Applying Machine learning and Deep Learning Techniques for Improvement in Network Intrusion Detection System

MSc Research Project
MSc Cybersecurity

Chaitanya Londhe
Student ID: X19212518

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Chaitanya Anand Londhe		
Student ID:	X19212518		
Programme:	M.Sc. Cybersecurity	**Year:**	2020-2021
Module:	Academic Internship		
Supervisor:	Mr. Imran Khan		
Submission Due Date:	16th August 2021		
Project Title:	Applying Machine learning and Deep Learning Techniques for Improvement in Network Intrusion Detection System		
Word Count:	7450	**Page Count:**	24

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: Chaitanya Londhe

Date: 16/08/2021

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):	

Applying Machine learning and Deep Learning Techniques for Improvement in Network Intrusion Detection System

Chaitanya Londhe
X19212518

Abstract

The quick progress in the web and networking domains has taken place by means of a huge growth of the network size and data. Moreover, attackers might be spotted within the network with the intention of initiating numerous attacks. The Intrusion Detection System (IDS) is a system that monitors network traffic and ensures integrity, confidentiality, and availability from network intrusion. Even despite the greatest efforts of academics, IDS faces problems till date in developing identification accuracy while minimizing false alarms and detecting new attacks. Machine learning (ML) and Deep Learning (DL) based IDS software were recently deployed as viable techniques of detecting network breaches rapidly. The purpose of this study is primarily focused on the prominent ML and DL techniques utilized in the modeling of Network-based IDS (NIDS) architectures. The recommended technique, performance measures, data gathering and recent breakthroughs in ML and in the DL of Network Intrusion Detection System (NIDS) are reviewed. Many research problems have been identified, and possible research opportunities have emerged with the limits of the existing technique in creating ML and DL based NIDS.

Keywords: Network Intrusion Detection System, Machine Learning/Deep Learning.

Table of Content

1 Introduction

A vast number of essential data is generated and exchanged by the nodes in the network. The protection of these storage and network nodes has become burdensome because of the emergence of a vast range of modern threats, either as a replication of an existing intrusion or as a completely new intrusion. For a business, the data node might be highly important. Any leaking of the details of the node might negatively affect the corporate impression and financial issues of the organization. Ongoing IDSs have shown incompetency to recognize a range of attacks, including zero-day threats and reducing the False Alarm Rates (FAR). In the final analysis, this creates the demand for a reliable, precise, and affordable NIDS.

In line with the demands for effective IDS, researchers discovered the use of ML and DL methods. ML and DL are part of artificial intelligence (AI) that aim at extracting meaningful insights from vast quantities of data. ML and DL are both techniques of gathering meaningful insights from network traffic and of anticipating routines and pattern-based abnormal behaviors. The ML-based IDS relies completely on feature engineering to analyze essential facts from network traffic. On the contrary, DL-based IDS does not focus on feature engineering; owing to its complex structure, it is adept at recognizing the complicated functioning of raw data.

During the last 10 years, professionals have created several ML and DL techniques to improve the effectiveness of the NIDS in recognizing malware assaults. There is indeed a significant amount of area for investigation into adding DL approaches to NIDS to successfully identify perpetrators on the network. This research is therefore exploitable across NIDS. NIDS are basically located inside the network at very precise strategic places for the monitoring of traffic into and out of all network-wide components. When the unparalleled behaviour is discovered, the Network Administrator is alerted immediately. NIDS can also be classed in two different kinds, for example Online and Offline NIDS. Online NIDS handles network real-time data, while the off-line NIDS monitors saved data to determine whether or not the attack is occurring.

The objective of this project is to get an overview of current developments and breakthroughs in ML and DL methods for NIDS. The main purpose is to provide new experts interested in delving into this growing development, including updated details about latest ML and DL based NIDS. Several academic research articles have explained how to construct the IDS. This paper provides more up-to-date information and new advances in the AI based NIDS architecture. The purpose of this research project is to provide researchers with more upgraded information on a particular region on an AI-based NIDS from which they can seek out new trends and possibilities to study the domain.

Table 1 from section 2, provides a comprehensive comparison of this paper with other review articles. The following section is arranged as follows: The methodology and the initial process are described in section 3. The design specifications and architecture are described in

Section 4. The ML and DL methods proposed are outlined in section 4. Section 5, Section 6 show the evaluation of the research with the metrices used and this research paper ends with Section 7 that includes Conclusions, limitations, and future scope for the development of NIDS.

Research Question:

How can ML and DL techniques be used to make improvements in network intrusion detection system?

As computing has become more interconnected, the responsibility of keeping networks secure and free of vulnerabilities has increased. Each year, these attacks cost billions and result in the destruction of vital infrastructure. As a result, cyber protection, especially NIDS, has become more necessary. Furthermore, there is no study on the implications of DL techniques and next-level analysis to see whether there would be an increase in performance over conventional ML techniques. Therefore, the main objective of this paper is to find the latest developments or trends in the architecture of ML and DL based NIDS, the new methodologies for NIDS design using ML and DL, the future scope of artificial intelligence in the betterment of NIDS. The investigation of this question is important in terms of knowledge gain for the researchers in this field, as it will provide detailed methodology and information regarding different ML and DL techniques that would be feasible for improving the NIDS. The answer to this question will give more insights about the topic and may help to upgrade the technologies of NIDS.

2 Related Work

After a comprehensive study of the research papers below, it can be said that these are various drawbacks in these works. By mainly focusing on some of the relevant papers, it can be said that, even if the Deep Neural Network (DNN) approach how been utilized by Wang et al. in his research work, the dataset set used is not the updated one and the approach used is for high dimensions. In addition, the sparse autoencoder technology and supporting vector machine have also been used in study of Yan El, however their dataset is not updated, also for R2L and U2R attack, it is going a reasonable detection, but it is still very less as compared to other attacks and no other DL models are used. In the majority of the following research articles, the Support Vector Machine (SVM) is widely used. In addition, few of them are using DNN technologies, which might be highly important for the issue.

The current study **(García-Teodoro *et al.*, 2009),** reviews the ideas and fundamental operational design of significant A-NIDS systems and allocates a category based on the nature of the processing in relation to the 'behavioral' concept of the system. Another important finding of this study is that it highlights the main features of various IDS systems available to the public. At last, A-NIDS lists the most frequent difficulties with specific emphasis paid to the assessment issue. In IDS research and development, the material given is an excellent starting point.

This article is designed **(Shah and Singh, 2012)** to investigate our suspect behavior, Snort and WinPcap, via the instruments of our IDS. Snort is a standard system for intrusion detection of network (NIDS) monitoring and comparing data packets with a central database threat signature that must be regularly changed. Warnings from Snort are typically seen utilizing BASE, a core research and safety engine.

The study **(Nevlud *et al.*, 2013)** focuses on network identification of anomalies. Network anomalies relate to stuff that doesn't operate properly. Anomalies were identified using ML methods. ML can be regarded as an addition or a subdivision of artificial intelligence. A ML algorithm usually starts from some input and related data structure to analyze, assess, and validate the knowledge learnt There are several approaches for ML. Testing was carried out on the Bayesian networks and Decision Tree (DT). WEKA has been used to evaluate recognizing, clustering, and connecting algorithms and display results, as a web-based data mining tool. WEKA is a range of ML algorithms linked to data mining.

In this paper, a DL technique for creating an optimized and efficient NIDS was proposed **(Javaid *et al.*, 2016)**. The Self-Training Learning (STL) approach was utilized in NSL-KDD, a standard data set for system interference. The success of the technique was described and a few previous research were carried out. Parameters were examined with accuracy, recall, precision and f-measure values. Also, a soft-max regression (SM) technique was used for 2-classes and 5-classes. For the 2 class, the accuracy of STL was lower than that of SM. The accuracy values for STL and SM range from 85.44% to 96.56%, respectively. In comparison to SM, however, STL had better recall results. STL and SM recall results are 95,95% and 63,73% respectively. STL also exceeded SM because of an excellent recall value for the f-measure value. STL obtained 90.4% f-measure while SM reached just 76.8%. Similar comments were raised for the 5-class. For STL and SM, the f-measure values are 75.76% and 72.14% respectively.

This study **(Dong and Wang, 2016)** provides a comparison of profound learning approaches for network intrusion detection. In this paper, autoencoder can be classified with 98.9% accuracy by the attack types. The LSTM model, in comparison, generated a 79.2% rating. Further hyperparameter adjustment may be necessary to increase the LSTM model's accuracy. Benefiting from the formation set, the prediction of such models might lead to a less accurate level of class imbalance. The self-taught learning model decreases the number of features in the autoencoder to 10 by the reduction in dimensionality. This results in a higher level of accuracy compared to the SMR findings achieved on the purified NSL-KDD Dataset, with all 41 variables in the original dataset being much lesser than 75.23%. So, it may infer that the DL autoencoder is a suitable NIDS model.

In this research, the DL method is used **(Nguyen Thanh Van, Tran Ngoc Thinh, and Le Thanh Sach, 2017)** to apply NIDS depending on anomalies. Comparison of the anomaly and the group classification in just the same RBM methodology is done, where the anomaly classification is lesser, while the slope in the training process has minimal oscillations. The two RBM and Autoencoder methods compared, indicate that in the classifications of four attack groups and normal, Autoencoder is much superior to RBM with 0.02 and 0.13 respectively. The comparative analysis of stacked RBM and stacked Autoencoder execution times in the same direction is performed and it takes longer for Stacked Autoencoder to do so. While the autoencoder's cost function is simpler, considerably more computations are carried out in the training phase to reduce the dimensionality.

This article **(Kim and Aminanto, 2017)** examines several strategies, notably bio-inspired ones, in order to make the field of IDS science more important. It is thought that the present procedures may be upgraded by following the guidance of nature. The beginning of this article is the observation of Ant behavior, and the clustering technique subsequently uses the

Ant clustering algorithm. On the other hand, various techniques were designed to improve IDS efficiency. The capacity of ACA to distinguish between innocuous and assault instances is believed to be low. Consequently, the newest development of the neural system was moving away from ACA and towards more current bio-inspired technologies, such as DL. DL techniques as an algorithm would be challenging to execute in real time. Either retrieval or complication reduction were the findings of prior studies which utilized DL methods in the IDS scenario. In addition, DL techniques may also accomplish grouping jobs. 7 In summary, SAE is very helpful for applications like as clustering, feature extraction, and classification.

Three concepts, a standard DNN, a Self-Taught Learning (STL) and a Recurrent Neural Network (RNN), are compared to precision and performance, depending on a Long Short-Term Memory (LSTM) in this research **(Lee and Amaresh, 2018)**. Knowledge Discovery in databases intrusion data systems are used to evaluate their findings. In conjunction with KDD Cup 1999 these data were used for the 3rd International Competition for Information Exploration and Data Mining. The results were compared to a typical deep model, using multinomial logistics regression, to evaluate whether DL algorithms outperform deep algorithms with this model.

In this post **(Aminanto and Kim, no date)**, an examination of past IDSs is provided using DL approaches. DL methods are used by firewalls. Such methods to DL are then reviewed and their advantages and disadvantages are evaluated so that you can understand more about why you can utilize DL. It is acknowledged that there is still a mistake concerning how a DL software may better be implemented. This study states that IDS is useful in DL, especially in reducing dimensionality. To support this argument, the implementation of DL in IDS is subject to possible problems and advice. Finally, in prospective research on detection of unknown threats, DL models can aid.

This article **(Karatas, Demir and Koray Sahingoz, 2018)** provides a glimpse of the IDS technique for DL by giving a work in the literature. It offers an overview of several methods to profound IDS by delivering an overview of profound knowledge and IDS principles. The most popular datasets utilized in this work are KDD Cup99, NSL-KDD and CSE CIC-IDS2018. A comparison is also made between the implementation of IDS applications based on profound knowledge, which also compare ML and DL techniques.

In this research paper, a mixed and organized IDS has been proposed **(Çavuşoğlu, 2019)** to provide increased intrusion detection in a range of intrusion-scenarios using a mix of ML and feature selection and evaluation techniques. In the existing context, the NSL-KDD dataset is normalized and subsequently the schema volume is lowered utilizing a range of feature selection processes. Two novel techniques have been created for the selecting process. All sorts of attacks are observed to be high in precision and have low FPR values. The optimized parameters for the system suggested for the R2L attack type are 0.94 and 0.91 according to the DR and TPR criteria. The result calculated in the DR and TPR criterion are likewise found to be quite high in all forms of attacks. The minimum value in R2L type is determined in the F-Measure and MCC criterion. When evaluating the runtime in relation to attack types, a very short processing time is identified for a U2R attack type as well as a long attack type for R2L is related to the application of a stacking structure than in other attack types.

In this study, the usual data set of KDD99 is analyzed and evaluated **(Li *et al.*, 2019)**. Methodologies have been presented in this paper for the identification of Android malware, and the use of ML methods have been exhibited for this purpose, with a focus on SVMs. The dataset utilized in SVM malware identification was created using authorization requests, API calls, application categorization and descriptions. The study compared the SVM and DroidRisk effectiveness on the dataset and concluded that the SVM has a clear edge over DroidRisk. SVM-RFE was utilized in the assessment of characteristics and the recognition of their contributions therefore to improve SVM performance. The removal of non-contributing characteristics resulted in a 94,15% accuracy.

This paper **(Peng *et al.*, 2019)** provides an intrusion detection mechanism for deep network learning. The KDD CUP99 data from Lincoln Laboratory of the Technology Institute of Massachusetts has been tested. The results show that the recommended technique is more accurate than standard ML. In the bigger DBN data set, S4 is an 11.58% greater DBN-based functional learning technique than PCA and 12.91% greater than that of the gain. The DBN-based learning method is indeed more suited to high-dimensional functional learning tasks.

In this **(Taher, Mohammed Yasin Jisan and Rahman, 2019)**, it was found to be over the SVM approach in classify network traffic, together with ANN ML and wrapper selection function, using the NSL KDD data. Analysis results showed that the model developed utilizing the choices of ANN and wrapper feature selection exceeded all other methods with a 94.02 % detection rate.

In this research **(Dong, Wang and He, 2019)**, the incursion approach has been developed to integrate natural language processing, huge data, and extensive teaching methodologies, a DL driven real-time network. The data collection utilized for this research comprises of 5 million entries in a network intrusion data set of KDD99. The training data sub-set and test set also includes 10 percent training. This research has been randomly chosen from a training set of 137,200 data as a training set, dynamically obtained from a subset of the tested strip 2000. Research tells that the AE-AlexNet model has a high attack recognition accuracy, with a total detection rate of 94.32 % context.

This study **(Zhong, Yu and Ai, 2020)** describes the Big Data Hierarchical DL System (BDHDLS) to enhance the efficiency of machine-based IDS. BDHDLS utilizes behavioral and structural functions to examine all network traffic data and trends included in the packet. Each BDHDLS profound learning method relies on studying the particular propagation of data inside one cluster. In comparison with prior simple classification model techniques, the classification rate of disruptive attacks is improved by this methodology. The BDHDLS model creation time decreases considerably when several computers 6 are utilized because of parallel training and Big Data techniques. The combined 5 x 2 CV F test is done during the training dataset for the independent analysis carried out in the test set to assess the considerable improvement to the intrusion detection efficiency of BDHDLS compared with the remaining four models in the ISCX12 data set. The p-value obtained by the 5 x 2 CV F test defines the important level to which the zero assumption may be rejected of algorithms with the same error rate. In contrast to CNN-RNN, the BDHDLS enhances TPR by about 2 percentage points. DARPA1998 dataset does not build the RNN-CNN since the number of packets is relatively tiny for each sample.

The six ML IDS methods presented in this article **(Karatas, Demir and Sahingoz, 2020)** are the K-Nearest Neighbor (KNN), Gradient Boosting, Random Forest, DT, Adaboost and Linear Discriminant Analysis Algorithms. A data sampling strategy was employed to minimize the imbalanced ratio by increasing the minority groups' data size. The empirical findings indicated that in comparison with contemporary literature, the model constructed has a very excellent degree of accuracy. The introduction of a sampled data set resulted in an improvement of 4.01% to 30.59% in average models’ precision. The revised CSE-CIC-

IDS2018 dataset is used to provide a more realistic IDS instead of older and widely used datasets. Furthermore, the facts chosen are incoherent. Thus, a hybrid data collecting approach known as Synthetic Minority Oversampling technique minimizes the coefficient of anomaly to enhance device performance depending on assault methods and prevent missed infringements and false alerts. The data of small classes are 8 and the amounts are raised using this technique to the average data size. According on observational data, the proposed approach enhances the identification rate for assaults that are rarely identified.

The proposed arrangement **(Bharati and Tamane, 2020)** would help to better comprehend the numerous approaches used by IDS work. A DL test method and a mechanical learning system is recommended by this report in ninth form, with overactive boundary enhancement using a handy IDS-digital dataset (CIC-CSE-IDS-2018) identified in higher performance assaults (PCAP) and called sources providing more than 80 features (CSVs). For no parameter set for the model, the unknown layer-sizes are 100 neurons in particular layer and the defaulting alpha is 0.0001. The default in configuration is excessive fit. MLP test accuracy is 95%.

In this article the malware analysis in unstable network transport ML and DL were studied **(Liu *et al.*, 2021)**. In order to deal with the problem of inviable data, a unique DSSTE approach is suggested. Originally, with the Edited Nearest Neighbor (ENN) approach, split the unstable training data in a hard and easy set. The approach avoids inequity in the first training sets and provides the minority group that wishes to examine focused data replenishment. It enables the algorithm to learn the changes in the learning phase correctly and to enhance the rate of accuracy. The CIC-IDS-2018 dataset shows DSSTE+AlexNet well. In some assaults, the rate of detection reaches over 100% and the identification of brute force and infiltration attempts also improves. The research technique has been tried on NSL-KDD, as well as CSE-CIC-IDS2018, as a traditional intrusive dataset.

In this research analysis, some of the techniques used in earlier trials are included. The following section of this article shows this in more depth. This project is more fascinated in the techniques utilized and in the strong topologies of upgraded data sets in this study as opposed to most of this prior research. Also, in this project the most updated dataset of NIDS is used which outperforms the existing approaches that are mentioned in the literature review section. Most effective and efficient models of both, Machine learning and Deep learning techniques are used in this project which is a unique approach than most of the research papers mentioned. Also, in this, the users can choose the models to run for their system improvement depending on their requirement whether they need a time effective model or a model with best accuracy. Besides all of these, running a new model which is AutoEncoder was also attempted but was not completed within the given time because of certain difficulties of the data processing and its standardization but the attempt of using it was done, keeping in mind that it is an unsupervised learning model which means it does not need any supervision for generating the results and could have been used for a lot of improvement in the intrusion detection system in future. But, since the data that was used in this project was predefined and trained accordingly, some difficulties occurred for the AutoEncoder model and so it could not be implemented but it is still being examined and in future it may be implemented in this project and give us a very good result, if it is implemented with the real time data generated. Furthermore, several techniques to DL and ML are implemented. The approach is evaluated using accuracy, recall, false alarm rate, false negative rate, precision, and f-measures.

Table 1: A comparison of related literature review's results and methods

Sr. No.	Year	Author	Dataset	Algorithm used	Accuracy
1	2016	(Javaid *et al.*, 2016)	NSL-KDD-99	KNN, SVM, DT, RF, ANN, Naïve Bayes network	72.14%
2	2016	(Dong and Wang, 2016)	NSL-KDD	LSTM, SMR	75.23%
3	2019	(Li et al., 2019)	KDD99	SVM, DroidRisk, SVM-RFE	94.15%
4	2019	(Taher, Mohammed Yasin Jisan and Rahman, 2019)	NSL-KDD	SVM, ANN	94.02%
5	2019	(Dong, Wang and He, 2019)	KDD99	AE-AlexNet	94.32%
6	2020	(Karatas, Demir and Sahingoz, 2020)	CSE-CIC-IDS2018	KNN, ADABOOST, RF, DT, GB, LDA	Increased by 30.59%
7	2020	(Bharati and Tamane, 2020	CSE-CIC-IDS2018	MLP	95%
8	2021	(Liu et al., 2021).	NSL-KDD, CSE-CIC-IDS2018	DSSTE, DSSTE+AlexNet	100%

3 Research Methodology

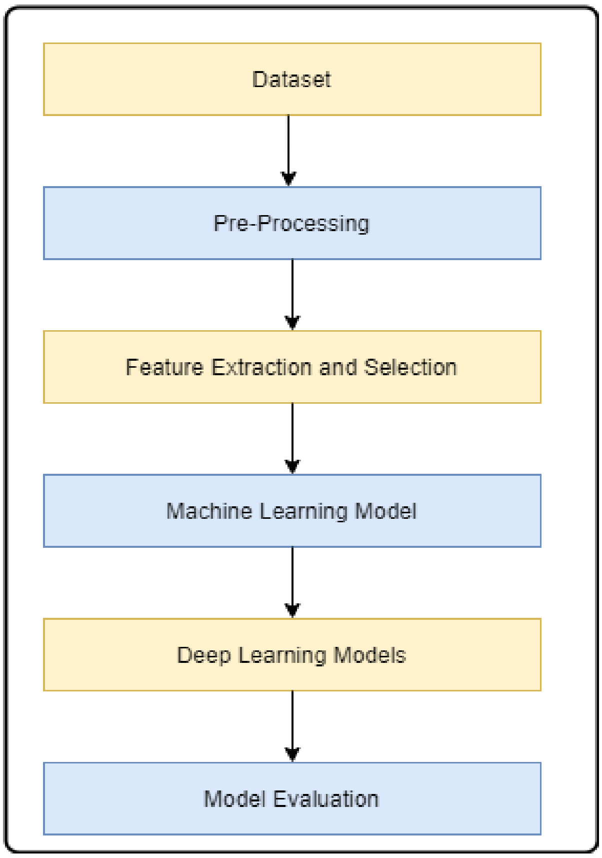


Fig 1: Research Structure

The dataset was collected from the web initially. Then standard ML techniques were followed which included the data pre-processing, feature extraction and feature selection. Then the ML models are applied on the newly generated dataset. Next the DL models are performed and then the evaluation of all the, model was carried out. All these steps are carried out using python language in the Google Colab tool.

3.1 Importing the Dataset:

This dataset was initially produced for the analysis of DDoS data by the University of New Brunswick. The Label column is the most essential part of the data when constructing ML notebooks, as it indicates whether the packets that have been delivered are or are not malicious. In the dataset there are eighty columns, each of which represents an IDS logging system entry in place by the University of New Brunswick. Since its system categorizes traffic forward and behind, columns are available for both. All the variables in the dataset are numerical accept the Label variable which is categorical. A network connection is a sequence of packets that begin and terminate at a certain period during which the data travels from the source IP to the destination IP address where every connection is either labelled as benign or as malicious with just one particular form of assault in this dataset.

3.2 Pre-Processing:

Initially, since the data set was very large, the pre-processing of the data was implemented. In which, the data was first converted into a pickle, to store the large dataset into lesser memory. Then the int variables were converted into float, to equalise the complete dataset, as it was an important step to be performed[1]. Next, the Flow Pkts/s, Timestamp and Flow Byts/s variables was removed as it had a lot of NaN and infinite values as well as outliers and it could further create a disturbance in the results and accuracy of the models. Further, the two categories in the variable 'Label' which are 'Benign' and 'Bot' were then replaced with 0 and 1 respectively, in the dataset. This is how the data pre-processing was carried out followed by the feature extraction and feature selection which is explained in section 5.

4 Design Specification

Access to information in order to breach its credibility, security, or functioning within a system or network architecture is considered to as intrusion. A technology to analyze the traffic of a user or connection for any criminal behavior that violates the security of a network and weakens its confidentiality, integrity and availability is therefore an IDS or NIDS, to be more precise. When suspicious behavior is identified, the NIDS can convey the user or system administrator, alerts. The image below demonstrates the static NIDS implementation using port mirror technology connected to a switch port. Its objective is to transmit all traffic entering and departing in the form of packets to NIDS to monitor traffic and identify intrusions. It can be situated between the firewall and the network switch to enable all traffic to pass over NIDS. And if any malicious packet is detected, then give an alert. The following architecture underlies the implementation of this project.

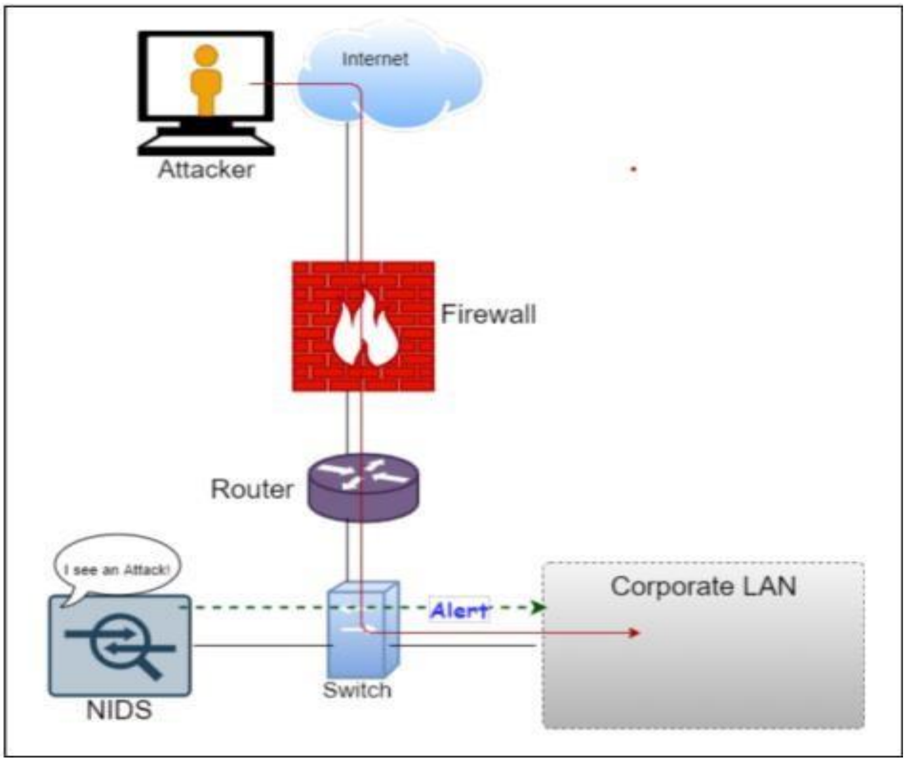


Fig 2: Architecture that underlie the implementation

[1] https://datatofish.com/integer-to-float-datafra

5 Implementation

First the data is standardized to get a good shape of the distribution of the data. And further, after the feature extraction and selection, the models are executed to get the results.

5.1 Feature Extraction and Selection:

After the pre-processing is done, the Random Forest classification is performed here, for feature selection. RandomForestClassifier function is imported from the Scikit library. Further, a new data frame is created with only the most important variables selected by the Random Forest feature extraction method. As the final data frame is now retrieved, the feature and the target variables are further derived. Since, the variable ‘Label’ is the output variable, it is set as the target and rest of the variables are set as the features. Further, the data is split in the two parts, training data and the testing data. The size of the training data is set as 80% of the data randomly and the remaining 20% of it as the test data.

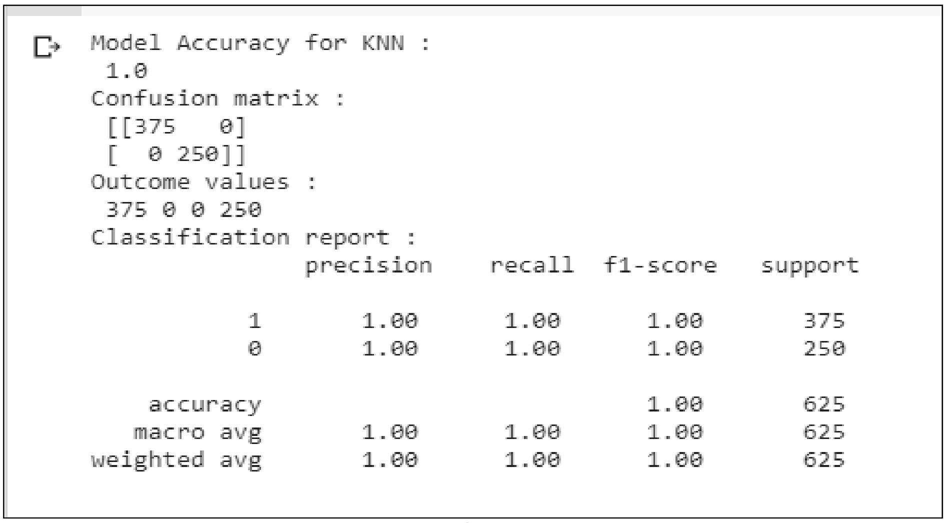
5.2 Models and model evaluation

The ML models used in this project are KNN, DT and Artificial Neural Network (ANN) and the DL models used are Multilayer Perceptron (MLP), which is a subset of DNN and Convolutional Neural Network (CNN).

Machine Learning models:

K-Nearest Neighbour (KNN):

The approach of classification is quite simple in K-Nearest Neighbor (KNN). By looking for nearest occurrences in the training data set, KNN classifies brand - new observations in their respective categories. In closest circumstances, the distance metric called the Euclidean distance is defined. As in every other supervised learning approach, KNN consists of training and assessment phases. K is the number of closest neighbors in KNN. The K value is usually an odd integer when the class number is 2. If K=1, the technique is simply referred to in the nearest algorithm. The result revealed a 99.92% model accuracy with the Cross Validation Average of 0.99 for K-Nearest Neighbor (KNN). The findings were likewise high regarding anomaly precision and normal network recall.

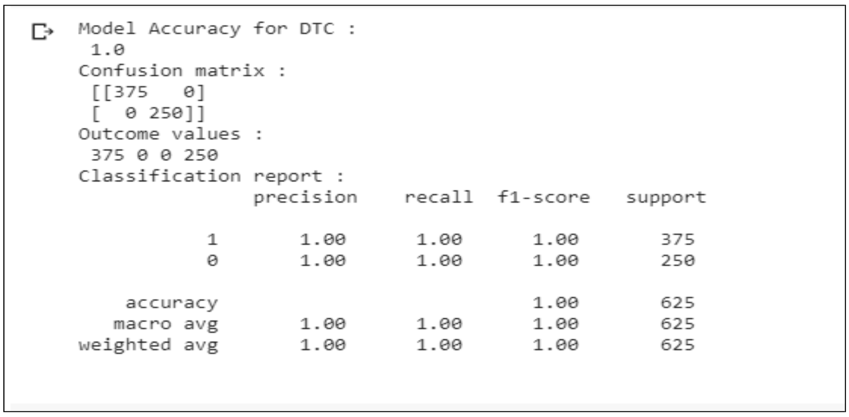


```
Model Accuracy for KNN :
 1.0
Confusion matrix :
 [[375   0]
 [  0 250]]
Outcome values :
 375 0 0 250
Classification report :
               precision    recall  f1-score   support

           1       1.00      1.00      1.00       375
           0       1.00      1.00      1.00       250

    accuracy                           1.00       625
   macro avg       1.00      1.00      1.00       625
weighted avg       1.00      1.00      1.00       625
```

Fig 3: Result for KNN model

Decision Tree:

Decision Tree (DT) is the supervised technique of ML that is used to classify and predict the data set. It is one approach of showing an algorithm which simply includes conditions of control. The model demonstrated 100% accuracy based on results generated using the DT Model Evaluation. Also, the default hyper parameters such as criterion as ‘gini’ and random_state as None, are passed to this model to get the accuracy. It demonstrates that the dataset we have altered is inaccurately labelled for splitting from the dataset.[2]

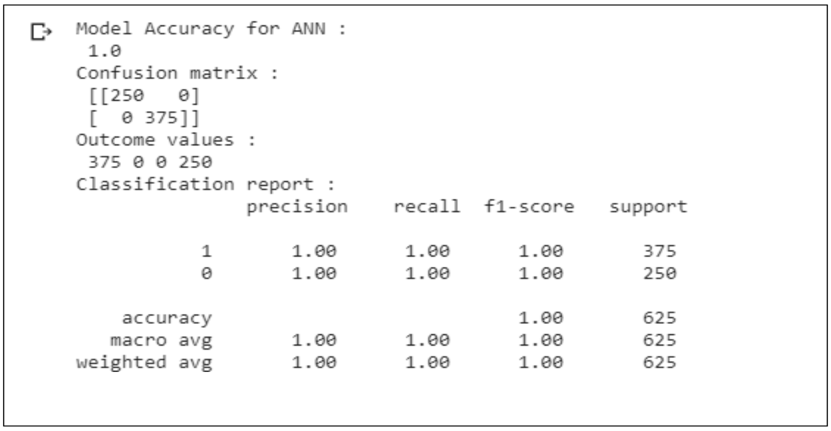


```
Model Accuracy for DTC :
 1.0
Confusion matrix :
 [[375   0]
 [  0 250]]
Outcome values :
 375 0 0 250
Classification report :
               precision    recall  f1-score   support

           1       1.00      1.00      1.00       375
           0       1.00      1.00      1.00       250

    accuracy                           1.00       625
   macro avg       1.00      1.00      1.00       625
weighted avg       1.00      1.00      1.00       625
```

Fig 4: Result for DTC model

Artificial Neural Network (ANN):

It also operates how the data is collected by a human brain. ANN contains a wide range of linked processing devices that deal with information processing. ANN is usually structured in several layers, the input layer, the hidden layer and the output layer. The layers consist of multiple of linked nodes with an 'activation function.' The back-spread training of the network comprises three stages: feed-forward of the input-training pattern, calculating and backpropogating the related error and adjusting weight to create a power vector for a certain input vector depending on a particular state of network weights. Here the use of the keras library, which is a neural network library in python, is done. The ANN model accuracy is 98.68% based on the findings received from model evaluation. With an computational time of 16 seconds approximately. (Chudasma, no date)[3]



```
Model Accuracy for ANN :
 1.0
Confusion matrix :
 [[250   0]
 [  0 375]]
Outcome values :
 375 0 0 250
Classification report :
               precision    recall  f1-score   support

           1       1.00      1.00      1.00       375
           0       1.00      1.00      1.00       250

    accuracy                           1.00       625
   macro avg       1.00      1.00      1.00       625
weighted avg       1.00      1.00      1.00       625
```

Fig 5: Results for ANN model

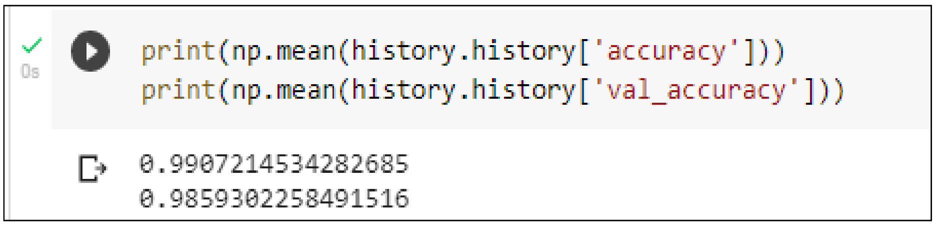
[2] https://datascience.foundation/sciencewhitepaper/understanding-decision-trees-with-python

[3] https://stackoverflow.com/questions/68185988/valueerror-input-0-of-layer-sequential-is-incompatible-with-the-layer-expected

Deep learning models:

Deep Neural Network:

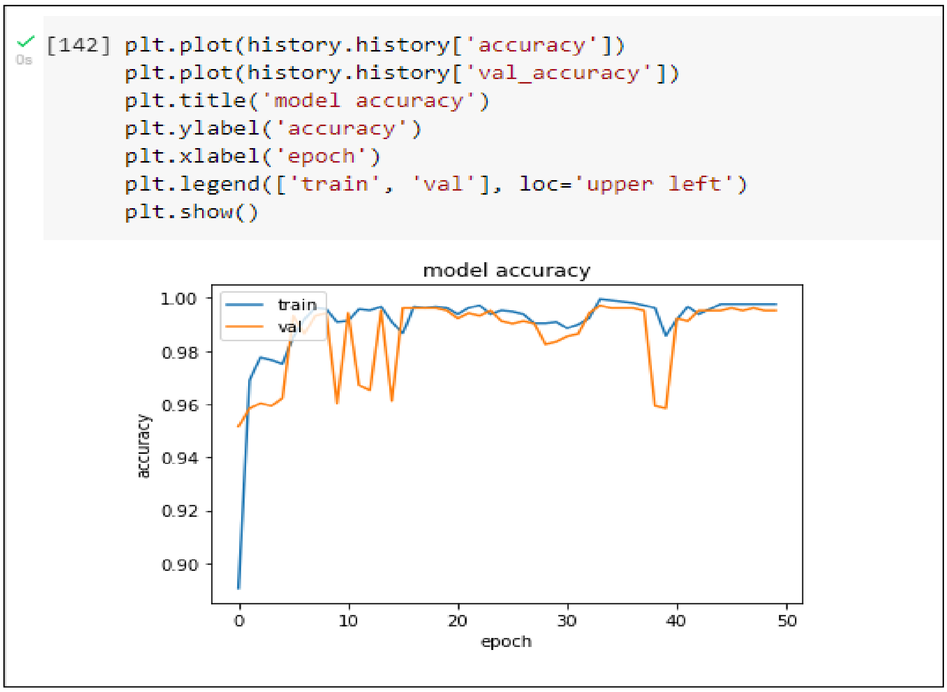
In this step, the MLP, which is a subset of DNN is build. This classifier is build using the keras library backend of tensorflow. This model gives a 100% of accuracy with an execution time of 13 seconds approximately. The optimizer Adadelta (Adam) is a stronger Adagrad extension which modifies training rates based on a moving gradient update window, rather than accruing all the previous gradients. This manner, even if numerous upgrades have been done, Adadelta continues to learn.[4]



```
print(np.mean(history.history['accuracy']))
print(np.mean(history.history['val_accuracy']))
```

```
0.9907214534282685
0.9859302258491516
```

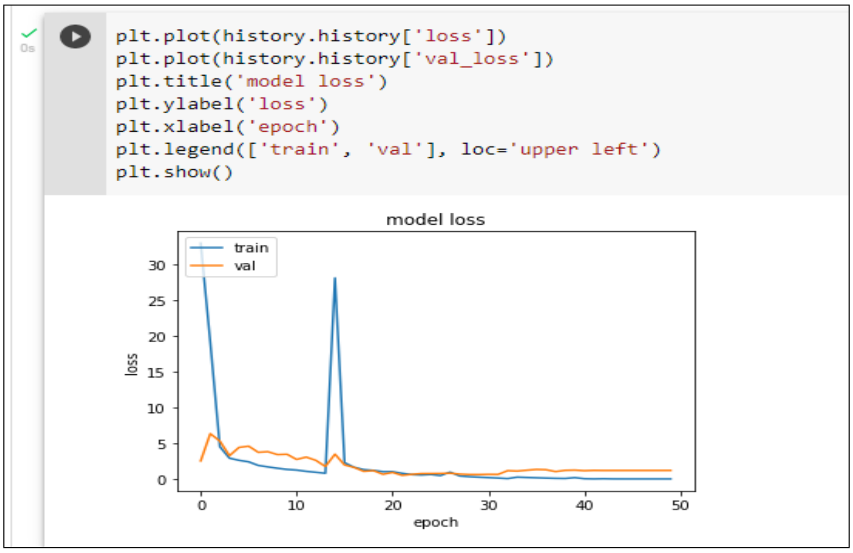
Fig 6: Result for accuracy and validation accuracy for MLP



```
[142] plt.plot(history.history['accuracy'])
      plt.plot(history.history['val_accuracy'])
      plt.title('model accuracy')
      plt.ylabel('accuracy')
      plt.xlabel('epoch')
      plt.legend(['train', 'val'], loc='upper left')
      plt.show()
```

Fig 7: MLP model accuracy of actual and predicted result graph

[4] https://elearning.dbs.ie/pluginfile.php/1301095/mod_resource/content/1/Deep%20Learning%20Tutorial.html

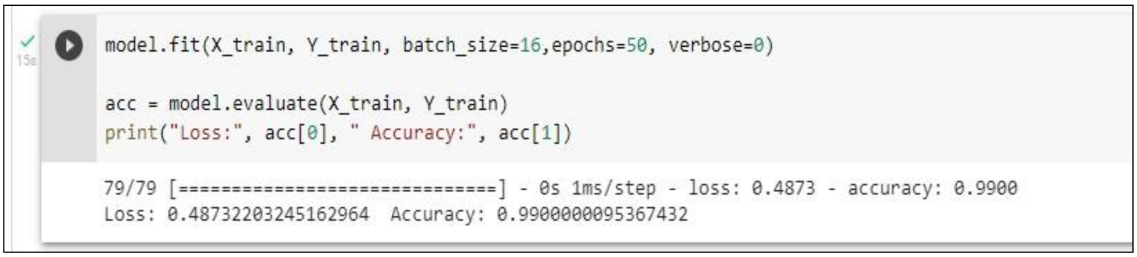


```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

Fig 8: MLP model loss in actual and predicted result graph

Convolutional Neural Network:
In CNN, the convolutional layer understands local data patterns. The characteristics of input data are extracted to supply the output. A CNN (CNN or ConvNet) in DL is the most widely used ANN to evaluate visual imaging. CNNs are MLP's regularized versions. MLP typically indicate that every neuron inside a layer is linked to all the neurons in the next layer. This refers to a completely connected network. This model of DL gives a 98% of accuracy with the execution time of 11 seconds approximately.

Note: Since the size of the training data is set as 80% of the entire data randomly. Each time the code is run, the sample accuracy and the execution time changes. But most of the time the accuracy is near to 99% in an average for the models.[5]



```
model.fit(X_train, Y_train, batch_size=16,epochs=50, verbose=0)

acc = model.evaluate(X_train, Y_train)
print("Loss:", acc[0], " Accuracy:", acc[1])
```

```
79/79 [==============================] - 0s 1ms/step - loss: 0.4873 - accuracy: 0.9900
Loss: 0.48732203245162964  Accuracy: 0.9900000095367432
```

Fig 9: Accuracy of actual and predicted result for CNN model

Tools used for research implementation:

For a long time, Python is now the most important language for developers of ML and artificial intelligence. Python offers a broad variety of flexibility and functions for developers to increase not only their usability but also their development consistency. Accordingly, Python is utilized to implement this project as well. It has employed library package like Keras, Scikit-Learn, TensorFlow, etc. Python is a highly utilized language which uses mathematical formulas and maps to analyze data.

[5] https://www.datatechnotes.com/2020/02/classification-example-with-keras-cnn.html

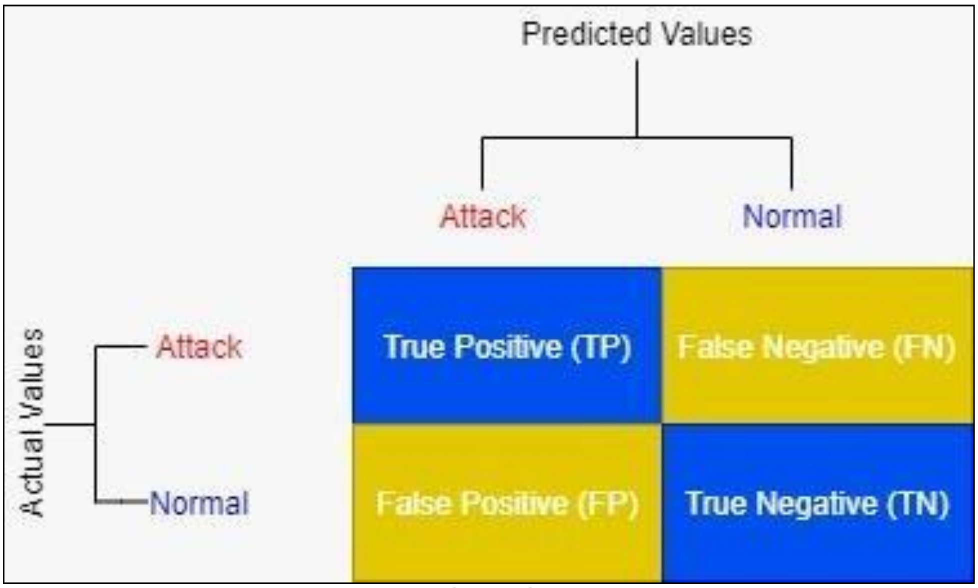
6 Evaluation

The evaluation of the models is done on the following metrices:

Performance Metrics:
In this section, the most used assessment metrics for this research is covered to calculate the effectiveness of ML and DL techniques. Many assessment measures are generated from the different properties of the two-dimensional matrix, which includes data on the measured and expected classes, contained in the Confusion Matrix:

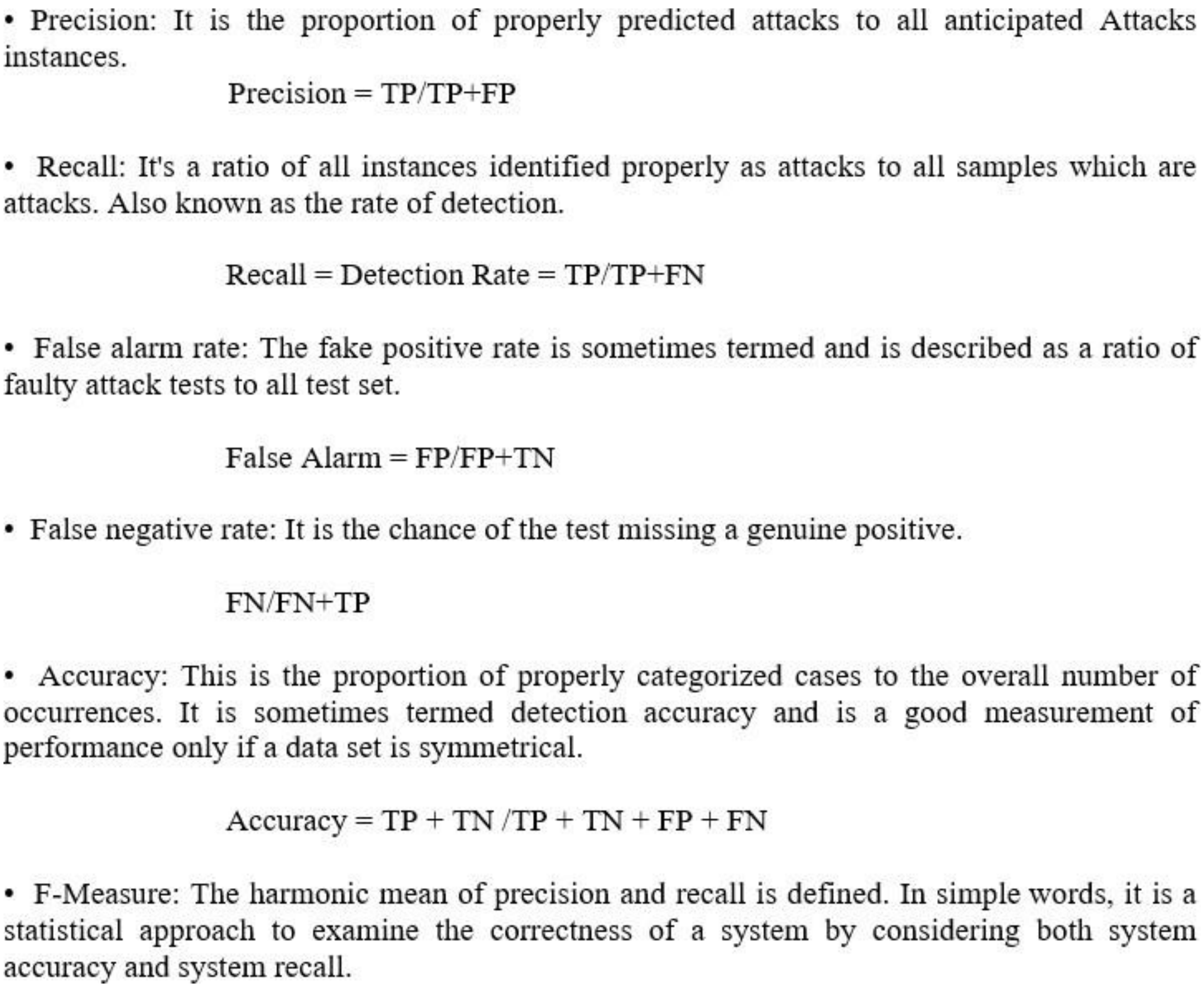
- True Positive (TP): The data instance is anticipated as a classifier attack successfully.
- False Negative (FN): The instance of data as normal instances is being mispredicted.
- False Positive (FP): The data instance is categorized incorrectly as an assault.
- True Negative (TN): The case was categorized appropriately as normal.

The correct judgments are expressed by the diagonal elements of the confusion matrix, whereas non-diagonal components indicate the wrong predictions. The following table shows some properties of the confusion matrix. In addition, the various evaluation metrics employed throughout this investigation are as follows:



		Predicted Values	
		Attack	Normal
Actual Values	Attack	True Positive (TP)	False Negative (FN)
	Normal	False Positive (FP)	True Negative (TN)

Fig 10: Properties of Confusion Matrix



- Precision: It is the proportion of properly predicted attacks to all anticipated Attacks instances.

Precision = TP/TP+FP

- Recall: It's a ratio of all instances identified properly as attacks to all samples which are attacks. Also known as the rate of detection.

Recall = Detection Rate = TP/TP+FN

- False alarm rate: The fake positive rate is sometimes termed and is described as a ratio of faulty attack tests to all test set.

False Alarm = FP/FP+TN

- False negative rate: It is the chance of the test missing a genuine positive.

FN/FN+TP

- Accuracy: This is the proportion of properly categorized cases to the overall number of occurrences. It is sometimes termed detection accuracy and is a good measurement of performance only if a data set is symmetrical.

Accuracy = TP + TN /TP + TN + FP + FN

- F-Measure: The harmonic mean of precision and recall is defined. In simple words, it is a statistical approach to examine the correctness of a system by considering both system accuracy and system recall.

F-Measure = 2 (Precision × Recall /Precision + Recall)

Table 2: The results of all the models implemented in this research.

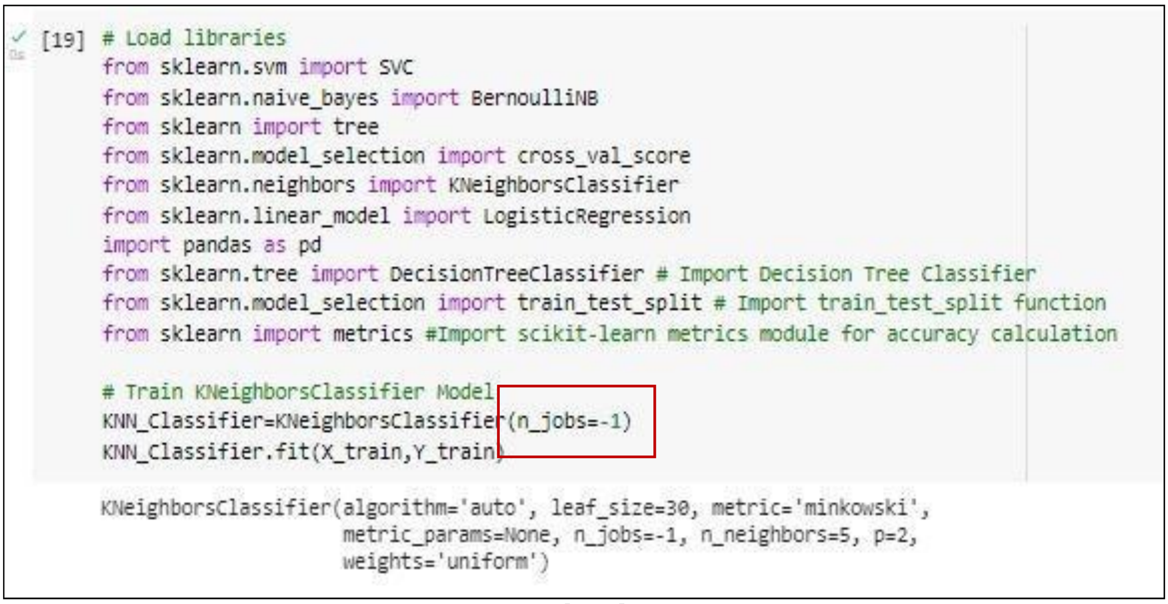
Models	Accuracy	Computational Time
KNN	99%~	Cannot be measured (in nano seconds)
DT	99%~	Cannot be measured (in nano seconds)
ANN	99%~	16 sec
MLP (a type of DNN)	100%~	13 sec
CNN	99%~	11 sec

In this proposed research, the accuracy of all the models is better than the accuracy of the existing approaches as observed from Table 1. Also, various models are used and being compared for better results. As we can see here, the accuracy given by KNN model is 99% approx., where the value of k is 3 and the number of false negatives is 0-1. Similarly, the accuracy given by the DT classifier is 99- 100% approximately by making the use of default hyper parameters and the number of false negatives is 0-1 approximately. In ANN, the model is given 50 samples to get trained with. Here, the accuracy between the actual and the predicted results is 99% approximately with the number of false negatives id 0-1 approximately. DL models are used when the data set is very large and complex. Here, the MLP model, which is a type of DNN is used. Also, CNN model is used. Though, CNN is majorly used for image processing, it is used just to make a unique comparison between different neural networks and to check its way and behaviour of execution. The results of the predicted accuracy 98% and the validated accuracy is 99% approximately in MLP. It is also visualized in the plots executed., where we can see the training and the validated accuracy and the trained and the validated loss of the model. In CNN the accuracy given by the model is 99% approximately.

6.1 Case Studies

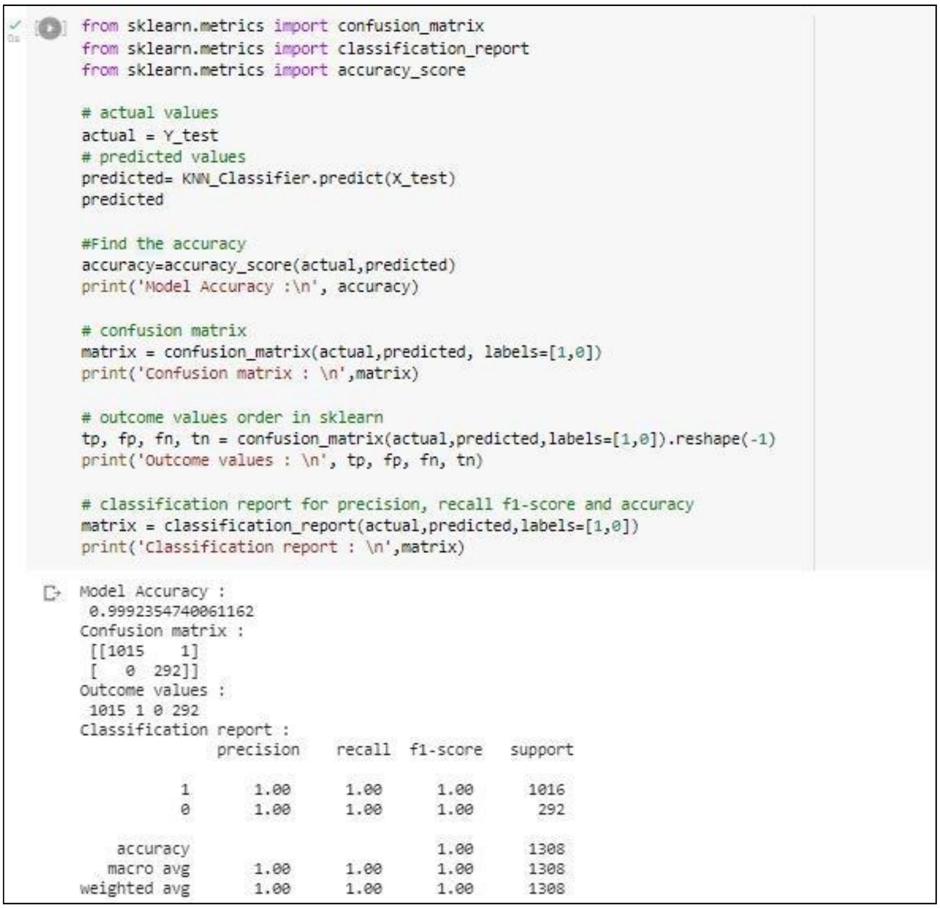
Feature Extraction:

In this scenario, the features were manually selected in the beginning, but some how the accuracy was occurring somewhere around 97-99% for most of the models as highlighted in the figure 11. Then, by using the RandomFeatureElimination (RFE) method, only the top 15 features were selected which help in improving the accuracy.



```
[19] # Load libraries
     from sklearn.svm import SVC
     from sklearn.naive_bayes import BernoulliNB
     from sklearn import tree
     from sklearn.model_selection import cross_val_score
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.linear_model import LogisticRegression
     import pandas as pd
     from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
     from sklearn.model_selection import train_test_split # Import train_test_split function
     from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

     # Train KNeighborsClassifier Model
     KNN_Classifier=KNeighborsClassifier(n_jobs=-1)
     KNN_Classifier.fit(X_train,Y_train)

     KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                          metric_params=None, n_jobs=-1, n_neighbors=5, p=2,
                          weights='uniform')
```

Fig 11: k value as 1



```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

# actual values
actual = Y_test
# predicted values
predicted= KNN_Classifier.predict(X_test)
predicted

#Find the accuracy
accuracy=accuracy_score(actual,predicted)
print('Model Accuracy :\n', accuracy)

# confusion matrix
matrix = confusion_matrix(actual,predicted, labels=[1,0])
print('Confusion matrix : \n',matrix)

# outcome values order in sklearn
tp, fp, fn, tn = confusion_matrix(actual,predicted,labels=[1,0]).reshape(-1)
print('Outcome values : \n', tp, fp, fn, tn)

# classification report for precision, recall f1-score and accuracy
matrix = classification_report(actual,predicted,labels=[1,0])
print('Classification report : \n',matrix)
```

```
Model Accuracy :
 0.9992354740061162
Confusion matrix :
 [[1015    1]
 [   0  292]]
Outcome values :
 1015 1 0 292
Classification report :
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      1016
           0       1.00      1.00      1.00       292

    accuracy                           1.00      1308
   macro avg       1.00      1.00      1.00      1308
weighted avg       1.00      1.00      1.00      1308
```

Fig 12: Accuracy when k value is 1

Selecting the k value:

The k-value is usually an odd integer when the class number is 2. If k=1, the technique is simply referred to in the nearest algorithm. So, the value of k was considered as 1, 3, 5 and 7. But then it was concluded that the k=3 was giving the best accuracy for all the models.

6.2 Discussion

In this proposed research, the accuracy of all the models is better than the accuracy of the existing approaches as observed from Table 1. Also, various models are used and being compared for better results. There is a lot more required in this field and betterment in the (NIDS). ML and DL techniques are utilized in most of the fields nowadays. Researchers are making a huge impact in the cyber security and networking world. NIDS with ML and DL is just the beginning to make a small improvement for making a big improvement in this sector. This research work can be used for monitoring a real time network traffic and detecting the packets travelling in and out of the network system. This can be a limitation for this research but can be a good future scope.

7 Conclusion

Since, the main motive of this paper to make improvement in the NIDS by using ML and DL models and to check which models are more helpful in this task, the focus of this implementation is on the false negatives, while evaluating models based on precision, recall, accuracy, and F- measures. Every model is giving a different accuracy and in different computational time. As mentioned above, the training size is set as 80% randomly of the data frame created. Based on the confusion matrix, it can be examined that which model is giving the better accuracy and less numbers of false negative. Since, a greater number of false negatives can harm our system. For example, if a packet entering through a network in the NIDS, is malicious in real, and if the predicted result for the receiving packet is benign (non-malicious), from the model, then it can be said that the chances of the system getting exploited, increases.

As it can be observed from this research that some of the ML Models are most of the time giving more accurate results than the DL models, while neural networks, the DL models are giving more accuracy than ANN in less computational time. So, it can be concluded that the DL models can give better accuracy than ANN, when it comes to neural networks, but the KNN and DT algorithms are the best fit models for this dataset (Results may vary by different dataset). Though in case of large and complex datasets. DL algorithms are much preferable for better accuracy and validation. The limitations of this research are as follows; Use of only one dataset is done and executed for all the models and Visualization of only one model is shown.

Future Scope:
The artefact of this project can be displayed in a user interface. Also, multiple datasets can be used on these models. Various other complex models may be executed in this research. Also, the use of signature-based NIDS and Anomaly based NIDS may be implemented and worked on in future.

References

Aminanto, M. E. and Kim, K. (no date) ‘Deep Learning in Intrusion Detection System: An Overview’, p. 12.

Bharati, M. P. and Tamane, S. (2020) ‘NIDS-Network Intrusion Detection System Based on Deep and ML Frameworks with CICIDS2018 using Cloud Computing’, in *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC). 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, Aurangabad, India: IEEE, pp. 27–30. doi: 10.1109/ICSIDEMPC49020.2020.9299584.

Çavuşoğlu, Ü. (2019) ‘A new hybrid approach for intrusion detection using ML methods’, *Applied Intelligence*, 49(7), pp. 2735–2761. doi: 10.1007/s10489-018-01408-x.
Chudasma, P. (no date) ‘Network Intrusion Detection System using Classification Techniques in ML’, p. 74.

Dong, B. and Wang, X. (2016) ‘Comparison deep learning method to traditional methods using for network intrusion detection’, in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, Beijing, China: IEEE, pp. 581–585. doi: 10.1109/ICCSN.2016.7586590.

Dong, Y., Wang, R. and He, J. (2019) ‘Real-Time Network Intrusion Detection System Based on Deep Learning’, in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China: IEEE, pp. 1–4. doi: 10.1109/ICSESS47205.2019.9040718.

García-Teodoro, P. *et al.* (2009) ‘Anomaly-based network intrusion detection: Techniques, systems and challenges’, *Computers & Security*, 28(1–2), pp. 18–28. doi: 10.1016/j.cose.2008.08.003.

Javaid, A. *et al.* (2016) ‘A Deep Learning Approach for Network Intrusion Detection System’, in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, New York City, United States: ACM. doi: 10.4108/eai.3-12-2015.2262516.

Karatas, G., Demir, O. and Koray Sahingoz, O. (2018) ‘Deep Learning in Intrusion Detection Systems’, in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT). 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, ANKARA, Turkey: IEEE, pp. 113–116. doi: 10.1109/IBIGDELFT.2018.8625278.

Karatas, G., Demir, O. and Sahingoz, O. K. (2020) ‘Increasing the Performance of ML-Based IDSs on an Imbalanced and Up-to-Date Dataset’, *IEEE Access*, 8, pp. 32150–32162. doi: 10.1109/ACCESS.2020.2973219.

Kim, K. and Aminanto, M. E. (2017) ‘Deep learning in intrusion detection perspective: Overview and further challenges’, in *2017 International Workshop on Big Data and Information Security (IWBIS). 2017 International Workshop on Big Data and Information Security (IWBIS)*, Jakarta: IEEE, pp. 5–10. doi: 10.1109/IWBIS.2017.8275095.

Lee, B. and Amaresh, S. (2018) ‘Comparative Study of Deep Learning Models for Network Intrusion Detection’, 1(1), p. 14.

Li, J. *et al.* (2019) ‘ML Algorithms for Network Intrusion Detection’, in Sikos, L. F. (ed.) *AI in Cybersecurity*. Cham: Springer International Publishing (Intelligent Systems Reference Library), pp. 151–179. doi: 10.1007/978-3-319-98842-9_6.

Liu, Lan *et al.* (2021) ‘Intrusion Detection of Imbalanced Network Traffic Based on ML and Deep Learning’, *IEEE Access*, 9, pp. 7550–7563. doi: 10.1109/ACCESS.2020.3048198.

Nevlud, P. *et al.* (2013) ‘Anomaly-based Network Intrusion Detection Methods’, *ADVANCES IN ELECTRICAL AND ELECTRONIC ENGINEERING*, 11(6), p. 7.

Nguyen Thanh Van, Tran Ngoc Thinh, and Le Thanh Sach (2017) ‘An anomaly-based network intrusion detection system using Deep learning’, in *2017 International Conference on System Science and Engineering (ICSSE). 2017 International Conference on System Science and Engineering (ICSSE)*, Ho Chi Minh City, Vietnam: IEEE, pp. 210–214. doi: 10.1109/ICSSE.2017.8030867.

Peng, W. *et al.* (2019) ‘Network Intrusion Detection Based on Deep Learning’, in *2019 International Conference on Communications, Information System and Computer Engineering (CISCE). 2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, Haikou, China: IEEE, pp. 431–435. doi: 10.1109/CISCE.2019.00102.

Shah, S. N. and Singh, P. (2012) ‘Signature-Based Network Intrusion Detection System Using SNORT And WINPCAP’, *International Journal of Engineering Research*, 1(10), p. 7.

Taher, K. A., Mohammed Yasin Jisan, B. and Rahman, Md. M. (2019) ‘Network Intrusion Detection using Supervised ML Technique with Feature Selection’, in *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST). 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh: IEEE, pp. 643–646. doi: 10.1109/ICREST.2019.8644161.

Zhong, W., Yu, N. and Ai, C. (2020) ‘Applying big data based deep learning system to intrusion detection’, *Big Data Mining and Analytics*, 3(3), pp. 181–195. doi: 10.26599/BDMA.2020.9020003.