important part of cyber security in order to protect data, network, and IoT devices from phishing and any other kinds of malicious attacks or any types of unauthorized access. An intrusion detection system, in this context, helps to establish a monitoring system to detect and identify suspicious activities, if any, followed by the generation of alerts after the detection.

1.1 Motivation and Project Background

The traditional system has been found to be limited by its scale, refresh rate of the database, and feature filtering system which makes it unable to detect and identify all types of attacks, especially, the variants of new attacks. In this scenario, the researchers have been found to pay more attention to the area of machine learning to combine it with the traditional network security and intrusion detection approach to make the entire system more powerful. Various machine learning algorithms have been used by researchers in recent years to increase the eligibility of the traditional intrusion detection system by the application of Decision Tree, Random Forests, Support Vector Machine, Neural Networks, and others (da Costa et al., 2019). However, a thorough analysis is required to check the reliability and capability of various machine learning algorithms because it has been found that several algorithms may perform well depending on the type of the attacks, whereas some others do not perform well and show poor performance creating several issues. In the IoT environment, the application of machine learning in the intrusion detection system is an important and interesting area using various classifiers in computer networks. In the last decade, the Network Intrusion Detection System (NIDS) has been found to be associated with the extension of intelligent machine learning algorithms through Software Defined Network (SDN) (Alzahrani and Alenazi, 2021). Various machine learning and deep learning techniques have been reviewed on the basis of their capability of intrusion and malware detection systems over the network but all of them are associated with high costs of computation and resource usage.

1.2 Research question

- How to detect intrusive traffic with improved accuracy and performance over the network layer using machine learning?

1.3 Objective

- The aim of this research is to identify the network-based intrusions by evaluating and comparing the performance of different ML models. The performance of the ML models is measured based on accuracy, F-1 score and Cross-validation Score.

Rest of the report is structured as follows, chapter 2 discusses the literature about the studies that have been done in the field; based on the understanding and the literature gap present in chapter 2, chapter 3 introduces the reader to the methodology that has been followed through the study; chapter 4 discusses the implementation of a set of ensemble classifiers while chapter 5 discusses the design specification part and chapter 6 discusses the results obtained from the study; chapter 7 concludes the study with discussing the future work that can be carried out with the evidence of the results.

2 Related Work

2.1 Overview

In recent days, the advent of data science and information technology has provided key insights into the development of machine learning technique that helps in the data security approach and this type of IoT service can be utilized to detect various types of malicious attacks. This segment of the study is going to analyze the application of machine learning and the deep learning approach to the provision of network-centric data security and the detection of intrusion. The machine learning models can be helpful to analyze the patterns of the cybersecurity system and can help in the prevention of similar types of attacks.

2.2 Previous literature

In accordance with a study by Javaidet al., 2016, it can be stated that the Network Intrusion Detection System (NIDS) can help to administer and detect network security breaches by the analysis of the network traffic entering the network devices. Various machine learning techniques using Support Vector Machine, Artificial Neural Network, Random Forest, and Self-Organized Maps have been detected in this paper as the classifiers to distinguish normal traffic from anomalous traffic to extract the dataset in the NIDS process. The deep learning-based approach has been proposed and suggested by the researcher in the paper to use as the benchmark in the network intrusion dataset in the evaluation of the accuracy of anomaly detection.

Another paper by (Ford and Siraj, 2014) revealed the issues with the machine learning techniques and model in the cybersecurity and network intrusion detection system as diverse machine learning approaches have successfully been deployed in a wide range of cybersecurity issues in various organizations. This literature depicted that although machine learning models can be used as effective tools in the network intrusion detection system, various classifiers may be themselves vulnerable to malicious attacks including various classifiers such as Logistic Regression, Random Forest, SVM, BART, CART, and NNets and their error rates are described by the following figure 1.

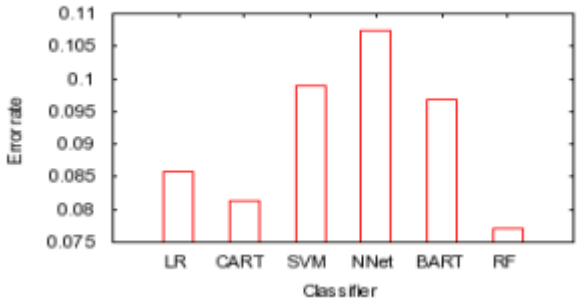

Figure 1: Error rates of machine learning classifiers in data security systems

(Source: Ford and Siraj, 2014)

Research has depicted that due to the rapid growth of network traffic, the early approach of rule-based methods has been eliminated due to their time-consuming manner and the machine learning models can be proposed by the use of computer algorithms in the context of intrusion detection. The approach to the machine learning-based feature selection in the network intrusion detection system in the context of data security must consist of four subsequent steps including the selection of dataset and pre-processing, selection of features, selection of models, and evaluation. Two feature selection methods including RF-FSR and RF-BER have been proposed as the novel ensemble of decision tree-based algorithms (Al-Jarrahet al., 2014).

2.3 Current literature

The concept of a Network Intrusion Detection system

According to a study by Ahmad et al., 2021, the Intrusion Detection System (IDS) can be referred to as the tool that is supposed to prevent the network from any probable intrusion by detecting the network traffic and assuring its confidentiality and integrity. Recently, with the advent of data science, the use and application of machine learning and deep learning method have been proposed as potential solutions to network intrusion considering the constant monitoring of the network traffic for any malicious or suspicious attacks. It can be referred to as any type of unauthorized access to the computer network system for the detection of any illegal activity.

Various research studies have acknowledged the Network Intrusion Detection System (NIDS) as the key security approach by the use of technology to detect vulnerability by exploiting various approaches of machine learning. The NIDS framework is given in the below figure 2. It is the concept of the detection of malicious traffic on a network requiring promiscuous network access by the use of passive devices which do not tend to interfere with the traffic they are supposed to monitor. It provides a complete framework for the network security approach including fur key principles such as delay, deterrence, denial of a breach, and detection (Sultana et al., 2019).

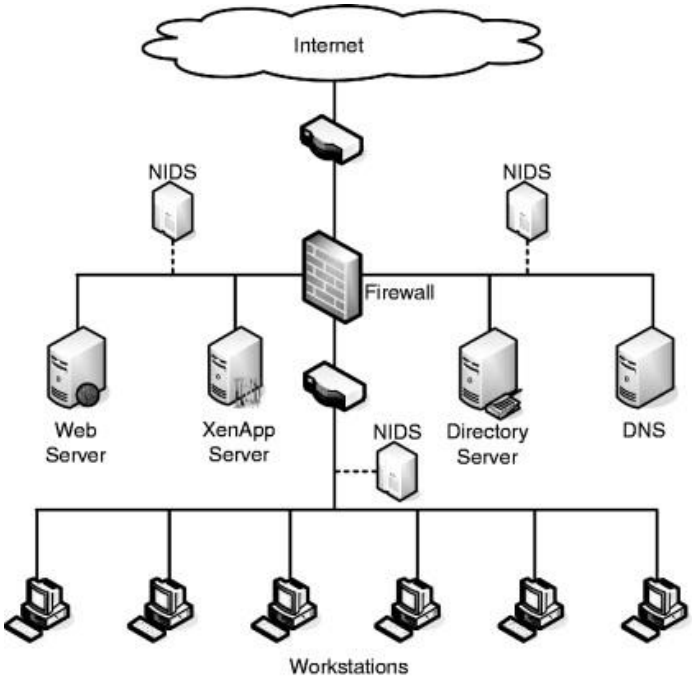

Figure 2: Framework of the NIDS

(Source: sciencedirect.com, 2017)

According to a study by Papamartzivanoset al., 2019, it can be stated that Network Intrusion Detection is one of the significant elements in the security of the ICT infrastructure achieving a high-level detection rate at the acceptable level. However, this system may be suffered from the loss of agility and thus, it must be kept up-to-date with any chance of efficiency drop to protect against cyberattacks by building a self-adaptive feature in an automated manner.

Detection of intrusive traffic with improved accuracy and performance over the network layer using machine learning

As technology has advanced massively and industrial revolution has happened, so has the Internet of Things, computer security, fog computing and cyber-attacks have also evolved exponentially on a larger scale (Abdullahiet al. 2022). The rapid development of IoT helps to follow the standard networking approaches by addressing different cybersecurity threats and maintain the promising approaches so that applicable categorization of the AI methods and its related support plan through improved accuracy planning can be maintained and followed. In-depth investigation in this prospect helps to follow the necessary ideas regarding coordinated initiations towards networking layers and maintain the intelligent architectural frameworks which can be highly appropriate to maintain standard security assistance and process with different methods. Neural network with its process plans further connects with efficient memory directions and considers the helpful insights which directs the helpful ideas about detection regarding any threat by managing potential investigation planning.

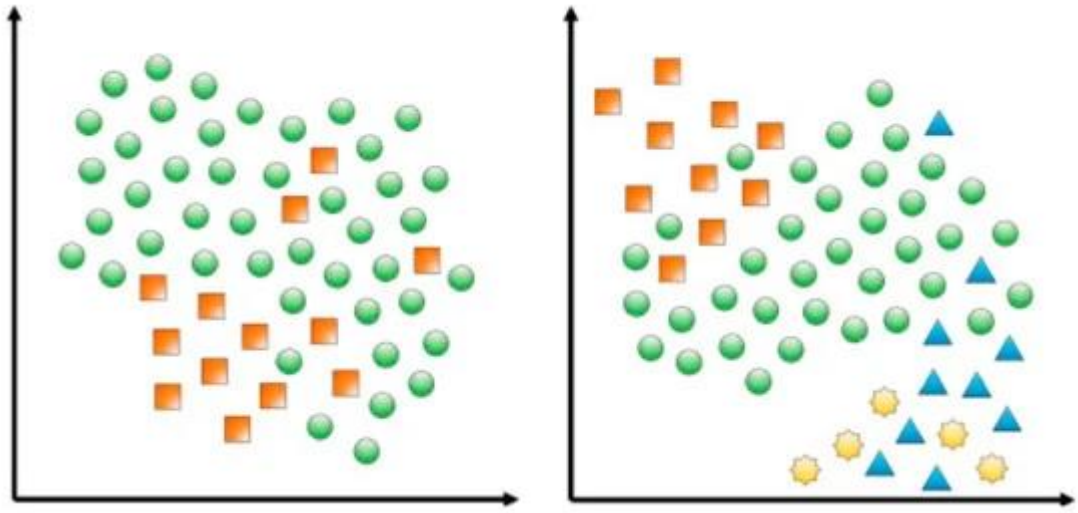

Figure 3: Classification

Source: (Abdullahiet al. 2022)

Network centric therapy for machine learning classification:

The conformal wearable and wireless inertial sensor helps to enable expanded utility for progressive evolution of gait specific quantification. The acquired signal data like gyroscope further process with wirelessly transmitted to secure Cloud Computing environment for subsequent post-processing. Further, amalgamation of machine learning helps to classify the necessary direction towards applied action distinction process related to hemiplegic gait in the segment of the hemiplegic impacted leg and the unaffected leg (LeMoyne and Mastroianni, 2020). The integration here connects with technologies that enable the appropriate realization of Network Centric Therapy for that the patient asserts different benefits so that quantification through extreme lightweight conformal wearable and wireless inertial sensor system specific profiles relative aspects get accessed. Furthermore, the clinical rehabilitation team remotely instills therapy optimized with the augmented acuity of this machine learning. Machine learning processes with the ideas regarding hemiplegic affected leg and the unaffected leg during the gait.

Application of machine learning approach to network-centric data security

Despite an increase in the number of cyber security budgets as well as complex architectures of security, most of the companies are still struggling in order to maintain the right level of threat visibility as well as ability to detect the various attacks and malware. The number of challenges and issues are growing day by day because of the various devices which are outdated IT control, shadow IT, IOT, and many other devices which grow year after year. The common denominator of the various devices in the organization workplace is the network traffic, which helps in making the network centric security the most appropriate and effective approach in defending the modern dynamic environment of the organization. There are mainly 2 major approaches for network security which are flow analysis and content analysis. The content analysis is known to be one of the expensive as well as complicated because of the storage and computational resources requirements which are combined with

the various challenges of strong traffic encryption. The flow data analysis leverages the meta data of the traffic such as protocols, source, destination and many more and helps in focusing on the various patterns of the traffic and on the changes in the communication network behavior for the purpose of detecting advanced and harmful attacks as well as compromised endpoints (Bhatia et al., 2019).

Applying machine learning to the network traffic helps the solution of network security in order to improve the detection of the advanced attacks and threats that may target the overall range of the network regarding the various connected devices. Out of three base machine learning models such as supervised, semi supervised and unsupervised learning , the Bitdefender Network Traffic Security Analytics (NTSA) relies on the semi supervised model of learning in order to provide accurate threat and attack detection as well as in real time. NTSA's semi-supervised machine learning model does not depend on the labeled training data alone. Beside samples, the data of labels helps in identifying key patterns as well as trends in the live of data flows without the help or need for the input of the human. The application of machine learning in network centric data security helps in providing visibility into any kind of unusual activities or anomalies, helps in automatic fine tuning of the analytics of the behaviour without the dependence on the knowledge about the previous threats or attacks (LeMoyne and Mastroianni, 2020).

IDS can be referred to as an application of software to detect any type of network intrusion and malicious attacks by the utilization of machine learning algorithms supervising any unauthorized access and malicious activities (Gurunget al., 2019). Various algorithms of machine learning must be associated with its application in the data security approach including the incorporation of the Decision tree, KNN, ANN, SVM, and Ensemble Methods. An IDS provides a key significance in data security by scanning the network system for any type of policy breaching and harmful activity through core supervision in case of any unauthorized access (Jiang et al., 2020).

2.4 Literature gap

The current segment of the literature review has successfully discussed and analyzed the applicability of machine learning models in the data security and network security approach in terms of the application and exploitation of a Network Intrusion Detection System. There are some limitations in traditional intrusion detection system of false alarming and not able to detect the novel attacks. Some existing machine learning models are not able to identify the attacks because of resource efficiency, time and memory usage complexity, and hardware overhead etc.

2.5 Research Niche

In the below table 1 the literature review is summarised based on ML models, accuracy and datasets used in these given literatures.

Table 1: Summery of Literature Review

Reference	Proposed model	Dataset Used	Accuracy%
Abel A. Reyes (2020)	Two stages of ML algorithms using NB, RF, Bagging, XGBoost	Aegean Wi-Fi Intrusion Dataset (AWID)	99.42
Sydney Mambwe Kasongo (2020)	Standard machine learning (ML) methods including Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), and k-Nearest Neighbor are compared to the WFEU-FFDNN (kNN)	UNSW-NB15 and the AWID	99.77
Abdul Rehman Javed & Saif ur Rehman (2021)	CNN	CAN	10% improvement to conventional models
MingJian Tang & Mamoun Alazab (	Big Data, time series, cyber security, ARCH model	National Vulnerability Database (NVD)	Data from the investigated multivariate time series have been successfully pre-whitened using GARCH models.
Alzahrani & Alenazri (2021)	XGBoost	NSL-KDD	95%
Fang, Tan & Wilbur (2020)	SVM	KDDCUP99	87.3%
Gao et al (2019)	DT, RF, kNN, LR,	KDD Test+	81.6% with DNN

	SVM, DNN, AdaBoost		
Liao & Tao (2021)	Autoencoders and Generative Adaptive Networks (GAN)	UNSW-NB15 & CICIDS 2017	98% on UNSW-NB15 and 84% on CICIDS 2017

3 Research Methodology

This section describes the methodology that has been used to implement the study. The methodology that has been used is Knowledge Discovery in Database (KDD) since the business deployment of the model is not required. This methodology is modified to support the research. Modified KDD involves data mining through the following steps. Below in figure 4 the flow diagram of methodology is given.

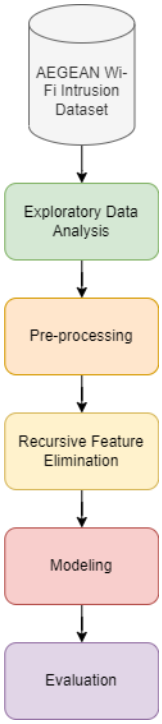

Figure 4: Flow Diagram of Methodology

Dataset: The AWID3 dataset is used in this study which is released in 2021 (Chatzoglou, E., Kambourakis, G., & Kolias, C. (2021)) and this is new version of AWID dataset. The dataset consists of 8 CSV files that contain 50000 samples and 33 attributes each related to the

attacks pertaining to the Wi-Fi network which are listed below. The data is acquired from the University of AEGEAN website[1].

1. Deauth
2. SQL Injection
3. Disass
4. RogueAP
5. Botnet
6. Krack
7. Malware
8. SSH

As the machines find it difficult to process non-numerical values of the data, encoding has to be done on the dataset as it contains string values for some attributes.

0	frame.encap_type	400000 non-null	int64
1	frame.len	400000 non-null	int64
2	frame.number	400000 non-null	int64
3	frame.time	400000 non-null	object
4	frame.time_delta	400000 non-null	float64
5	frame.time_delta_displayed	400000 non-null	float64
6	frame.time_epoch	400000 non-null	float64
7	frame.time_relative	400000 non-null	float64
8	radiotap.channel.flags.cck	400000 non-null	int64
9	radiotap.channel.flags.ofdm	400000 non-null	int64
10	radiotap.channel.freq	400000 non-null	int64
11	radiotap.dbm_antsignal	400000 non-null	object
12	radiotap.length	400000 non-null	int64
13	radiotap.present.tsft	400000 non-null	object
14	radiotap.rxflags	400000 non-null	object
15	radiotap.timestamp.ts	400000 non-null	int64
16	wlan.duration	400000 non-null	int64
17	wlan.fc.ds	400000 non-null	object
18	wlan.fc.frag	400000 non-null	int64
19	wlan.fc.order	400000 non-null	int64
20	wlan.fc.moredata	400000 non-null	int64
21	wlan.fc.protected	400000 non-null	int64
22	wlan.fc.pwrmgt	400000 non-null	int64
23	wlan.fc.type	400000 non-null	int64
24	wlan.fc.retry	400000 non-null	int64
25	wlan.fc.subtype	400000 non-null	int64
26	wlan.ra	400000 non-null	object
27	wlan_radio.duration	400000 non-null	int64
28	wlan_radio.channel	400000 non-null	int64
29	wlan_radio.data_rate	400000 non-null	float64
30	wlan_radio.frequency	400000 non-null	int64
31	wlan_radio.signal_dbm	400000 non-null	int64
32	wlan_radio.phy	400000 non-null	int64
33	Label	400000 non-null	object

Figure 5: Dataset Attributes

Feature Vector: The data from the files are combined to obtain a single large dataset. This is done by appending the data from the files and samples are denoted with classes corresponding to each type of attack. This forms the feature vector that has been utilized in the study.

EDA: Exploratory Data Analysis (EDA) helps an analyst visually inspect the data to obtain important features from the dataset. It also helps the analyst the type of data that is present in the dataset. Using the EDA an analyst can select the operations such as 'encoding' that need to be performed on the data in order to process it successfully and reliably.

Data Pre-processing: Data pre-processing is an important step in developing a model. It makes the data suitable for the model to perform operations on it. As the machines find it difficult to process non-numerical values of the data, encoding has to be done on the dataset

[1] https://icsdweb.aegean.gr/awid/download-dataset

as it contains string values for some attributes. Encoding maps a non-numerical value to a numerical value.

Feature Selection: To get the final features from the dataset the method of Recursive Feature Elimination (RFE) has been incorporated to reduce the number of features. It involves finding the best features by developing a base classifier to obtain the highest accuracy possible. In this study, decision tree classifier has been used to achieve RFE.

Modeling: Once the final pre-processed feature vector is generated, the data is modeled using ensemble classifiers such as Bagging, AdaBoost, Random Forest, Extra Trees, Gradient Boosting, Isolation Forest, Stacking, and Voting which are evaluated based on three metrics viz. accuracy, f1-score, and cross-validation score.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Number\ of\ Total\ Predictions\ performed}$$

F1-score is given as the harmonic mean of precision and recall values of the classifier whereas the cross-validation score is obtained by testing the classifier on the training data itself.

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall}$$

4 Design Specification

This section describes the architecture of the model. Figure 6 below depicts the architecture of the implementation of the study. The dataset and the classification results are presented in the presentation tier. The processing performed during the classification is present in the Background tier of Business Logic.

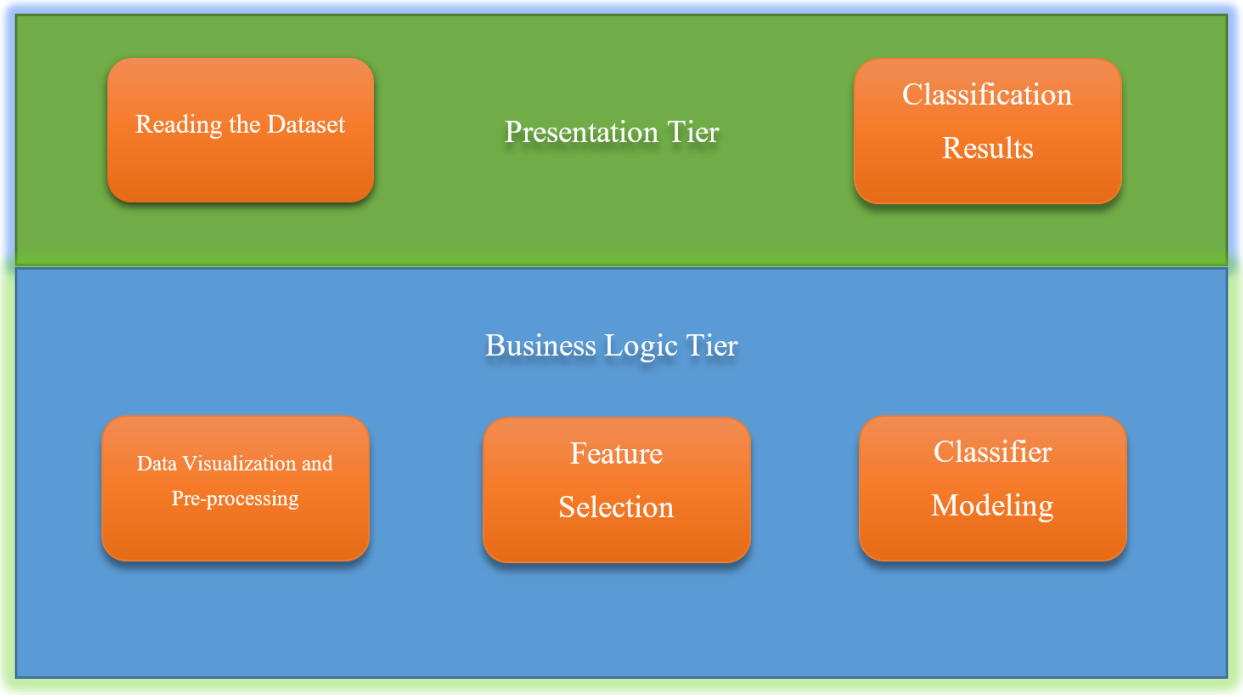

Figure 6: Implementation architecture

5 Implementation

This section discusses the implementation of the methodology discussed above. It explains all the necessary components of the implementation in the given steps:

- This study is implemented in python programming language using the Jupyter Notebook computing platform.
- First, all the required libraries are imported. The models are implemented using the "sklearn" library of Python.
- By using the pd.read_csv function we imported all 8 attacks from AWID3 dataset. Then, all of the data is first combined using the concat() function of the pandas data frame.
- By using the isnull function we checked the number of missing values in each attribute of dataset. After that we removed the attributes which having null values with the help of dropna function.
- By using label encoding we convert all string and object types of data into numeric type with the help of the function fit_transform. In last we got the total 33 attributes which having the valid data.
- In the feature selection process, Recursive Feature Elimination is used to select the features using Decision Tree classifier and running a pipeline to find the best features. The Repeated Stratified K-Fold cross-validation is used to check to validate the features selected.
- After that the data is split into two sets, the training set which contains 70% of the samples, and the testing set containing 30% samples. This split data are subjected to classification using Bagging Classifier, AdaBoost Classifier, Random Forest Classifier, Extra Trees Classifier, Gradient Boosting Classifier, Isolation Forest Classifier, Stacking Classifier, and Voting Classifier. The implementation and the discussion of their results are as follows.

5.1 Implementation of Bagging Classifier

The bagging classifier is an ensemble classifier that fits base classifiers on the subset of the data. These subsets of the data are random. Their individual predictions are used to find the final prediction based on either by voting or by averaging. The bagging classifier in the study is implemented using the BaggingClassifier() function of the sklearn. The base classifiers that are implemented are Decision Tree classifiers.

5.2 Implementation of AdaBoost Classifier

AdaBoost classifier works by correcting the weights of the incorrectly classified samples to aggregate the final prediction. The base estimator is applied on the data to find incorrect predictions and the subsequent copies of the base estimators then weighs of the incorrectly classified instances are corrected. The final prediction model is a weighted sum of the prediction of these individual predictions. The AdaBoost classifier in the study is

implemented using the AdaBoostClassifier() function of the sklearn. The base estimator used in the study is a Decision Tree Classifier with a max depth of 1. The AdaBoost classifier in sklearn makes use of the SAMME method for weighing. The number of base estimators that are used in the study is 50.

5.3 Implementation of Random Forest Classifier

Random forest is another ensemble of classifiers that works by fitting decision trees on sub samples of the data and averaging their predictions. The random forest classifier in the study is implemented using the RandomForestClassifier() function of sklearn library. The number of estimators used for the classification is 100.

5.4 Implementation of Extra Tree Classifier

The extra trees classifier is similar to the Random forest classifier in which the decision trees are randomly fit on the sub-samples of the data. The final prediction is obtained by averaging the predictions of these individual trees. The extra trees classifier is implemented using the ExtraTreesClassifier() function of the sklearn library. The number of estimators for the classifier is 100.

5.5 Implementation of Gradient Boosting Classifier

The gradient boosting classifier involves implementing logistic regression classifiers in stage-wise method. The aim of the gradient boosting classifier is to optimize the log loss function. Gradient boosting classifier in the study is implemented using the GradientBoostingClassifier() function of the sklearn library. Number of estimators in gradient boosting classifier is 100.

5.6 Implementation of Isolation Forest Classifier

Isolation forest classifier detects the anomaly in the dataset by randomly selecting the sub-samples and calculating their anomaly score. The final prediction is based on the calculations. The isolation forest classifier that has been implemented in the study has 100 base estimators. It is implemented using the IsolationForest() function of the sklearn library.

5.7 Implementation of Stacking Classifier

Stacking classifier stacks the output of the individual classifiers in order to obtain the best classification. The final classifier is selected based on the cross validation of the base estimators. Stacking classifier is implemented using the StackingClassifier() function. The individual estimators that are stacked for implementing it are random forest with 10 estimators, support vector regressors (SVR), and AdaBoost classifier. Pipelining is done using the steps = ‘Pipeline()’ function to get the final estimator.

5.8 Implementation of Voting Classifier

Voting classifier identifies the best classifier of the ensemble by voting for the classifiers with best prediction. This voting can be hard or soft. Hard voting takes into account the majority votes for classification while the soft voting uses the probabilities of the classes that are present in the data. Study makes use of the VotingClassifier() function of the sklearn library to implement the voting classifier. The individual estimators that have been used in the classifier are AdaBoost, Random Forest, and Gradient Boosting with 'Hard' voting type for best estimator selection.

6 Evaluation

This section evaluates the performances of the classifiers that have been implemented in the study. These classifiers are evaluated based on the evaluation metrics mentioned in past sections. Confusion matrix which represents the reliability of the classifier is also given with the classifier results.

6.1 Bagging Classifier

An accuracy of 100% is obtained for the bagging classifier. It also achieved a precision, recall, and F1-score of 100%. The Cross-validation result of 100% shows that the classifier can equally classify all the test data samples correctly. The bagging classifier successfully classified all the intrusion types.
Confusion Matrix is shown in figure 7:

```
array([[15049,     0,     0,     0,     0,     0,     0,     0],
       [    0, 14993,     0,     0,     0,     0,     0,     0],
       [    0,     0, 14882,     0,     0,     0,     0,     0],
       [    0,     0,     0, 14788,     0,     0,     0,     0],
       [    0,     0,     0,     0, 15012,     0,     0,     0],
       [    0,     0,     0,     0,     0, 15249,     0,     0],
       [    0,     0,     0,     0,     0,     0, 14930,     0],
       [    0,     0,     0,     0,     0,     0,     0, 15097]],
      dtype=int64)
```

Figure 7: Confusion Matrix for Bagging Classifier

6.2 AdaBoost Classifier

The AdaBoost classifier showed an accuracy of 62%. This can be because of the underlying efficiency of the decision tree classifier as the max depth of this classifier in the default scenario is 1, to correctly classify the samples according to the classes. AdaBoost has been unable to identify three types of attacks viz. SQL injection, Krack and SSH. Other metrics that are obtained from the results are precision = 0.53, recall = 0.62, and cross-validation score of 41.67%.

```
array([[15049,     0,     0,     0,     0,     0,     0,     0],
       [    0,     0, 14993,     0,     0,     0,     0,     0],
       [    0,     0, 14882,     0,     0,     0,     0,     0],
       [    0,     0,     0, 14788,     0,     0,     0,     0],
       [    0,     0,     0,     0, 15012,     0,     0,     0],
       [    0,     0, 15249,     0,     0,     0,     0,     0],
       [    0,     0,     0,     0,     0,     0, 14930,     0],
       [    0,     0, 15097,     0,     0,     0,     0,     0]],
      dtype=int64)
```

Figure 8: Confusion Matrix for AdaBoost Classifier

6.3 Random Forest Classifier

Random forest classifier similar to boosting classifier showed an impressive accuracy of 100%. It successfully classified all the types of network attacks. It also showed an impressive cross-validation score of 100%.

```
array([[15049,     0,     0,     0,     0,     0,     0,     0],
       [    0, 14993,     0,     0,     0,     0,     0,     0],
       [    0,     0, 14882,     0,     0,     0,     0,     0],
       [    0,     0,     0, 14788,     0,     0,     0,     0],
       [    0,     0,     0,     0, 15012,     0,     0,     0],
       [    0,     0,     0,     0,     0, 15249,     0,     0],
       [    0,     0,     0,     0,     0,     0, 14930,     0],
       [    0,     0,     0,     0,     0,     0,     0, 15097]],
      dtype=int64)
```

Figure 9: Confusion Matrix for Random Forest Classifier

6.4 Extra Trees Classifier

The extra trees classifier similar to the random forest classifier and boosting classifier showed an impressive accuracy of 100%. It successfully classified all the types of network attacks. Similar to random forest it also showed an impressive cross-validation score of 100%. F1-score of 100% is obtained for the classifier.

```
array([[15049,     0,     0,     0,     0,     0,     0,     0],
       [    0, 14993,     0,     0,     0,     0,     0,     0],
       [    0,     0, 14882,     0,     0,     0,     0,     0],
       [    0,     0,     0, 14788,     0,     0,     0,     0],
       [    0,     0,     0,     0, 15012,     0,     0,     0],
       [    0,     0,     0,     0,     0, 15249,     0,     0],
       [    0,     0,     0,     0,     0,     0, 14930,     0],
       [    0,     0,     0,     0,     0,     0,     0, 15097]],
      dtype=int64)
```

Figure 10: Confusion Matrix for Gradient Boosting Classifier

6.5 Isolation Forest Classifier

This classifier has not been able to classify the attack samples correctly. With an accuracy of only 11%, it has struggled to detect the anomaly in the data. It also showed an f1-score of 3% and a cross-validation score of 9%. This has made the isolation forest the poorest classifier in the results.

```
array([[    0,     0,     0,     0,     0,     0,     0,     0,     0],
       [ 3804,     0, 11245,     0,     0,     0,     0,     0,     0],
       [ 1752,     0, 13241,     0,     0,     0,     0,     0,     0],
       [ 1398,     0, 13484,     0,     0,     0,     0,     0,     0],
       [14788,     0,     0,     0,     0,     0,     0,     0,     0],
       [ 2557,     0, 12455,     0,     0,     0,     0,     0,     0],
       [ 2488,     0, 12761,     0,     0,     0,     0,     0,     0],
       [ 3846,     0, 11084,     0,     0,     0,     0,     0,     0],
       [ 1745,     0, 13352,     0,     0,     0,     0,     0,     0]],
      dtype=int64)
```

Figure 11: Confusion Matrix for Isolation Forest Classifier

6.6 Stacking Classifier

The stacking classifier showed an accuracy of 99.99%. However, it has shown a low cross-validation score of 61%. This might have been due to overfitting in one of the estimators that has been used in stacking. The f1-score obtained for the classifier is also 99.99%.

```
array([[15049,     0,     0,     0,     0,     0,     0,     0],
       [    0, 14993,     0,     0,     0,     0,     0,     0],
       [    0,     0, 14882,     0,     0,     0,     0,     0],
       [    0,     0,     0, 14788,     0,     0,     0,     0],
       [    0,     0,     0,     0, 15012,     0,     0,     0],
       [    0,     0,     0,     0,     0, 15249,     0,     0],
       [    0,     0,     0,     0,     0,     0, 14930,     0],
       [    0,     0,     0,     0,     0,     0,     0, 15097]],
      dtype=int64)
```

Figure 12: Confusion Matrix for Stacking Classifier

6.7 Voting Classifier

A voting classifier votes for the best estimator, its prediction is also similar to the best estimator. As shown above, the Random Forest and the Gradient Boosting classifier showed 100% accuracy, the voting classifier also shows an accuracy of 100% to successfully classify all the attacks. F1-score and Cross-validation scores of 100% are also obtained for this classifier.

```
array([[15049,     0,     0,     0,     0,     0,     0,     0],
       [    0, 14993,     0,     0,     0,     0,     0,     0],
       [    0,     0, 14882,     0,     0,     0,     0,     0],
       [    0,     0,     0, 14788,     0,     0,     0,     0],
       [    0,     0,     0,     0, 15012,     0,     0,     0],
       [    0,     0,     0,     0,     0, 15249,     0,     0],
       [    0,     0,     0,     0,     0,     0, 14930,     0],
       [    0,     0,     0,     0,     0,     0,     0, 15097]],
      dtype=int64)
```

Figure 13: Confusion Matrix for Voting Classifier

6.8 Discussion

Results obtained for the classifiers implemented in the study are enlisted in the table 2 below. The table clearly shows that accuracy, F1-score and cross-Validation Score is 100 % in the Random Forest, Extra Trees and Voting classifiers. That means these three models are most accurate to identify the network-based intrusion attacks on AWID3 dataset. Accuracy and F1-score is 100% in Bagging classifier as well but lagged in cross-validation score behind the best models with a small margin of 1.33%. Gradient Boosting Classifier performed well with 99.999167 % accuracy, 99.999167 % F1-Score and 100.00& % cross validation score. In Stacking Classifier accuracy and F1-score is 99.999167 % but the cross-validation score is 61.38 % that means stacking classifier seems to have suffered from overfitting. In AdaBoost Classifier accuracy, F1-score and cross validation score are 62.217500 %, 54.730726 % and 41.67 % respectively which means not performing well. In Isolation Forest Classifier accuracy, F1-score and cross validation score are 11.034167 %, 3.224387 % and 9.43 % respectively. That means Isolation Forest classifier was worst performing among all of them.

Table 1: Comparison of Results for the Classifiers

Model	Accuracy	F1-Score	Cross-Validation Score
Bagging Classifier	**100.000000**	**100.000000**	98.67
AdaBoost Classifier	62.217500	54.730726	41.67
Random Forest Classifier	**100.000000**	**100.000000**	**100.00**
Extra Trees Classifier	**100.000000**	**100.000000**	**100.00**
Gradient Boosting Classifier	99.999167	99.999167	**100.00**
Isolation Forest	11.034167	3.224387	9.43
Stacking Classifier	99.999167	99.999167	61.38
Voting Classifier	**100.000000**	**100.000000**	**100.00**

7 Conclusion and Future Work

The network attacks that are being deployed over the internet are getting sophisticated as the day passes. Various companies across the domain are working on systems to mitigate these attacks. Several frameworks and software have been developed to do so but these are not precise. Attackers somehow find a backdoor to these devices compromising safety, privacy, and security of devices and systems. These attacks can hamper services provided by enterprises by invoking downtime in the systems. So, there is need of improved Network based intrusion detection system to detect the malicious activity. The key motive of this research is to proposed the ensemble based ML models along with intrusion detection system to ensure the better intrusion detection accuracy. AWID3 dataset is used in this research which is new dataset. By evaluating and comparing the different 8 models performance we found three best models such as Random Forest Classifier, Extra Trees Classifier, and Voting Classifier.

Future Work

The future of this kind of system lies in artificial intelligence and neural networks. We unquestionably require the greatest amount of data feasible in order to gather evidence, examine it, and draw assumptions about whether the system is under attack or not as well as to identify malicious activity on different networks. The implemented models can also be assessed for other datasets as well.

References

Fang, W., Tan, X. and Wilbur, D., 2020.Application of intrusion detection technology in network safety based on machine learning. Safety Science, 124, p.104604.

da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R. and de Albuquerque, V.H.C., 2019. Internet of Things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, *151*, pp.147-157.

Alzahrani, A.O. and Alenazi, M.J., 2021. Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet*, *13*(5), p.111.

Javed, A.R., Ur Rehman, S., Khan, M.U., Alazab, M. and Reddy, T., 2021. CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. IEEE transactions on network science and engineering, 8(2), pp.1456-1466.

Javaid A Y, Niyaz Q, Sun W, Alam M, 2016. *A Deep Learning Approach for Network Intrusion Detection System.* 9th EAI International Conference on Bio-inspired Information and Communications Technologies. New York. IEEE

Ford, Vitaly & Siraj, Ambareen. (2014). Applications of Machine Learning in Cyber Security. *27th International Conference on Computer Applications in Industry and Engineering*, CAINE 2014.

O. Y. Al-Jarrah, A. Siddiqui, M. Elsalamouny, P. D. Yoo, S. Muhaidat and K. Kim, *Machine-Learning-Based Feature Selection Techniques for Large-Scale Network Intrusion Detection*, IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2014, pp. 177-181.

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. and Ahmad, F., 2021. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), p.e4150.

Sultana, N., Chilamkurti, N., Peng, W. and Alhadad, R., 2019. Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, *12*(2), pp.493-501.

Papamartzivanos, D., Mármol, F.G. and Kambourakis, G., 2019. Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE Access*, *7*, pp.13546-13560.

Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L.F. and Abdulkadir, S.J., 2022. Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review. *Electronics*, *11*(2), p.198.

LeMoyne, R. and Mastroianni, T., 2020, October. Network centric therapy for machine learning classification of hemiplegic gait through conformal wearable and wireless inertial sensors. In *2020 International Conference on e-Health and Bioengineering (EHB)* (pp. 1-4). IEEE.

Bhatia, R., Benno, S., Esteban, J., Lakshman, T.V. and Grogan, J., 2019, December. Unsupervised machine learning for network-centric anomaly detection in IoT. In *Proceedings*

of the 3rd acm conext workshop on big data, machine learning and artificial intelligence for data communication networks (pp. 42-48).

Gurung, S., Ghose, M.K. and Subedi, A., 2019. Deep learning approach on network intrusion detection system using NSL-KDD dataset. *International Journal of Computer Network and Information Security*, *11*(3), pp.8-14.

Jiang, F., Fu, Y., Gupta, B.B., Liang, Y., Rho, S., Lou, F., Meng, F. and Tian, Z., 2018. Deep learning based multi-channel intelligent attack detection for data security. *IEEE transactions on Sustainable Computing*, *5*(2), pp.204-212.

Chatzoglou, E., Kambourakis, G., & Kolias, C. (2021). Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset. IEEE Access, 9, 34188-34205.

A. Reyes, A., D. Vaca, F., Castro Aguayo, G.A., Niyaz, Q. and Devabhaktuni, V., 2020. A machine learning based two-stage Wi-Fi network intrusion detection system. *Electronics*, *9*(10), p.1689.

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

Javed, A.R., Ur Rehman, S., Khan, M.U., Alazab, M. and Reddy, T., 2021. CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE transactions on network science and engineering*, *8*(2), pp.1456-1466.

Tang, M., Alazab, M. and Luo, Y., 2017. Big data for cybersecurity: Vulnerability disclosure trends and dependencies. *IEEE Transactions on Big Data*, *5*(3), pp.317-329.

Fang, W., Tan, X. and Wilbur, D., 2020. Application of intrusion detection technology in network safety based on machine learning. *Safety Science*, *124*, p.104604.

Gao, X., Shan, C., Hu, C., Niu, Z. and Liu, Z., 2019. An adaptive ensemble machine learning model for intrusion detection. *IEEE Access*, *7*, pp.82512-82521.

Tao, X.L., Liu, Z.Y. and Yang, C.S., 2021. An efficient network security situation assessment method based on AE and PMU. *Wireless Communications and Mobile Computing*, *2021*.

Feng, G. and Buyya, R. (2016). Maximum revenue-oriented resource allocation in cloud, *IJGUC* 7(1): 12–21.

Gomes, D. G., Calheiros, R. N. and Tolosana-Calasanz, R. (2015). Introduction to the special issue on cloud computing: Recent developments and challenging issues, *Computers & Electrical Engineering* 42: 31–32.

Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* 46(1): 79–105.