National College of Ireland

Cyber Security Intrusion Detection Deep Learning Model for Internet of Things (IoT)

MSc Research Project

MSc in Cyber Security

Abhirup Bhattacharjee

Student ID: x20250029

School of Computing

National College of Ireland

Supervisor: Mr. Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Abhirup Bhattacharjee
Student ID:	X20250029
Programme:	MSc in Cyber Security **Year:** 2021-22
Module:	MSc Research Project
Supervisor:	Mr. Michael Prior
Submission Due Date:	15.08.2022
Project Title:	Cyber Security Intrusion Detection Deep Learning Model for Internet of Things (IoT)
Word Count:	5496 **Page Count** 22

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: Abhirup Bhattacharjee

Date: 14.08.2022

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):	

Cyber Security Intrusion Detection Deep Learning Model for Internet of Things (IoT)

Abhirup Bhattacharjee

x20250029

Abstract

The current paper focuses on a detailed investigation of deploying deep learning techniques for IoT system intrusion detection utilizing specific neural network models. The current research represents a future to be conducted over NSL-KDD dataset archived by UNSW lab. This paper's primary objective is to develop a Convolutional Neural Network-based Intrusion Detection System (IDS) to improve internet security. The recommended IDS architecture classifies all network packet traffic into types that are benign or malicious in order to identify network intrusions. For the suggested approach, CNN, DNN, Logistic Regression, Adaboost, and Random Forest four important experimental DL models, are taken into consideration. Performing a comparison analysis using variables like accuracy, precision, recall, F1, and run time is a key component of the study.

1 Introduction

As computer networks are so commonly used in different fields, network security has increased in importance. Networks that are subject to a variety of threats handle a variety of sensitive user data. To identify and execute attacks, a range of tools including firewalls, antimalware, antivirus software, etc. are used. These attacks vary in type and sophistication, making it impossible for traditional techniques to detect them, which results in data breaches. These malicious assaults provide security problems, making an effective and adaptable IDS crucial. Network security has gained significance as a result of how frequently computer networks are utilized in various industries. Multiple types of sensitive user data are handled via networks that are vulnerable to several attacks. A variety of tools, such as firewalls, antimalware, antivirus software, etc., are used to detect and carry out attacks. Data breaches occur as a result of the variety and sophistication of these attacks, which make it impossible for traditional procedures to identify them. Security issues are caused by these malicious attacks, making an efficient and flexible IDS essential.

1.1 Background and Motivation

This project's objective is to develop a basic understanding of latest innovations and improvements in ML and DL approaches for NIDS. The primary goal is to give new professionals who are keen to investigate this expanding development current updated information about the most relevant ML and DL-based NIDS. Several academic research articles have explained how to construct the IDS. This research provides to the most recent research and innovative advances in AI-based NIDS design. The purpose of this research is to provide academics with more recent data on a certain area on an AI-based NIDS so they can search for novel patterns and chances to investigate the area.

1.1.1 Internet of Things

The Internet of Things, or IoT, is a technology made up of several physical devices or object groups that contain processing units, software, and sensors that are connected via communication networks or the Internet. Such a system can be interconnected to a communication network while yet remaining individually addressable; it is not necessary for it to be fully connected to the public network or the Internet. The system is thus a misnomer system. Through numerous sensors like optical sensors, temperature sensors, accelerometers, and actuators like physical objects, switches, and electrical circuits, an IoT-based ecosystem interacts with the environment in which it exists. They are consequently a kind and flexible technology.

1.1.2 Machine learning in intrusion detection

IoT security solutions have significant difficulties in anticipating and identifying intrusion threats. Systems are being trained to recognize these intrusions using machine learning techniques and technology. These methods offer models that, after being trained on labelled or unlabelled data intrusion datasets, can predict and raise alert flags of such intrusions. It is difficult to reduce computing expenses and overheads without sacrificing accuracy, though. Many studies are focusing on using supervised and unsupervised machine learning (ML) approaches to improve the performance of IDS. Naive Bayes, Convolution Neural Network (CNN), Random Forest, Logistic Regression, Dense Neural Network (DNN), and ML techniques are deployed. IDS generated with machine learning have a high accuracy with little input data. However, if a significant dataset is chosen, it takes a lot of time and has a considerable latency. These traditional algorithms have several drawbacks. Due to the increased feature count, ML methods cannot successfully classify several classes. Overfitting and high bias caused by duplicate or irrelevant characteristics are two challenges.

1.2 Research Questions

The main objective of the research project is to achieve different machine learning and deep learning models to answer below research questions. The key research questions of the study would be as follows:

- How can network intrusion detection systems be improved using ML and DL techniques?
- How to detect intrusions into an IoT system with overhead and sufficient accuracy with the proposed machine learning model?

1.3 Research Objective

A detailed comparison of this study with previous review articles is given in Table 1 from Section 2. The next part is structured as follows: Section 3 describes the technique and the beginning procedure. Section 4 provides a description of the design criteria, architecture and also overview of the suggested ML and DL techniques. Sections 5 and 6 present the implementation, evaluation of the research. The research article concludes with Section 7 that covers conclusions, limitations.

2 Related Work

2.1 The penetration of IoT in a daily day uses

In order to understand the importance of a sophisticated level of data security for IoT technology, it is essential to understand the importance of IoT in our daily lives. There is a wide range of use cases for IoT in our daily-day lives, and the applications are gradually increasing. As per (Ullah, et al., 2019), one of the critical cases of IoT is intelligent home applications in our regular lives. Though it is a variable concept, widely popular applications are becoming more and more popular. It uses intelligent connected systems (NickolaosKoroniotisNourMoustafa, et al., 2019) for utilities, entertainment and security. Intelligent illuminations systems, smart appliances (kitchen utilities, washing machines, refrigerators and so on), smart controllable electrical appliances and so on - all can be connected as a misnomer system of things and can be controlled through a smart portable device like mobile phones or tablets. Furthermore, smart TV to record shows on-demand, accessing online and OTT platforms, accessing various applications through remote controllers, and smartphones. The literature review should end in a paragraph that summarises the findings from the state of the art, (Serrano, 2019) why the previous solutions are not adequate and justifies the need for your research question. (Al-Garadi, et al., 2019) The content sections of your report should of course be structured into subsections. Note that here there are 2 subsections subsection 2.1 and subsection 2.2. are all applications of IoT. IoT systems are also becoming famous for home security systems like smart fire alarms, surveillance systems, intelligent lock systems, home intrusion detection, etc. As explored by (Liu, et al., 2020) another critical use case of IoT applications is the Smart City context. It is a concept of connected devices beyond mere internet access to all (MengmengGe, et al., 2021). This incorporates a well-connected city targeting an interconnected ecosystem of various vital infrastructural requirements such as traffic management, intelligent power grid, waste management, etc (Baldini & Raimondo Giuliani, 2018). This incorporates a concept of an intelligent infrastructural ecosystem solution for an urban population. Health care and monitoring is another widespread use case for IoT applications proving to be very useful. (Huma, et al., 2018)

2.2 Machine Learning & deep learning

Machine learning (ML) models or algorithms allow a system to get trained from a training dataset (supervised or unsupervised) to predict the nature and hidden data patterns in test and application datasets. Deep learning (DL) is a sophisticated ML technique used to make a system learn from examples and use them as training data, like how human intelligence works. It is achieved through artificial neural networks (ANN), a network of virtual neurons or programming layers (Jhon Alexánder Parra, 2022). Each layer performs a particular machine learning task to pass on the output as an input to the next layer. (Costa, et al., 2018) Thus, a multiplied and multi-dimensional assessment of the dataset is possible to be achieved through this system. Each layer takes a multiplied version of the input variables with random weights. These inputs are added with a certain static bias unique to that particular layer (Smys, et al., 2020).

This input is then passed to the activation function that decides whether to pass it on to a particular neuron node in the next layer or not. The outcome is a well-sophisticated classification of the data. There are several neural network (NN) techniques (Luo, et al., 2021) that are used widely (Farhin, et al., 2021), such as CNN (Convolution NN), FNN (Feed-forward NN), DNN (Deep NN), RNN (Recurrent NN) and so on. Two essential methods are most widely used for intrusion detection in the current data analysis process. Two of the most widely used models for the previous related works are convolution neural network (CNN) (Shen, et al., 2019) and feedforward neural network (FNN). However, (Pang & Ma, 2018). Both of these models come with their strengths and weaknesses, and hence a comparative analysis of two different model in most previous works, the focus has been on developing a single model or comparing different models based on qualitative and literature studies based on the same dataset can help in determining a suitable model among them. (McDermott, et al., 2020)

2.3 Feature Selection

In this study, the performance and precision of three concepts—a regular DNN, a Self-Taught Learning (STL), and a Recurrent Neural Network (RNN)—are compared based on a Long Short-Term Memory (LSTM) (Lee and Amaresh, 2018). Database intrusion data systems utilize knowledge discovery to analyse their findings. These data were utilized for the third International Competition for Information Exploration and Data Mining, which was held in conjunction with KDD Cup 1999. Multinomial logistic regression was used to compare the findings to a typical deep model in order to determine whether DL methods outperformed deep algorithms in this model.

This post (Aminanto & Kim, 2022) provides a DL-based analysis of previous IDSs. Firewalls use DL techniques. The benefits and drawbacks of these DL approaches are then discussed and assessed so that you may learn more about when and why you should use DL. It is agreed that there is still room for improvement in terms of how DL software should be used.

According to this study, IDS is helpful in DL, particularly in lowering dimensionality. This claim is supported by the fact that implementation of DL in IDS is subject to suggestions and potential issues. Finally, DL models can be helpful in future study on the detection of unidentified threats. In this (Taher, et al., 2019), it was discovered that the ANN ML and wrapper selection function outperformed the SVM technique in classifying network traffic using the NSL KDD data. The results of the analysis demonstrated that the model created using the options of ANN and wrapper feature selection outperformed all other techniques with a detection rate of 94.02%. (Zhang, et al., 2018)

By citing a piece of research, this article (Karatas, et al., 2018) offers a brief overview of the IDS technique for DL. The three most often used datasets in this research are KDD Cup99, NSL-KDD, and CSE CIC- IDS2018. The implementation of IDS applications based on profound knowledge, which also compare ML and DL approaches, are also contrasted.

The research mentioned in presented a distributed approach to network anomaly based on independent component analysis. Dimensionality reduction was achieved with the aid of the autoencoder model and essential information for the input was obtained by compression of the code layer between encoders and decoders. The principal component analysis analysis, which assists in finding hidden aspects in multivariate data, was used to extract the features of network traffic. According to the study, one component's information should not be inferred by the others hence independent component analysis was implemented. (Francesco Palmieri, 2013)

Sr No	Year	Author	Limitations
1	2017	Aminanto, M. E. and Kim, K	Less consideration of experimental techniques
2	2019	Taher, Mohammed Yasin Jisan and Rahman	Used machine learning (ML) approaches and time complexity is missing.
3	2020	Karatas, Demir and Sahingoz, 2020	This dataset lacks supportive features of IoT.
4	2021	Liu et al., 2021.	Results aren't efficient as the deep learning model
5	2020	Al-Garadi, M. A., Mohamed, Al-Ali, A. K., Du, X.	Less consideration of experimental techniques
6	2017	Baldini, G., Giuliani, R., Steri, G. & Neisse, R.	A Generic framework is followed for comparison.
7	2020	Farhin, F., Sultana, I., Islam, N., Kaiser, M. S., Rahman, M. S. & Mahmud	Need further research work.
8	2021	Ge, M., Syed, N. F., Fu, X., Baig, Z. & Robles-Kelly	No Comparative analysis of techniques
9	2019	Liang, F., Hatcher, W. G., Liao, W., Gao, W. & Yu, W. (2019)	Less consideration of experimental techniques.
10	2020	Liu, Y., Wang, J., Li, J., Song, H., Yang, T., Niu, S. & Ming, Z.	More focused on accuracy than computation resource costs
11	2020	Luo, C., Tan, Z., Min, G., Gan, J., Shi, W. & Tian, Z	Focused on a specific model
12	2019	Ma, H. & Pang, X.	Less focused on cyber security
13	2018	McDermott, C. D., Majdani, F. & Petrovski, A. V	Focused more on accuracy
14	2022	Parra, J. A., Guti´errez, S. A. & Branch, J. W.	A generic study is required
15	2019	Serrano, W.	Need more detailed analysis
16	2018	Shen, Y., Han, T., Yang, Q., Yang, X., Wang, Y., Li, F. & Wen	Only focused on CNN
17	2020	Smys, S., Basar, A., Wang, H. et al.	Focused more on accuracy
18	2019	Ullah, F., Naeem, H., Jabbar, S., Khalid, S., Latif, M. A., Al-Turjman, F. & Mostarda, L.	No comparative analysis of techniques
19	2018	Zhang, L., Zhou, G., Han, Y., Lin, H. & Wu, Y	No generic analysis of DL techniques
20	2019	da Costa, K. A., Papa, J. P., Lisboa, C. O., Munoz, R. & de Albuquerque	Less consideration of experimental techniques

Figure 1 Existing Literature Review Table

3 Research Methodology

The study methods used to identify the presence of an IoT botnet using a variety of machine learning algorithms on the UNSW NB15 dataset is covered in this section. Additionally, a basic overview of data mining and analytics approaches is provided in this section. The KDD data analysis method is the framework for the implementation of the research. KDD, which

stands for Knowledge Discovery process, is a nine-step interactive and iterative data mining process. The KDD target is determined at the outset of this process, and the application of the learned information comes to a close. Data acquisition, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge representation are the many stages of the KDD technique. Figure following shows how the KDD methodology is used in this research.

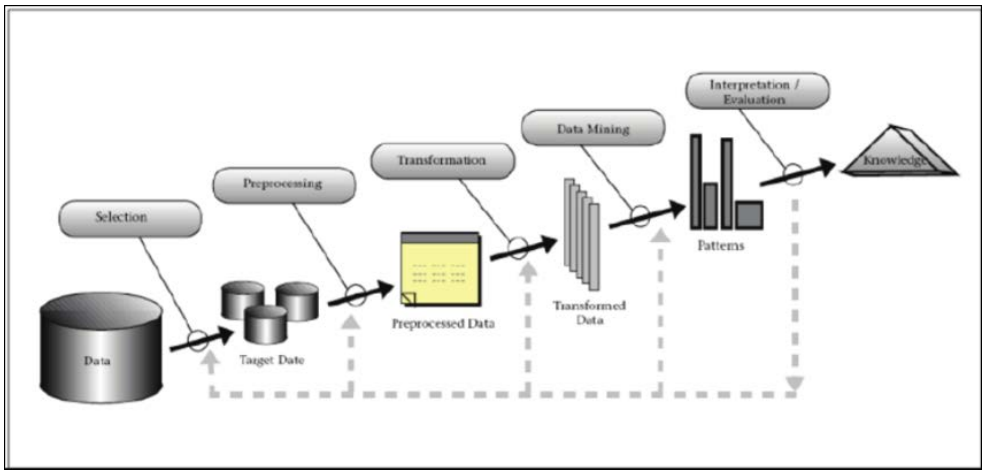


Figure 2 KDD Methodology in IoT botnet detection

3.1 Data Processing Module

Data processing often refers to the management and gathering of informational resources in order to produce useful data. The defect prediction system's data processing involves taking the data and then processing it to identify or create a useful model for later use. The built model is then tested to determine whether it is accurate or not. Training, Testing, and Evaluation are the three main components of the Data Processing Module, which also includes training data, test data, and evaluation factors.

3.1.1 Training Data

In this phase we prepare the data for calculation. It is also known as learning set. The information used in preparation includes both input data and the expected yield of comparison. As it is information from ground reality, the algorithm can develop to use innovations like neural networks to memorize and generate complex outcomes to make decisions when presented with current information later. The suggested work utilizes the publicly available data NSL-KDD dataset for the training purposes.

3.1.2 Testing Data

On the other hand, test data include input information rather than the projected output comparison. The testing data is used to evaluate the preparation quality of your calculation as well as the demonstrated properties. A conclusion demonstrate match mostly on preparation dataset may be evaluated objectively using the test dataset.

3.1.3 Evaluation Methodology

Standard evaluation measures were used in the suggested work, including recall, F1-score, precision, accuracy and confusion matrix for the proposed module. The subsections below contain the mathematical formulas.

Precision = TP/(TP+FP)
F1 Score = 2*TP/(2*TP+FP+FN)
Recall = TP/(TP+FN)
Accuracy = TP + TN/(TP+TN+FP+FN)

TP = True positive rate
TN = True negative rate
FP = False positive rate
FN = False negative rate

3.1.4 Model Summary: -

In this stage, five different models are applied to the pre-processed data. We used standard deep learning techniques on the pre-processed dataset, including Convolution Neural Networks, Random Forest, Logistic Regression, Dense Neural Network and Adaboost. The use of Adaboost and Random Forest model in this study to recognize hand movements is an innovation.

3.2 Data Pre-Processing

Pre-processing of the data was first carried out due to the size of the collection of data. In order to fit the enormous dataset into minimal capacity, the data was first transformed into a pickle. The DDOS category data separated from the entire dataset to further data preparation. The project is to use two different pre-processing processes for each of the data models with the same training and testing dataset. For all models we removed the difficulty level using panda's library. Then in the next phase we have check the null values and remove the attack flag. To do this, the trained model needs to first be tested over the test data set, and the estimates for both legitimate and fraudulent traffic classes must be validated by comparison with the actual classes. In doing so, the confusion matrices for all models would be made possible. Thus, Accuracy, Precision, and Recall were attained by the confusion matrices, and the F1 score is assessed. Additionally, an analysis of the run time is necessary.

3.3 Dataset Selection

A dataset that was already accessible in a public data repository was used for the investigation. This was done in order to prevent ethical problems from developing during the execution phase. The information was downloaded and saved locally for subsequent processing. To guarantee that participants remain anonymous, personal information was removed from the dataset. The University of New Brunswick initially invented this dataset for the examination of DDoS data. When building ML notebooks, the Label column

represents the most important component of the data because it tells whether or not the sent packets are malicious. The dataset has eighty columns, each of which is a record from the University of New Brunswick's IDS logging system. Columns are provided for both because the system divides traffic into advance and behind traffic categories. Except for the Label variable, which is a categorical variable, all the variables in the dataset are numerical.

In order to address some of the underlying issues with the KDD'99 data collection, which are detailed in, a data set dubbed NSL-KDD has been proposed (Tavallaee, et al., 2009). We believe this updated version of the KDD data set, despite still having some of the issues raised by McHugh, can still be used as an effective benchmark data set to aid researchers in comparing various intrusion detection techniques due to the dearth of publicly available data sets for network-based IDSs. The train and test sets of the NSL-KDD have a sizable number of records. With all these advantages, it is possible to conduct tests using all the data rather than just a small sample that must be selected at random. Evaluation results from different research initiatives will therefore be comparable and consistent. (Cybersecurity, 2009)

3.4 Description of features in Datasets

Name	Description
scrip	Source IP
sport	Source port
dstip	Destination IP
proto	Transaction protocol
dur	Record total duration
Sload	Source bits per second
Dload	Destination bits per second
Spkts	Number of packets from Source to Destination
Dpkts	Number of packets from Destination to Source
dsport	Destination port
attack_cat	The name of each attack category
Label	0 for normal and 1 for attack records

4 Design Specification

4.1 Purpose Scope

The five analytical models being developed and evaluated are primarily designed to improve the IoT technology's present security analytics and audit procedure to a better standard. The main goal of the design is to create a machine learning (or, to be more precise, deep learning) method that would more accurately and efficiently forecast the malware traffic entering a system. As a result, it is intended to compare and assess these models under development in terms of both accuracy and run time.

4.2 Processed Data Model

Three crucial processes are combined to create the recommended process and data model. In order to model the data for the two DL models to be used, it must first be extracted and pre-processed. To prepare for the following stage, the training and test datasets must be split up and pre-processed. The collected and pre-processed training datasets are then used to train the CNN, DNN, Random Forest, Logistic Regression, Adaboost models separately. Those various models created in such phases will then be individually tested by fitting to the same test dataset. The following Activity Model allows for observation of the process and data model:

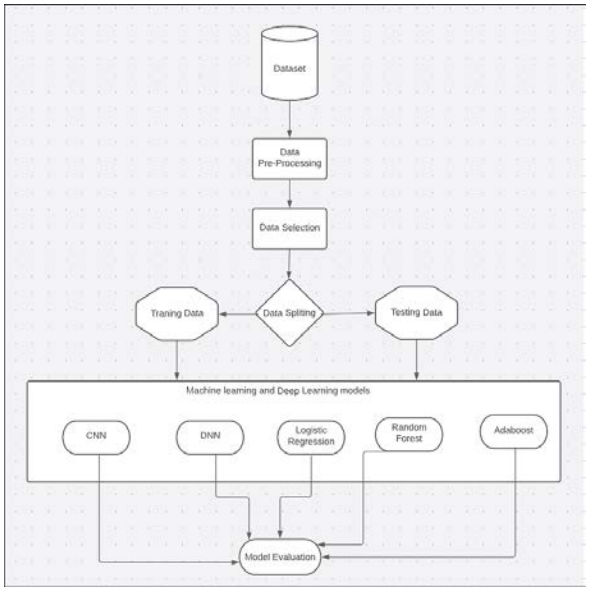


Figure 3 Process flow model

4.3 Process Model Flow

Models which used for this experiment are as follows:

- Convolution Neural Network
- Dense Neural Network
- Random Forest
- Logistic Regression
- Adaboost

5 Implementation

This section explained the implementation of IoT botnet detection using various machine learning technologies. Data preparation, cross-validation, and model refinement using hyperparameters were the three parts of the study process. The purpose of employing hyperparameters is to increase the models' accuracy in making predictions. To choose the best predictive analytic method, the results of various detection models are compared.

5.1 CNN Deep Learning Model

CNN is a neural network learning algorithm that uses feature representation to learn. The Convolution layer, also known as the Convolution feature extractor, is the first layer of the CNN in this method. From the primary feature of the images, it extracts features in batches and converts them into smaller image parameters. These characteristics include scale invariance, rotational invariance, and interpretation invariance. As a result, the overfitting issue is reduced to a collection of features that are optimized for analysis and model development. Therefore, the obtained features are fed into the next layer of the neural network.

The convolutional layer in CNN recognizes area data patterns. To provide the outcome, the input data's attributes are extracted. The most popular ANN for assessing visual imagery is a CNN (CNN or ConvNet) in DL. CNNs are standardized variants of MLP. MLP often show that all of the neurons in one layer are connected to all of the neurons in the following layer. This describes a network that is entirely interconnected. There are 3 layers that make up CNN which are Convolutional layer, Pooling layer and Fully-connected layer. In this model we have used three main pareameters such as drouptout layer, activation layer and dense layer.

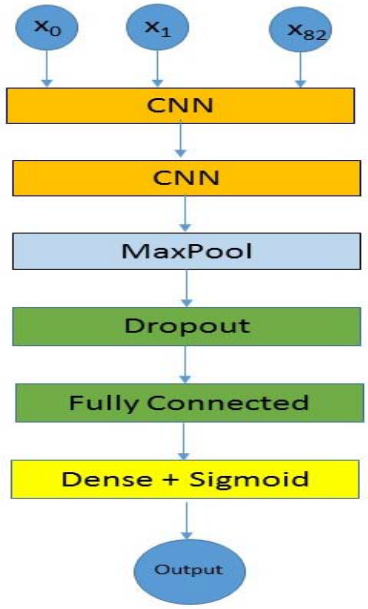


Figure 4 CNN Architecture

Activation Layer: -

We apply a nonlinear activation function, such as ReLU, ELU, or any of the various Leaky ReLU versions, after each CONV layer in a CNN. Since ReLU activations are most frequently employed, activation layers are typically denoted as RELU in network diagrams.

Dropout Layer: -

Dropout will be the final layer type we talk about. Dropout is actually a method of normalization that aims to reduce training accuracy at the expense of testing accuracy in order to assist prevent overfitting. Dropout layers randomly disconnect inputs from the previous layer to the next layer in the network infrastructure for each mini-batch in our training set with probability p. We use dropout because it explicitly modifies the network topology during training, which reduces overfitting. By intermittently discontinuing links, it is made sure that no one node in the network is in charge of "activation" in response to a specific pattern. Instead, dropout makes sure there are several redundant nodes which will activate when given comparable inputs; this, in turn, aids in the generalization of our model.

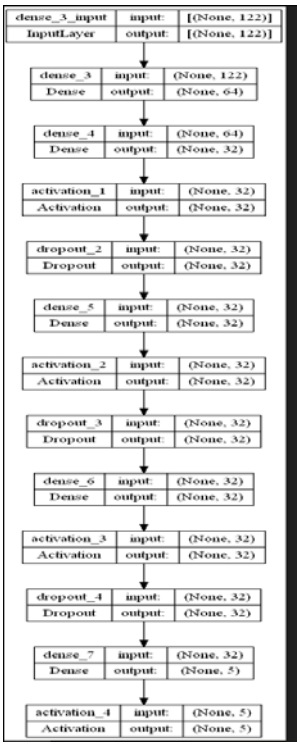


Figure 5 CNN Process

5.2 Adaboost Classifier Model

A boosting ensemble model for binary classifications is the AdaBoost classifier. This approach categorizes by strengthening a collection of weak classifiers. AdaBoost is typically combined with standard decision trees. Additionally, training data that is difficult to predict is valued more highly, whereas training data that has been simple to predict is provided less weight. Any machine learning algorithm can benefit from using this boosting approach, but the decision tree model is the one that it performs best with decision tree model. Therefore, a simple AdaBoost model was implemented for this study. A typical boosting algorithm is Adaboost. The addition model and the next step-by-step algorithm are the two fundamental

components of the Adaboost algorithm. A sequence of weak classifiers is linearly combined to create a strong classifier in the addition model.

```
#AdaBoost with decision tree
decision_clf = DecisionTreeClassifier()
clf = AdaBoostClassifier(decision_clf)
clf.fit(X_train,Y_train)
train_acc = clf.score(X_train, Y_train)
test_acc = clf.score(X_test, Y_test)
y_pred = clf.predict(X_test)
print("Training accuracy is:", train_acc )
print("Testing accuracy is:", test_acc)
```

```
Training accuracy is: 0.999833297611393
Testing accuracy is: 0.7649041873669269
```

Figure 6 Code Snippet of Adaboost model

5.3 Random Forest Classifier Model

We implemented Recursive Feature Elimination (RFE), which trains a specified ML model on all available feature subsets before discarding all the weak features until only a predetermined number of features remain. When using a random subset of samples to train multiple classification trees, random forest produces predictions by using the majority vote of these trees on test data. This study used a random forest classifier with hyperparameter tuning to accurately identify IoT botnet. This study uses the time function to take into account both model training and model detection times.

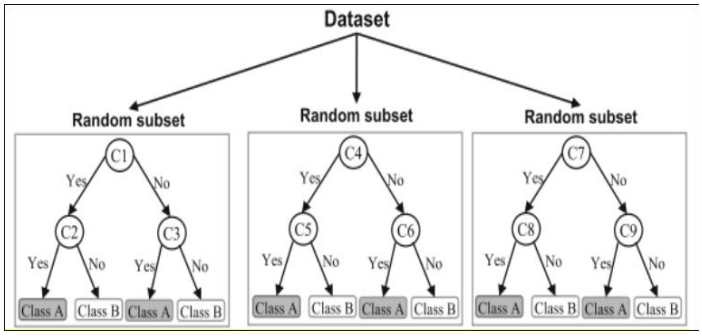


Figure 7 Random Forest Model

```
RandomForest_f1 = f1_score(Y_test, y_pred, average="macro")
RandomForest_precision = precision_score(Y_test, y_pred, average="macro")
RandomForest_recall = recall_score(Y_test, y_pred, average="macro")
RandomForest_accuracy = accuracy_score(Y_test, y_pred)
print("RF_metrics")
print("f1score",RandomForest_f1)
print("precision",RandomForest_precision)
print("recall",RandomForest_recall)
print("accuracy",RandomForest_accuracy)
```

```
RF_metrics
f1score 0.5028122750318944
precision 0.7333324924043723
recall 0.49000691836548765
accuracy 0.7588715400993612
```

Figure 8 Code snippet of RF model

5.4 Logistic Regression

An approach used to overcome classification issues is logistic regression. A limited set of classes are assigned to observations by logistic regression. It uses both discrete and continuous datasets to classify new data, and it can generate probabilities. It uses the sigmoid function to transform its output into a probability value and then returns that value. The sigmoid function is used to convert anticipated values into probabilities. It can provide a likelihood for an event to occur or not based on input variables (in term of 0 and 1).

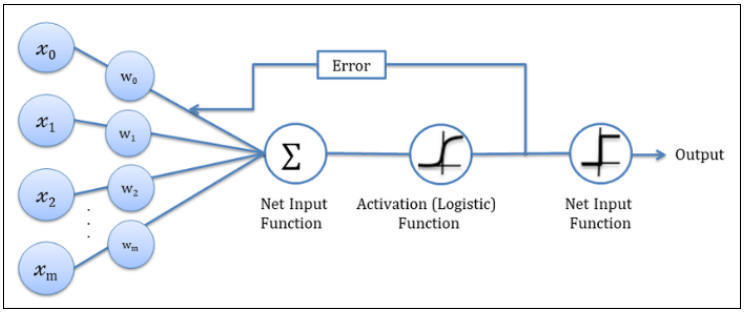


Figure 9 Logistic Regression Model

Additionally, multinomial effects of logistic regression are possible, such as the prediction of the chosen cuisine from a menu, such as Jamaican, Spanish, Irish, etc. A categorical target variable is estimated using the logistic regression method. Logistic regression is a widely used a categorical outcome variable is calculated using the logistic regression method. Researchers and analysts frequently utilize the data mining and statistical technique of logistic regression to locate, categorize, and evaluate proportional and binary response datasets. Logistic regression has many benefits, including the fact that the input characteristics don't need to be scaled, that it is very effective and simple to train, that it is not difficult to apply, that it is simple to regularize, and that it may be used to solve multiclass classification issues. Most frequently, problems at an industry-scale level can be solved using logistic regression. Most methods used to examine logistic regression models employ the same linear regression concepts.

```
Logistic_f1 = f1_score(Y_test, y_pred, average="macro")
Logistic_precision = precision_score(Y_test, y_pred, average="macro")
Logistic_recall = recall_score(Y_test, y_pred, average="macro")
Logistic_accuracy = accuracy_score(Y_test, y_pred)
print("Logistic_metrics")
print("f1score",Logistic_f1)
print("precision",Logistic_precision)
print("recall",Logistic_recall)
print("accuracy",Logistic_accuracy)
```

```
Logistic_metrics
f1score 0.48346192792657805
precision 0.6654614640283338
recall 0.490342062495691
accuracy 0.7498669268985095
```

Figure 10 Code snippet of Logistic Regression Model

5.5 Dense Neural Network

The MLP, a subset of the DNN, is generated in this phase. The keras library backend of TensorFlow was applied to generate this classifier. With an approximate execution duration of 13 seconds, this model provides 100% accuracy. The Adadelta (Adam) optimizer is a more powerful version of Adagrad that alters training rates based on a moving gradient update window rather than adding up all of the prior gradients. This way, Adadelta keeps picking up new skills even after many upgrades.

```
from sklearn.metrics import f1_score, precision_score, recall_score, accuracy_score
print('Precision: %.3f' % f1_score(true_lbls_dnn, pred_lbls_dnn, average='weighted'))
print('Precision: %.3f' % precision_score(true_lbls_dnn, pred_lbls_dnn,average='weighted'))
print('Recall: %.3f' % recall_score(true_lbls_dnn, pred_lbls_dnn, average='weighted'))
print('Accuracy: %.3f' % accuracy_score(true_lbls_dnn, pred_lbls_dnn))
```

```
Precision: 0.756
Precision: 0.824
Recall: 0.789
Accuracy: 0.789
```

Figure 11 Code Snippet of DNN

6 Evaluation

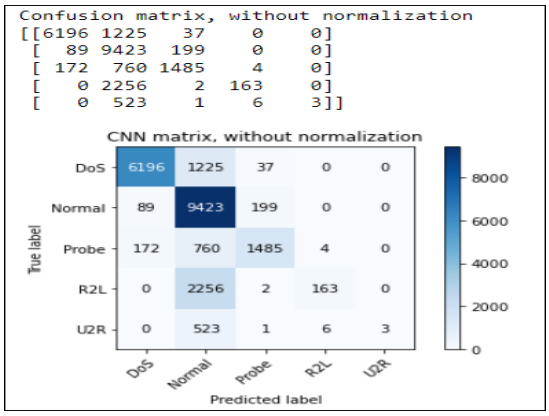
This section provides an example of the various experiments which were carried out to assess the efficacy and performance of the categorization models used in this research. The IoT botnet was exactly discovered using the classification methods CNN, Random Forest (RF), AdaBoost, KNN and Logistic Regression. The outputs of the confusion matrix will be used to compare the performance of the developed models. Using visual bar charts, plots to show the results. Accuracy, Precision, Recall, and F1-Score can be calculated for each detection models using the confusion matrix and these evaluating metrics are defined as follows.

6.1 Experiment 1 CNN Classifier

The convolutional layer in CNN recognizes regional data patterns. To provide the outcome, the input data's attributes are extracted. The most popular ANN for assessing visual imagery is a CNN (CNN or ConvNet) in DL. CNNs are conventional types of MLP. MLP normally indicate that all single neurons in one layer are connected to all of the neurons in the following layer. This describes a network that is entirely interconnected. This DL model executes in around 11 seconds and provides a 97% accuracy rate. Due to the training data's

size being selected at 95% of the total data at random. The execution time and sample accuracy vary with each running of the programme. However, the accuracy is often 95% in an average for the models.

Model	Training Accuracy	Testing Accuracy	Precision	F1-Score	Recall
CNN	95%	97%	92%	81%	16%



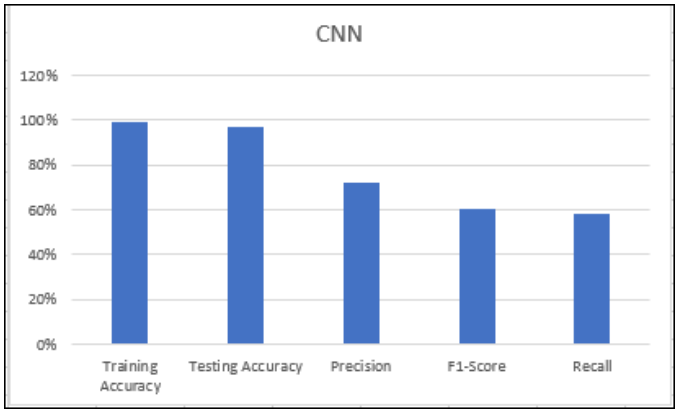
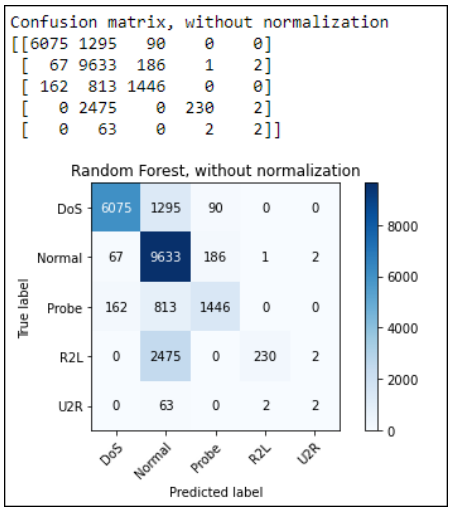


Figure 12 Confusion Matrix and bar chart of CNN model experiment

6.2 Experiment 2 Random Forest Classifier

A technique known as the Random Forest (RF) bagging ensemble generates a collection of decision trees at random from the training subset. To determine the object class, this technique aggregates the votes from decision trees that were built at random. An accuracy of 87%, precision of 81%, recall of 48% and F1-score of 50% were obtained using the Random Forest model. The evaluation indicators and outcomes of the Random Forest model are shown in the bar chart.

Model	Training Accuracy	Testing Accuracy	Precision	F1-Score	Recall
Random Forest	99%	75%	81%	50%	48%



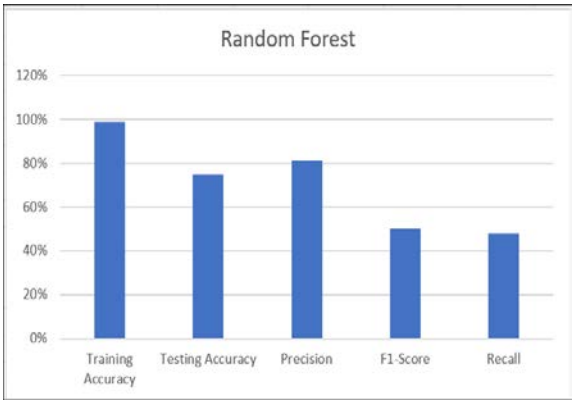


Figure 13 Confusion Matrix and bar chart of Random Forest model experiment

6.3 Experiment 3 Dense Neural Network Classifier

The Dense Neural Network, a component of machine learning techniques based on the artificial neural network, is the final model used in this investigation effort. It is used to provide learning features from all the combinations of the features of the preceding layers. After some iterations, the dense neural network used in this study produced an accuracy of 91.14% with a validation loss of 17.58%. So, 88% is the final mean accuracy for this model.

Model	Training Accuracy	Testing Accuracy	Precision	F1-Score	Recall
DNN	99%	78%	82%	75%	78%

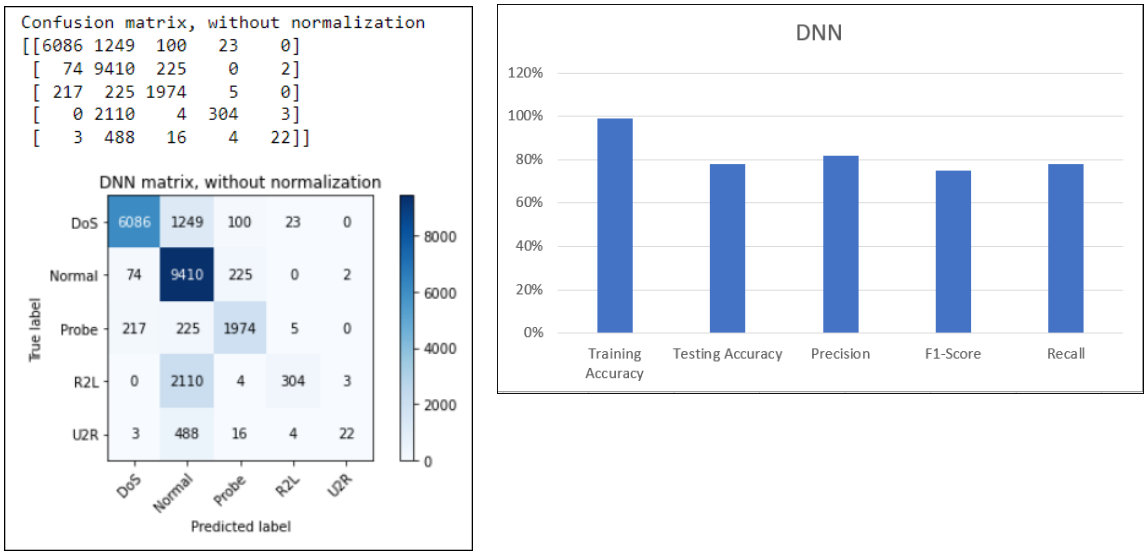


Figure 14 Confusion Matrix and bar chart of DNN model experiment

6.4 Experiment 3 Logistic Regression Classifier

The construction of an LR model is simple, and model training is effective. However, LR's use is constrained because it struggles with nonlinear data. The accuracy rate with Logistic Regression Model was 86% with precision 66%, recall 49% and F1-score of 48%. The evaluation metrics and outcomes are shown in the bar chart.

Model	Training Accuracy	Testing Accuracy	Precision	F1-Score	Recall
Logistic Regression	97%	74%	66%	48%	49%

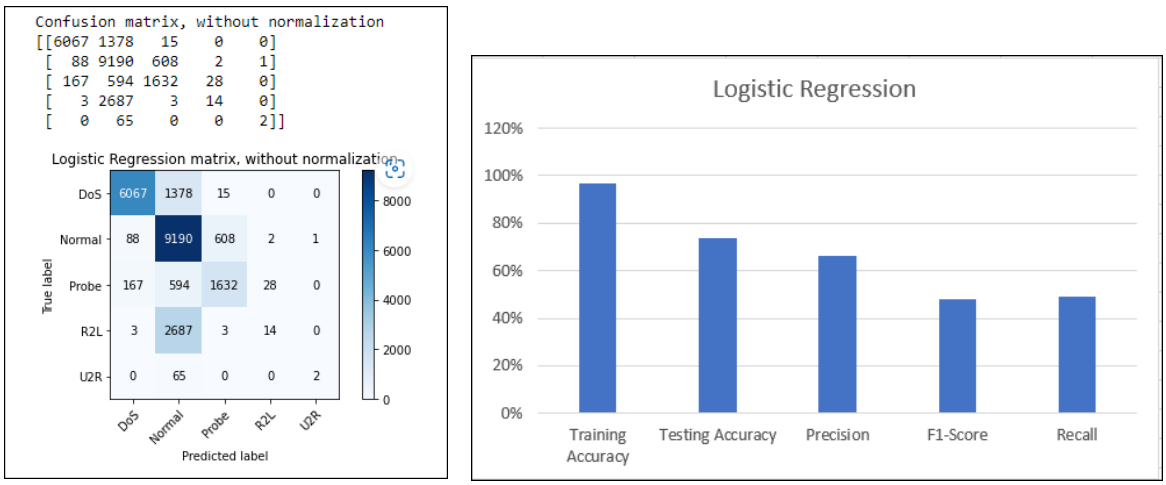


Figure 15 Confusion Matrix and bar chart of Logistic Regression model experiment

6.5 Experiment 4 Adaboost Classifier

Typically, the AdaBoost ensemble approach is used to improve the performance of classification model. To learn from the errors of the weak classifiers and transform them into strong classifiers, it uses an iterative methodology. AdaBoost is a boosting ensemble technique that improves accuracy while reducing bias. The accuracy rate with AdaBoost classifier was 87%, with precision of 82%, recall 51% and F1-score of 51%. The AdaBoost Classifier model's evaluation metrics and outcomes are shown in the bar chart.

Model	Training Accuracy	Testing Accuracy	Precision	F1-Score	Recall
Adaboost	99%	75%	82%	51%	51%

Figure 16 Confusion Matrix and bar chart of Adaboost model experiment

6.6 Discussion

The proposed research demonstrates how machine learning techniques could be used to accurately detect an IoT botnet by analyzing network traffic data. On the IoT Intrusion dataset, an investigation of the Random Forest, CNN, AdaBoost, and Logistic Regression was carried out. The results showed that the CNN and Random Forest classifier has the

potential to outperform other machine learning approaches and deep learning techniques when metrics like training time, detection time, accuracy, F1-score, precision, and false rate were taken into consideration. The implementation of classification minimized the false positive rate and detection time without compromising performance. Despite the good accuracy that all five machine learning approaches attained, random forest clearly outperformed with a faster detection rate.

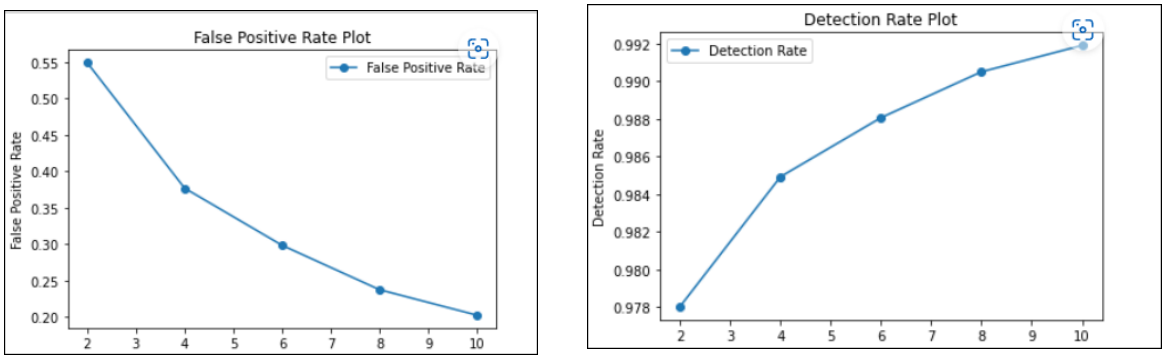


Figure 17 False Positive and Detection plot

7 Conclusion and Future Work

The focus of this implementation is on false negatives while evaluating models based on precision, recall, accuracy, and F- 1 score because the key objective of this research is to improve the NIDS by applying ML and DL models in order to determine which models are more beneficial in this process. Each model outputs a different accuracy and takes a different amount of time to compute. On the basis of the confusion matrix, it is possible to determine which model provides the best accuracy and the fewest false negatives. Because more false negatives could hurt our system. In this paper, a recommended Convolutional Neural Network (CNN) classifier-based Intrusion Detection System (IDS) for cybersecurity is presented. The proposed IDS model can learn complex structures of features using network traffic while maintaining reasonable storage and processing overhead due to CNN's convolution and pooling mechanism. This feature distinguishes the proposed IDS model from the conventional one, which necessitates the creation of a signature database by security professionals in order to perform categorization. Additionally, a cutting-edge dataset pre-processing technique is applied to improve the multiclass classification performance of the suggested CNN-based IDS.

As demonstrated by this research, some ML models consistently produce results that are more accurate than DL models, whereas neural networks produce outcomes that are more accurate than DL models in less computing time. Therefore, it can be concluded that when relates to neural networks, DL models can also provide better accuracy than any other model, however the CNN and Random Forest algorithms are the best fit models for this dataset. The CNN algorithm achieved 97% accuracy with precision of 92 %. On the other hand, Random Forest algorithm achieve 87% with precision of 81%. However, DL algorithms are much more preferred in cases of huge and complicated datasets for greater accuracy and validation.

Limitations of this Research:
For all of the models, a single dataset is used, and just one model's visualization is displayed. We were unable to build our own IoT dataset by simulating an IoT network due to time and resource limitations. Moreover, IoT networking was not widely supported by open-source simulators. However, it was not possible to extract the network information, hence the UNSW NB15 dataset was used in this study.

Future Scope:
The project's artifact can be observed in a graphical interface. Additionally, these models can be deployed with different datasets. Other intricate models might be included in this project. In the future, open source IoT simulators can be used to extract IoT network traffic data, and that data can then be deployed to train machine learning models to accurately identify the network system.

8 Acknowledgment

I want to express my gratitude to Mr. Michael Prior, my supervisor, for his unwavering support and direction. Throughout the research endeavour, Mr. Michael Priorconsistently encouraged and was eager and willing to help in any way he could. I genuinely appreciate all the inspiration and drive he has given me to advance my study.

9 References

Al-Garadi, M. A. et al., 2019. *A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/9072101
[Accessed 12 August 2022].

Aminanto, M. E. & Kim, K., 2022. *Koasas.kaist.ac.kr.* [Online]
Available at: https://koasas.kaist.ac.kr/bitstream/10203/214353/1/IRCET16_AM.pdf
[Accessed 13 August 2022].

Baldini, G. & Raimondo Giuliani, F. D., 2018. *Physical layer authentication of Internet of Things wireless devices using convolutional neural networks and recurrence plots.* [Online]
Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.81
[Accessed 13 August 2022].

Costa, K. A., Albuquerque, V. H. C., Munoz, R. & Papa, J. P., 2018. *Internet of Things: A survey on machine learning-based intrusion detection approaches.* [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S1389128618308739#!
[Accessed 13 August 2022].

Cybersecurity, C. I. f., 2009. *NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB.* [Online]

Available at: https://www.citethisforme.com/cite/sources/websiteautociteconfirm
[Accessed 10 August 2022].

Farhin, F. et al., 2021. *Attack Detection in Internet of Things using Software Defined Network and Fuzzy Neural Network.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/9306666
[Accessed 13 August 2022].

Francesco Palmieri, U. F. C., 2013. *https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3061.* [Online]
Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3061
[Accessed 13 August 2022].

Huma, Z. E. et al., 2018. *A Hybrid Deep Random Neural Network for Cyberattack Detection in the Industrial Internet of Things.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/9399085
[Accessed 14 August 2022].

Jhon Alexánder Parra, S. A. G. J. W. B., 2022. *A Method Based on Deep Learning for the Detection and Characterization of Cybersecurity Incidents in Internet of Things Devices.* [Online]
Available at: https://arxiv.org/abs/2203.00608
[Accessed 13 August 2022].

Karatas, G., Demir, O. & Sahingoz, O. K., 2018. *Deep Learning in Intrusion Detection Systems.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8625278
[Accessed 11 August 2022].
Liu, Y. et al., 2020. *Zero-Bias Deep Learning for Accurate Identification of Internet-of-Things (IoT) Devices.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/9173537
[Accessed 12 August 2022].

Luo, C. et al., 2021. *A Novel Web Attack Detection System for Internet of Things via Ensemble Classification.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/9261992
[Accessed 13 August 2022].

McDermott, C. D., Majdani, F. & Petrovski, A. V., 2020. *Botnet Detection in the Internet of Things using Deep Learning Approaches.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8489489
[Accessed 13 August 2022].

MengmengGe, et al., 2021. *Towards a deep learning-driven intrusion detection approach for Internet of Things.* [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S138912862031358X
[Accessed 11 August 2022].

NickolaosKoroniotisNourMoustafa, Turnbull, B. & Sitnikova, E., 2019. *Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset.* [Online]

Available at: https://www.sciencedirect.com/science/article/pii/S0167739X18327687
[Accessed 12 August 2022].

Pang, X. & Ma, H., 2018. *Research and Analysis of Sport Medical Data Processing Algorithms Based on Deep Learning and Internet of Things.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8809757
[Accessed 12 August 2022].

Serrano, W., 2019. *The Blockchain Random Neural Network in Cybersecurity and the Internet of Things.* [Online]
Available at: https://link.springer.com/chapter/10.1007/978-3-030-19823-7_4
[Accessed 13 August 2022].

Shen, Y. et al., 2019. *Shen, Y., Han, T., Yang, Q., Yang, X., Wang, Y., Li, F. & Wen H. (2018), Cs-cnn: Enabling robust and efficient convolutional neural networks inference for internet-of-things applications', IEEE Access 6, 13439–13448.*. [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8304540
[Accessed 12 August 2022].

Smys, D. S., Basar, D. A. & Wang, D. H., 2020. https://uniquescientificpublishers.com/wp-content/uploads/2022/ahp-v2/112-117.pdf. *International Journal of Veterinary Science,* Issue Volume 2, pp. 118-125.

Taher, K. A., Jisan, B. M. Y. & Rahman, M. M., 2022. *Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8644161
[Accessed 10 August 2022].

Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A., 2009. *A detailed analysis of the KDD CUP 99 data set.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/5356528
[Accessed 10 August 2022].

Ullah, F. et al., 2019. *Cyber Security Threats Detection in Internet of Things Using Deep Learning Approach.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8812669
[Accessed 13 August 2022].

Zhang, L. et al., 2018. *Application of Internet of Things Technology and Convolutional Neural Network Model in Bridge Crack Detection.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/8410506
[Accessed 14 August 2022].