DETECTING SECURITY BREACH USING ARTIFICIAL NEURAL NETWORK

MSc Research Project

MSc Cyber Security

Mohammed Sharfuddin Hyder

Student Id: x21150362

School of Computing

National College of Ireland

Supervisor: Dr. Vanessa Ayala-Rivera

National College of Ireland

MSc in Cyber Security

School of Computing

Student Name:	MOHAMMED SHARFUDDIN HYDER ………………………………………………………………………………………………..
Student ID:	x21150362 ….……………………………………………………………………………………….……
Programme:	MSc in Cybersecurity ……………………………………………………………… **Year:** 2022 ……………..…..
Module:	MSc Research Project ……………………………………..……………………………………………………….………
Supervisor:	DR. VANESSA AYALA-RIVERA …………………………………………………….…………….…………..……………………….………
Submission Due Date:	08-MAR-23 ……………………..………………..………………………………………………………….………
Project Title:	IMPLEMENTATION OF ANN IN DETECTING SECURITY BREACH ……….………

Word Count: ……6726…………………………… **Page Count**……………………18……………….……..

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: Mohammed Sharfuddin Hyder ……………………………………………………………………………………………………

Date: 08-MAR-2023 ……………………………………………………………………………………………………

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):

DETECTING SECURITY BREACH USING ARTIFICIAL NEURAL NETWORK

Mohammed Sharfuddin Hyder
21150362

Abstract:

Networks are a part of almost every important sector in the world today. The amount of data being transferred through these networks is huge, and it is important that this data be protected as it may be of high value to a number of companies and individuals. But these networks are highly vulnerable, and people with malicious intent are always on the lookout for ways to exploit these vulnerabilities. As breaches of networks are becoming very frequent, in order to identify and classify these data breaches or attacks, a system must be developed. Thus, a method to detect and classify the various types of network intrusions is suggested here. A model of an artificial neural network (ANN) will serve as the foundation for the system's development. Using the CSE-CIC-IDS 2022 dataset's data, the ANN will be trained. The technique known as Analysis of Variance (ANOVA) will be used to pick the dataset's most crucial properties. The dataset will need to be balanced because of the imbalance; hence the Synthetic Minority Oversampling Method (SMOTE) will be utilised. The technology will be put into use as a desktop programmed that can identify the kind of intrusion that has taken place on a network based.

1. Introduction:

Information and communication technology (ICT) advancements and extensive use of interconnectivity and coordination are now required to modify our interactions with daily activities. The attitude of relying on ICT has improved the positions of people and organizations, enabling real-time global business continuity that constantly develops to provide convenience-related frontier solutions that manage the communication and connection of different devices, systems, and applications in a coordinated manner [1].

An efficient network security solution is essential for preserving availability, secrecy, and authenticity. Since the sharing of digital information over networks has created a way for the vulnerabilities in the system to be exploited, which may have negative repercussions on both people and organizations [2]. Most firms face a serious threat from network security breaches. Governments and companies are working very hard to prevent the network leakage of critical data. Networks are also a significant part of many technological advancements like IoT so the threats associated with networks automatically carry over to these technologies as well. A network can be secured using a variety of technologies, including firewalls, and traffic shaping devices.

Among the defensive mechanisms that have multiple layers and can handle a variety of attack vectors, network security measures are recognised as the first line of protection. Additionally, a variety of attack modelling techniques are available to assist businesses in comprehending the nature of an attack [3]. Cyberattack identification, prevention, and reduction are frequently successful with the use of robust authentication, monitoring, and access control systems. One mechanism that helps in fighting the security threats to a network is an intrusion detection system (IDS).

In order to detect an attack based on the pre-defined, tailored detection levels, an IDS examines network traffic. An intrusion will be stopped and removed from the system if it is detected early and effectively to prevent data loss, as shown in Fig 1. IDS evaluates intrusion behavior according to its features since it assumes that the behavioral characteristics of intrusions differ from those of legitimate users. Because an absolute separation is impossible, a successful IDS can highlight the overlap between legitimate and malicious actions. The bulk of attacks can also be detected by IDS at the Transport, Network and Perception layers.

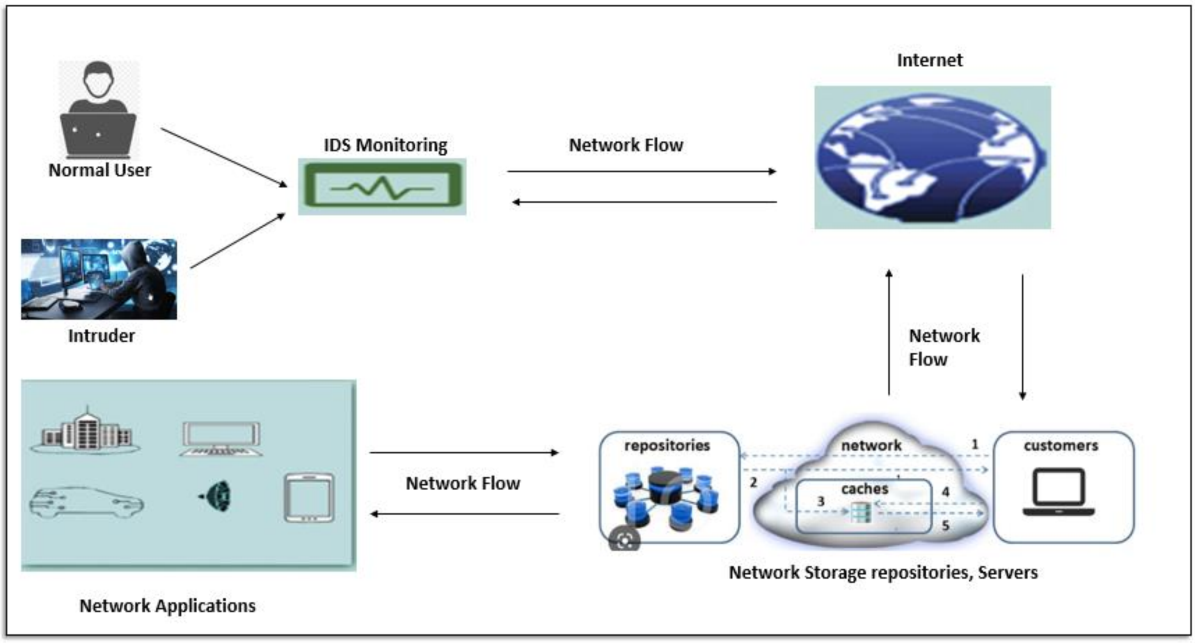


Fig 1. Working of an IDS [4]

The three primary categories of IDS are cloud intrusion detection systems, host intrusion detection systems, and network intrusion detection systems (NIDS) (CIDS). The NIDS may be composed of both software-like consoles and hardware like sensors for managing and keeping track of network traffic packets at various places for any potential intrusions or attacks. The HIDS monitors only activities on the host system, which is a specific server or computer. The CIDS is a combination of host, network, and cloud layers. For performing detection of security breaches, the IDS use different techniques. These methods include signature-based detection, where the network is examined to detect patterns that match known or existing signatures.

The network traffic is examined in anomaly-based detection to find patterns that differ from expected behavior. Popular protocol profiles are compared to network traffic in state full protocol analysis [5]. There are drawbacks to each strategy when tackling the intersection of normal and abnormal traffic patterns. Some of these flaws result in increased CPU usage, decreased network speed, and false negatives and false positives. Traditional machine learning techniques are used to overcome these shortcomings in detection. Although these techniques have substantially increased detection accuracy, processing the vast amounts of data still calls for professional expertise and engagement. These methods, also known as "shallow classifiers," can eventually produce subpar results for multiclass issues with more features since they let computers train without human assistance. Self-learning intrusion detection systems that can identify and categorise intrusions have further evolved thanks to research; these methods enable proactive steps to spot and stop malicious network traffic [6]. Deep learning is referred to as the complicated model or sophisticated subset of machine learning algorithms, which addresses some of the drawbacks of conventional shallow machine learning models. Deep learning approaches have shown to be effective in a variety of domains, including speech recognition, picture processing, and natural language processing. [7]. So, an IDS for detecting and classifying network breaches or intrusions based on deep learning will be very useful and effective for improving the security of a network.

1.1 Research question:

How can ANN help us detect security breaches? And how will an IDS developed using deep learning methods be able to detect and classify intrusions in a network accurately?

1.2 Aims

- Create a dataset that includes information on both typical networks and networks where cyberattacks have taken place.
- Remove the unwanted components from the data and make the data clean.
- Choose the most important features from the data for training the deep learning models.
- Create and train the deep learning based ANN model.
- Implement the IDS in the form of a desktop application.
- Evaluate the performance of the IDS in detecting and classifying network breaches or intrusions.

2. Related Work:

2.1 IDSs based on machine learning techniques:

An approach that used an IDS made up of a decision tree (DT) with improved quality of data was proposed in [8]. The pre-processed network data is used to choose the important features from the data using an entropy decision feature selection method. Both the CICIDS-2017 and the NSL-KDD datasets are used to train the DT model. With the NSL-KDD dataset, the trained DT model had a 99.42% accuracy rate, while for the CICIDS2017 dataset, it had a 98.8% accuracy rate. The data processing and feature selection techniques were found to improve the IDS's detection accuracy, according to the outcomes of this methodology. The machine learning model being employed here is unstable, which is the fundamental shortcoming of this strategy.

An anomaly-based IDS in the presence of long-range independence data known as benign outliers was created by utilising a neural projection architecture obtained using a selforganizing map (SOM) and proposed in [9]. The system accurately detects attacks and anomalies and also gives end users visible information and insights. Several datasets, including AAGM, NSL-KDD, VPN, non-VPN, and NSL-KDD, were used to assess the system's performance. It was noted that the system can successfully detect intruders. Because some of the datasets used in this approach are not the most recent, the results from this approach cannot be fully characterized as relevant because newer datasets are not utilized.

A method for detecting intrusions from IoT streaming data in real time based on machine learning was proposed in [10]. Different machine learning algorithms and datasets are utilised in this approach. Apache Spark Streaming and Apache Flink, two frameworks for stream processing, were used to construct and test this, but the results and studies associated with this approach are mainly focused on the IoT streaming data, and insights about the approach on a general network are limited. The intrusions in IoT networks are detected using machine learning methods [11] The UNSW-NB15 dataset is used in this approach, and features in the data are scaled using the min-max scalar. The machine learning algorithms like extreme gradient boosting (XgBoost), cat boost, K nearest neighbour (KNN), This approach makes use of Nave Bayes (NB), support vector machine (SVM), and quadratic discriminant analysis (QDA), and it was found from the results that the XgBoost and CatBoost algorithms both produce the best accuracies, with both achieving a value of 99.9%. It can be observed from this approach that feature scaling can be used for making the IDS more effective. The key drawback of the strategy is that because the data is unstructured, it might have an impact on the system's performance. A system for detecting DDoS attacks in IoT networks using DT was prposed. This approach reveals that feature selection methods result in the attack's detection accuracy increasing. The system here has a 99.98% accuracy rate. The biggest drawback of this is that just one form of assault is recognised, and it also takes a long time to train the models.

A majority voting approach is used for detecting and classifying intrusions from network traffic in [12]. The NSL-KDD dataset is used in this approach. It can be seen that the system proposed achieves an accuracy of 99.5%. Here, also, the importance of feature selection is described, as the approach is only effective along with the Chi-Square feature selection technique. The main limitation of this approach is that misclassifications are sometimes performed using the majority voting technique.

A technique for improving the performances of machine learning models in intrusion detection is proposed in [13]. In this approach, a technique named PCC-LSM (Pearson Correlation Coefficient-Least Square Method) is applied to the IDS for improving its performance. The dataset used here is the UNSW-NB15 dataset. A feature selection method based on PCC is utilised in this case to ensure that the IDS operates as effectively as possible, emphasising the significance of feature selection. This approach's primary drawback is the lack of in-depth research on the PCC-effects LSM's on intrusion detection.

To find network traffic intrusions, a two-tier classification model is utilized [14]. It consists of the NB and the Certainty Factor versions of KNN. In order to minimize the dimensionality of the data, the NSL-KDD dataset is employed, along with linear discriminate analysis (LDA) and component analysis. 84.82% accuracy was attained using this method. This method's primary flaws include a high percentage of false positives and poor detection precision.

For intrusion detection, a hybrid classification system that makes use of both the Artificial Fish Swarm (AFS) and Artificial Bee Colony (ABC) was used in [15]. This method does feature selection utilising fuzzy C-Means clustering (FCM) and correlation-based feature selection (CFS). The detection of intrusions is based on rules generated using the Classification and Regression Tree (CART) algorithm. This method uses the UNSW-NB15 and NSL-KDD datasets, and it is shown that the system is capable of a 99% detection rate. But even the slightest changes in the data used here are likely to make the tree unstable and hinder the performance of the system.

The ability of the feature selection techniques Sequential Feature Selection (SFS) and ANOVA in enhancing intrusion detection is studied in [16]. It is clear from the study's findings that the ANOVA can improve an intrusion detection system's performance by helping to select the best features and boost multi-class detection accuracy by more than 10%. This approach was used in a number of studies that used the ANOVA and SFS for intrusion detection. The system's biggest drawback is that it only studies previous techniques that employed ANOVA or SFS; no such system is currently in use.

For an IDS, the RF model was trained based on the CSE-CICIDS-2018 dataset [17]. This approach reveals that the CSE-CICIDS-2018 is an enhanced dataset compared to other datasets like the NSL-KDD used in intrusion detection. The CSE-CICIDS-2018 contains the latest and most sophisticated threats to networks and is observed to be highly effective in intrusion detection, as the IDS achieves an accuracy of 99%. The approach's primary drawback is that the model it produces could be harder to understand.

2.2 IDSs based on deep learning techniques:

A system that is dynamically scalable and is able to detect and classify network intrusions is proposed in [18] as shown in Fig 2. The detection and classification of intrusions in this approach are performed by the combination of the extreme learning machine (ELM) model, an ensemble model, and a logistic regression (LR) layer. The results show that the system is capable of accurately detecting and classifying network intrusions. The machine learning model is trained using the UNSW-NB15 dataset. This method demonstrates the significance of balancing the dataset's data points for successful intrusion detection. This method also demonstrates the significance of feature selection. The fundamental drawback of this strategy is that because it uses the ELM, its hidden layer could be very complex.

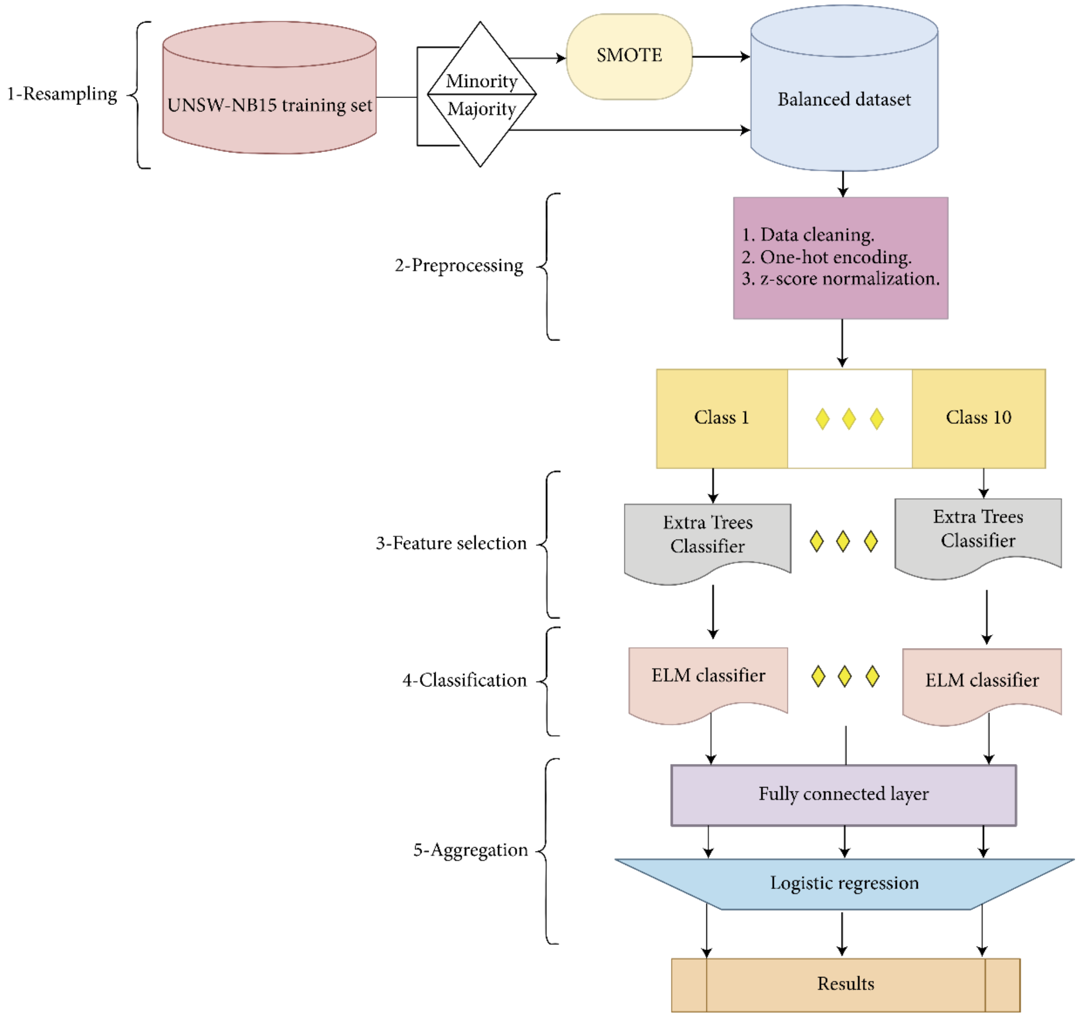


Fig 2. Deep learning-based Network IDS (Moualla, Khorzom and Jafar, 2021)

For wireless network intrusion detection, a Wrapper Based Feature Extraction Unit (WFEU) and Feed-Forward Deep Neural Network (FFDNN) are employed [19]. In this method, only the best features are chosen using the WFEU, and the FFDNN is trained using only these best features, producing a very good outcome. In this method, the AWID and UNSW-NB15 datasets are employed. For the UNSW-NB15 dataset, a multi-class classification accuracy of 77.16% and an AWID dataset accuracy of 99.77% are respectively obtained. This strategy highlights the significance of optimal features. This strategy's primary drawback is how long the training process takes.

Network intrusions are detected and classified using the Deep Neural Network (DNN) in [17]. This method uses the UNSW-NB15 dataset, and it is seen that the system detects intrusions with a 95.4% accuracy rate. This strategy demonstrates that methods based on deep learning are efficient at identifying intrusions. The main limitation of the approach is that a feature reduction or selection technique that may improve the performance of the system is not used here.

Both binary and multi-class classification of networks were carried out using the DNN in [20]. This method uses the KDD Cup 99 dataset, and the system achieves multi-class classification accuracy of 99.98%. The detecting method is more successful thanks to the system's capacity to learn several complicated features. The biggest drawback of this strategy is how slowly and laboriously long the training is. This restriction also pertains to the technique suggested in [21]. This method employs DNN and a hybrid feature selection strategy for intrusion detection. Using a method that combines the Principal Component Analysis (PCA), Chi Square, and ANOVA (Analysis of Variance) approaches, the approach's feature selection is carried out. The accuracy of the method suggested here, and the NSL-KDD dataset utilised in this technique is 99.73%. This method demonstrates the ANOVA's great effectiveness as a feature selection method.

Long Short-Term Memory (LSTM) model, a deep learning-based technique, is employed in [22] for intrusion detection. The CSE-CIC-IDS 2022 dataset is utilized in this approach, which demonstrates that the dataset can be used to train a deep learning model to yield results of a high calibre, as is obvious from the approach because the system obtains a 99% accuracy rate. Because it is dependent on the unbalanced CSE-CIC-IDS 2022 dataset, this method has a number of limitations.

In IoT networks, the DNN is used to provide flow-based intrusion detection [23]. This method trains the DNN, which achieves an accuracy of 90.25 percent, using the CSE-CIC-IDS2018 dataset. Deep learning models are utilised to demonstrate once more how well the CSE-CIC-IDS2018 dataset performs in intrusion detection. This approach's main drawback is that a vast amount of data is not handled adequately, and no feature selection method is employed.

In [24], ANN is used to detect network intrusions. This method uses the KDD CUP 99 dataset, and it is discovered that the ANN achieves an accuracy of 97.97%. This method demonstrates how well the ANN detects data leaks. The system's primary drawback is that it uses the comparatively dated KDD CUP 99 dataset and doesn't employ any feature selection techniques.

The ANN is used for detecting shell code in networks, as proposed in [25]. The ANN is trained using a dataset containing shell code and normal code, and it detects shell code intrusions in a network with an accuracy of 98%. The primary drawback of this strategy is that it exclusively concentrates on the detection of shell codes, ignoring other types of intrusions.

2.3 Summary:

Systems that now use machine learning and deep learning to identify and categorise data breaches or intrusions were investigated. From these studies, it is evident that deep learning techniques are highly effective in intrusion detection and classification. It was found from this literature that the data associated with networks is imbalanced, and so the data needs to be balanced using balancing techniques. The importance of feature selection in creating an effective IDS was highlighted in a number of publications. From these feature selection techniques, the ANOVA technique was observed to be highly effective. The ANN was also observed and was highly effective when combined with a feature selection technique. It was also observed that a newer dataset helped make the IDS more relevant. Some literature also revealed the importance of performing feature scaling on the data.

From the literature studied it was found that the ANN was highly effective in intrusion detection but no literature was studied where it was used along with the highly effective

ANOVA feature selection technique. Also ,most of the literature used relatively older datasets. So a system is proposed here that performs intrusion detection and classification using an ANN which is trained using the best features selected by the ANOVA technique. The approach will also use a relatively new dataset the CSE-CIC-IDS 2022.

3. Research Methodology:

3.1 Overall Working:

The system that was developed was able to detect and classify security breaches or intrusion detection in networks as shown in Fig 3. The CSE-CIC-IDS 2022 was used here and the data in the dataset was first pre-processed, then the ANOVA feature selection technique was used for selecting the best features from the dataset. These pre-processed selected features were then balanced using the SMOTE technique and the features were scaled using the Standard scaling technique. Finally, the data was given as input to the ANN model which was created. The trained ANN model was able to perform intrusion detection and classification based on a set of network features given as input.

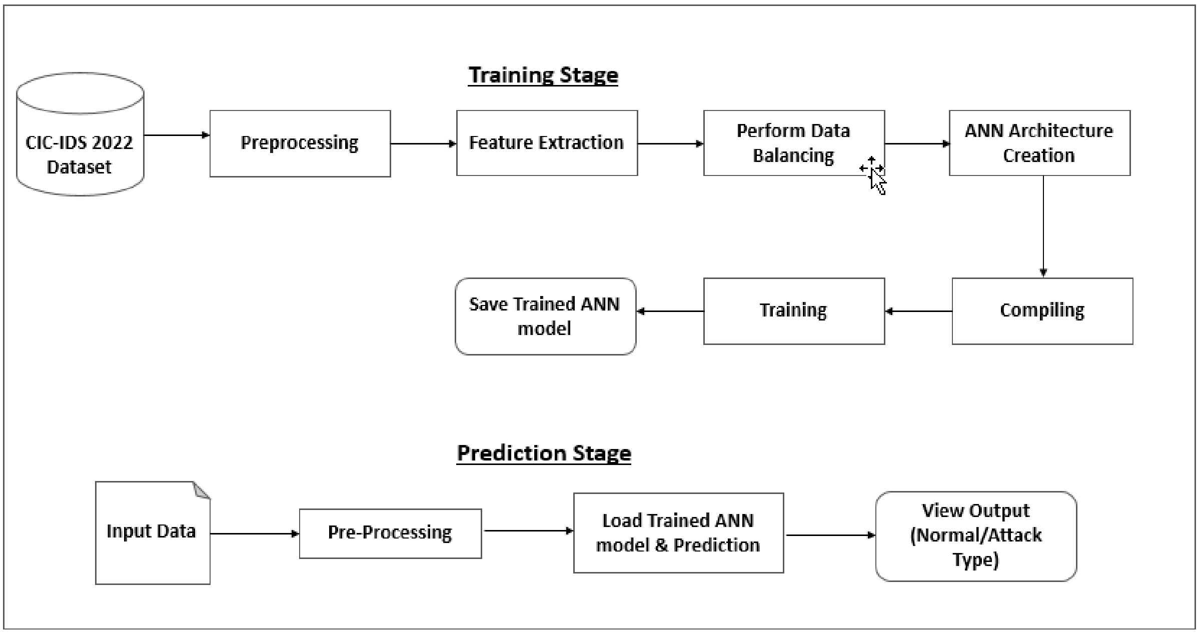


Fig 3. Proposed System Architecture

3.2 Dataset:

The CSE-CIC-IDS 2022 dataset consists of different attack scenarios in networks. It has data from networks that are normal and from networks in which data breaches have occurred. The dataset is in the form of a CSV file named ‘cic_ids_2022.csv’. In the file, the data about the different features associated with networks is represented as columns, and the data for each network is represented as rows. The dataset contains a column that represents the label or type of attack or class that has occurred on a network. The name of the column is ‘label’ and the values in this column are the names of the different attacks occurring in a network. This column also contains a value that represents a network in which no attack has taken place. The different kinds of attacks specified in ‘label’ are 'DDOS attack-LOIC-UDP', 'DoS attacks-Slowloris', 'SSH-Bruteforce' ,'DoS attacks-GoldenEye','FTP-BruteForce' , 'DDOS attack-HOIC' , 'DoS attacks-Hulk', 'DoS attacks-SlowHTTPTest' , 'Bot' and 'Infilteration'. The label also contains the value ‘Benign’ that represents a network in which no attack has occurred.

3.3 Pre-Processing the data:

The dataset's data included a sizable number of undesired and unpleasant values. These variables may have an impact on how well the deep learning model performs, thus the dataset is pre-processed and the undesired elements are eliminated. Firstly, the data from the dataset was loaded by loading the ‘cic_ids_2022.csv’ file. Firstly, the number of instances of the different attack classes was determined (Fig.4).

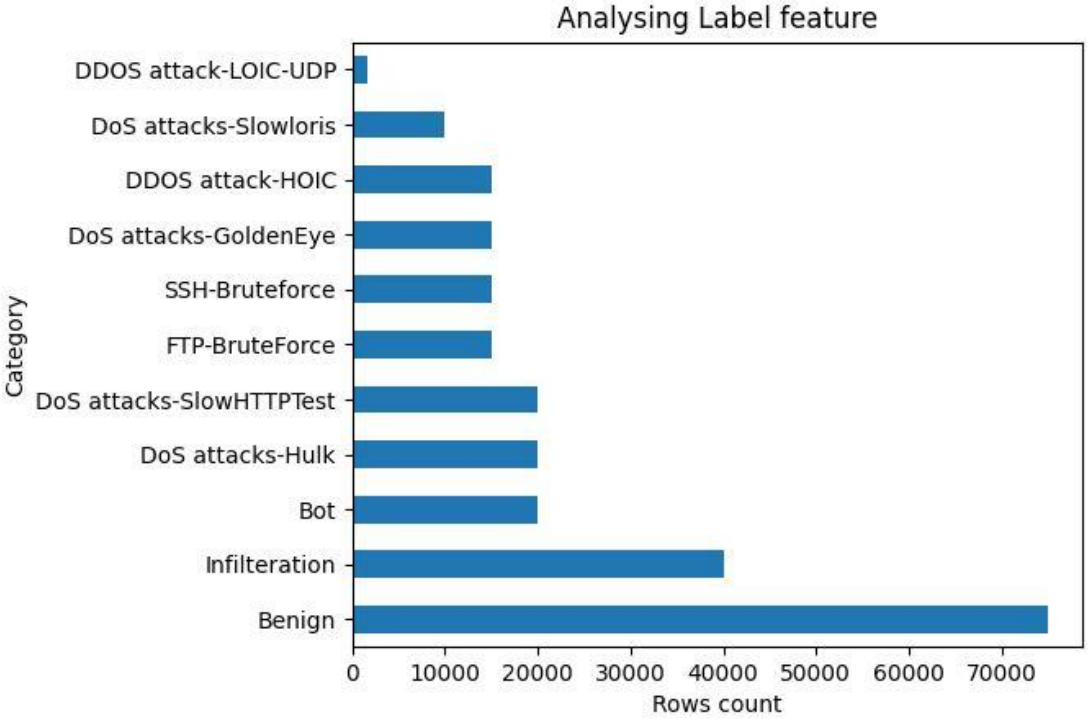
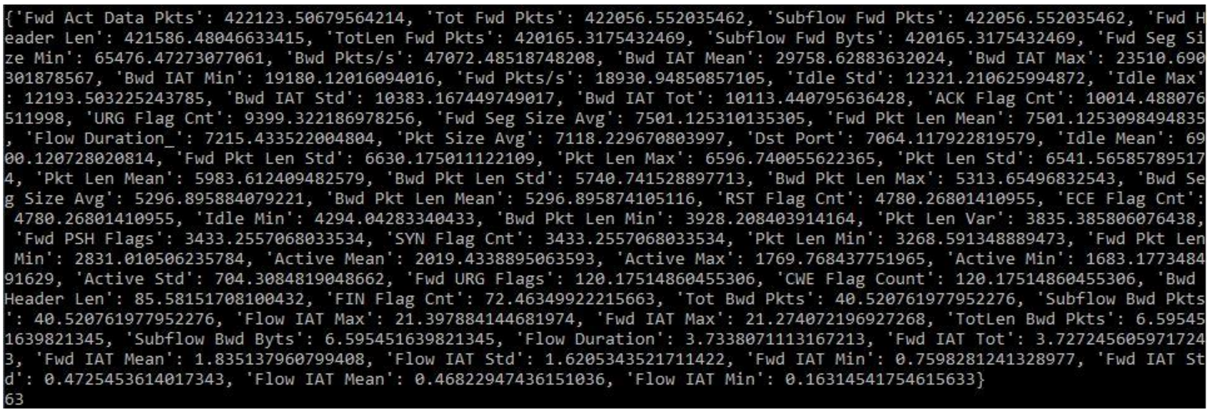


Fig 4. Categorizing the type of attack

On analyzing the data in the dataset it was found that the values in columns like 'Bwd PSH Flags', 'Bwd URG Flags', 'Fwd Byts/b Avg', 'Fwd Pkts/b Avg' , 'Fwd Blk Rate Avg' , 'Bwd Byts/b Avg' , 'Bwd Pkts/b Avg' ,'Bwd Blk Rate Avg' and 'Init Bwd Win Byts' were zero. These columns do not contribute anything to the learning process of the deep learning model so these columns were removed. Followed by this the data in all the remaining columns except ‘Label’ was stored in one variable and the data in ‘Label’ was stored in another variable.

3.4 Feature Selection:

The feature selection process, which comes before training the ANN model, is crucial since it aids in identifying the dataset's most crucial properties. The ANN model's efficiency was increased, and the training time was cut down by choosing the most crucial features. Using the ANOVA method, the features were chosen. The ANOVA technique was able to generate the feature score associated with every feature in the dataset, and from the feature scores, the best or most important features were determined. (Fig 5).



```
{'Fwd Act Data Pkts': 422123.50679564214, 'Tot Fwd Pkts': 422056.552035462, 'Subflow Fwd Pkts': 422056.552035462, 'Fwd H
eader Len': 421586.48046633415, 'TotLen Fwd Pkts': 420165.3175432469, 'Subflow Fwd Byts': 420165.3175432469, 'Fwd Seg Si
ze Min': 65476.47273077061, 'Bwd Pkts/s': 47072.48518748208, 'Bwd IAT Mean': 29758.62883632024, 'Bwd IAT Max': 23510.690
301878567, 'Bwd IAT Min': 19180.12016094016, 'Fwd Pkts/s': 18930.94850857105, 'Idle Std': 12321.210625994872, 'Idle Max'
: 12193.503225243785, 'Bwd IAT Std': 10383.167449749017, 'Bwd IAT Tot': 10113.440795636428, 'ACK Flag Cnt': 10014.488076
511998, 'URG Flag Cnt': 9399.322186978256, 'Fwd Seg Size Avg': 7501.125310135305, 'Fwd Pkt Len Mean': 7501.1253098494835
, 'Flow Duration_': 7215.433522004804, 'Pkt Size Avg': 7118.229670803997, 'Dst Port': 7064.117922819579, 'Idle Mean': 69
00.120728020814, 'Fwd Pkt Len Std': 6630.175011122109, 'Pkt Len Max': 6596.740055622365, 'Pkt Len Std': 6541.56585789517
4, 'Pkt Len Mean': 5983.612409482579, 'Bwd Pkt Len Std': 5740.741528897713, 'Bwd Pkt Len Max': 5313.65496832543, 'Bwd Se
g Size Avg': 5296.895884079221, 'Bwd Pkt Len Mean': 5296.895874105116, 'RST Flag Cnt': 4780.26801410955, 'ECE Flag Cnt':
 4780.26801410955, 'Idle Min': 4294.04283340433, 'Bwd Pkt Len Min': 3928.208403914164, 'Pkt Len Var': 3835.385806076438,
 'Fwd PSH Flags': 3433.2557068033534, 'SYN Flag Cnt': 3433.2557068033534, 'Pkt Len Min': 3268.591348889473, 'Fwd Pkt Len
 Min': 2831.010506235784, 'Active Mean': 2019.4338895063593, 'Active Max': 1769.768437751965, 'Active Min': 1683.1773484
91629, 'Active Std': 704.3084819048662, 'Fwd URG Flags': 120.17514860455306, 'CWE Flag Count': 120.17514860455306, 'Bwd
Header Len': 85.58151708100432, 'FIN Flag Cnt': 72.46349922215663, 'Tot Bwd Pkts': 40.520761977952276, 'Subflow Bwd Pkts
': 40.520761977952276, 'Flow IAT Max': 21.397884144681974, 'Fwd IAT Max': 21.274072196927268, 'TotLen Bwd Pkts': 6.59545
1639821345, 'Subflow Bwd Byts': 6.595451639821345, 'Flow Duration': 3.7338071113167213, 'Fwd IAT Tot': 3.727245605971724
3, 'Fwd IAT Mean': 1.835137960799408, 'Flow IAT Std': 1.6205343521711422, 'Fwd IAT Min': 0.7598281241328977, 'Fwd IAT St
d': 0.4725453614017343, 'Flow IAT Mean': 0.46822947436151036, 'Flow IAT Min': 0.16314541754615633}
63
```

Fig 5. Feature score associated with the features

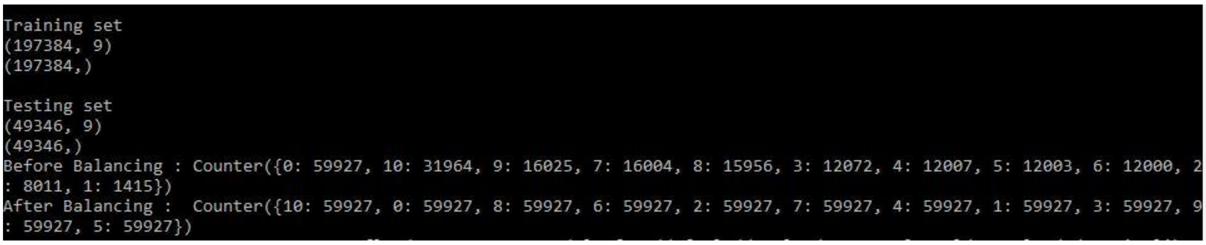
Ten features were selected to be the best and these were 'Dst Port', 'Flow Duration', 'Tot Fwd Pkts', 'Tot Bwd Pkts', 'Bwd IAT Mean', 'Fwd Act Data Pkts', 'Subflow Fwd Pkts', 'Bwd Pkts/s', 'Fwd Pkts/s' and 'Label'. A new data frame was created using the pre-processed data associated with the best 10 features and saved as the file 'final_dataset_preprocessed.csv'. The data in this data frame was used for training the ANN model.

The names of the different attack classes in the dataset were replaced with numerical values as deep learning models are able to handle numerical values more effectively as these models resemble machines. The labels were assigned numerical values in the following manner 'DDOS attack-LOIC-UDP' was assigned the value 1, 'DoS attacks-Slowloris' was assigned the value 2, 'SSH-Bruteforce' was assigned the value 3, 'DoS attacks-GoldenEye' was assigned the value 4,'FTP-BruteForce' was assigned the value 5, 'DDOS attack-HOIC' was assigned the value 6, 'DoS attacks-Hulk' was assigned the value 7, 'DoS attacks-SlowHTTPTest' was assigned the value 8, 'Bot' was assigned the value 9, 'Infilteration' was assigned the value 10 and 'Benign' was assigned the value 0.

3.5 Data balancing and featuring scaling:

3.5.1 Data balancing:

From Fig 4, it was evident that the number of instances of one of the attack classes in the dataset was highly greater than the other classes. This huge difference in the instances of the attack classes means that the data is imbalanced. The data was balanced because unbalanced data would cause the ANN model to perform worse because it would be more biassed towards the larger class with more members. The SMOTE method was used to balance the data. After balancing the numbers of all the attack classes, they were roughly the same as shown in Fig 6.



```
Training set
(197384, 9)
(197384,)

Testing set
(49346, 9)
(49346,)
Before Balancing : Counter({0: 59927, 10: 31964, 9: 16025, 7: 16004, 8: 15956, 3: 12072, 4: 12007, 5: 12003, 6: 12000, 2
: 8011, 1: 1415})
After Balancing :  Counter({10: 59927, 0: 59927, 8: 59927, 6: 59927, 2: 59927, 7: 59927, 4: 59927, 1: 59927, 3: 59927, 9
: 59927, 5: 59927})
```

Fig 6. Number of attacks before and after balancing

3.5.2 Feature Scaling:

Feature scaling was performed because the ranges of the values of the features were random, and some features had very high numerical values while other features had low numerical values. The data belonging to these random numerical ranges negatively affects the training of the ANN, as the features having smaller values might be considered insignificant by the ANN and ignored. The features were scaled using the Standard Scaler technique, and the scaling values used for scaling the data were saved in a file called "scaler.pkl'. This file was used during the prediction by the ANN model after training.

3.6 Training the ANN model:

A training set and a testing set were created out of the data used to train the ANN model. The data in the file "final dataset preprocessed.csv" made up 80% of the training set and 20% of the testing set. Then, utilizing 4 dense layers, 3 batch normalization layers, and 3 dropout layers, the ANN model was built. In these layers, the activation functions "relu" and "softmax" were employed. The final output layer of the ANN model was a dense layer that was made up of 11 neurons, as the ANN model had to detect and classify the intrusions into 11 different classes.

In order to compile the constructed ANN model, the "adam" optimizer and the loss function "categorical cross entropy" were used. The data in the training set was then used to train the ANN model. The features and labels were given separately as input to the ANN model, which was then trained and saved.

3.7 Evaluating the system:

The system that detected and classified data breaches contained the deep learning-based ANN model. As a result, the system's effectiveness was assessed by looking at the degree of precision with which it detected intrusions. The f1-score, precision, recall, and confusion matrix were chosen as additional performance criteria for this system's evaluation.

4. Design:

4.1 ANN model:

A vast number of small, densely connected processors make up ANNs, which can be thought of as a parallel and distributed processing system [26]. All neural network topologies that are a part of the ANN share the property that all neurons in a layer are connected to all neurons in all adjacent levels by unidirectional branches. The only direction in which the linkages and branches can broadcast information is forward. A feed-forward neural network is often trained using the back propagation algorithm. By utilizing the back propagation approach to train the network, a non-linear mapping between the input and output variables is created.

The artificial neurons present in one or more hidden layers of an ANN are given the inputs, which are then processed and weighted to determine the output to the next layer, as shown in Fig 7. The ANNs learn using the “learning rule,” which is often the gradient descent based back-propagation of errors. Using this rule, the set of biases and weights associated with the hidden layer and neurons in the output are tuned adaptively. Due to their ability to adapt on their own, ANNs may detect very complicated and non-linear correlations between independent and dependent variables without the need for prior knowledge [25].

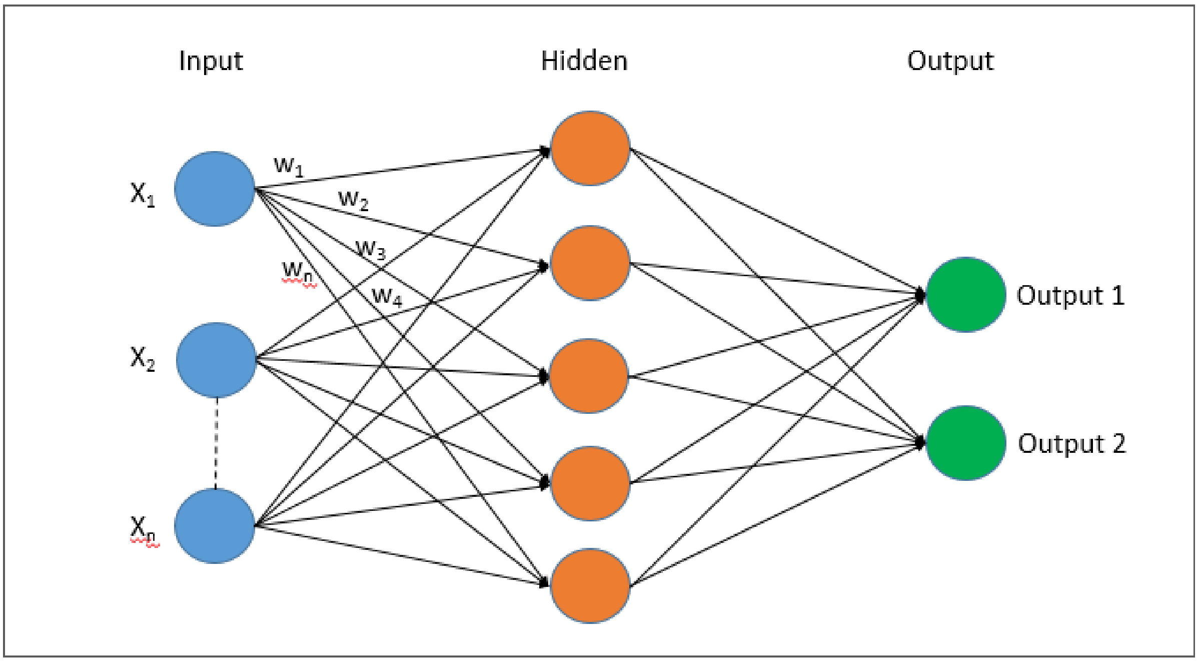


Fig 7. ANN model (Anita and Lawrence, 2019)

4.2 ANOVA feature selection:

ANOVA is a statistical method for comparing means that are independent [27]. By carrying out the comparisons, the ANOVA technique is able to help reduce the features. By computing the ratio of variances across and between groups, the ANOVA method ranks features [28]. ANOVA computes a score for every feature in the data that it is handling.

5. Implementation

5.1 Graphical User Interface:

The network security detection and classification system developed here was finally implemented in the form of a desktop application. The interface of this desktop application is able to receive the values of nine network features, as shown in Fig 8. On entering the values and clicking the button with the text 'Predict' these values are read, and a previously saved scaler value in the file 'scaler.pkl' is loaded for scaling the input values in the same way that the data was scaled during the training of the ANN. After scaling, the previously trained and saved ANN model is loaded, and the scaled values are given as the input to the ANN model. The ANN model is able to detect the class or type of intrusion associated with the values of the network features. The name of the class identified by the model will be displayed on the user interface.

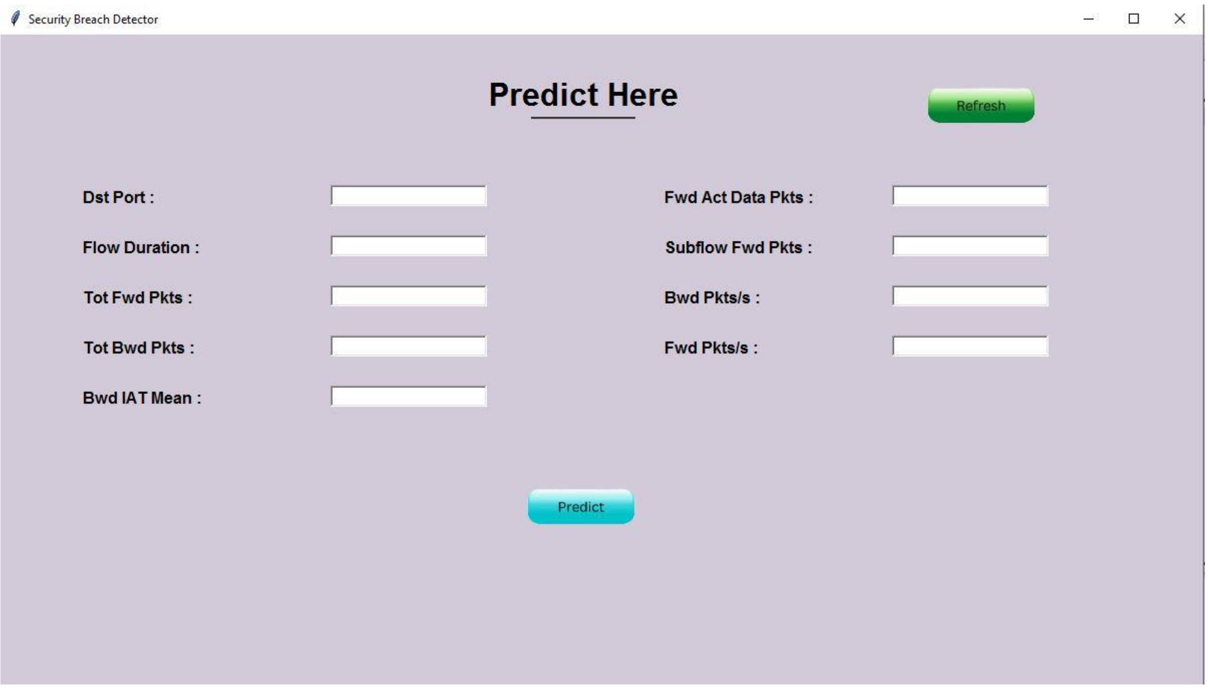
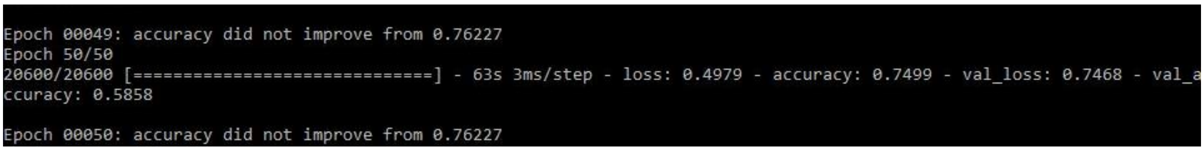


Fig 8. Desktop application user interface

6. Evaluation:

6.1 Accuracy

The accuracy of the system defines the ratio between the number of intrusions correctly classified by the ANN into the class to which they belong and the total number of classifications made by the ANN. The accuracy obtained was 76.22% as shown in Fig 9. The accuracy plot that visually represents the accuracy of the ANN was also generated, as shown in Fig 10.



```
Epoch 00049: accuracy did not improve from 0.76227
Epoch 50/50
20600/20600 [==============================] - 63s 3ms/step - loss: 0.4979 - accuracy: 0.7499 - val_loss: 0.7468 - val_a
ccuracy: 0.5858

Epoch 00050: accuracy did not improve from 0.76227
```

Fig 9. ANN accuracy

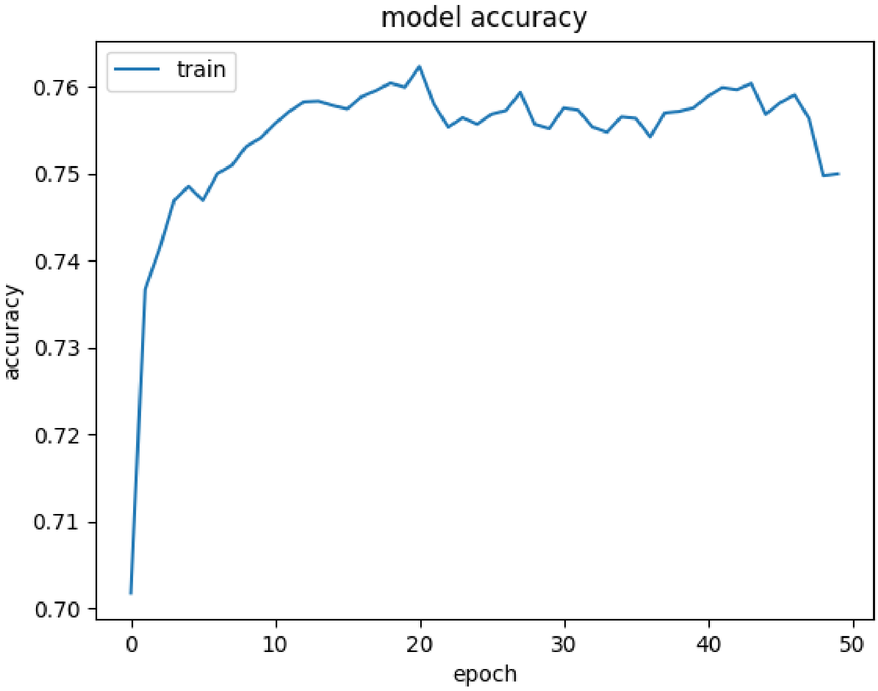
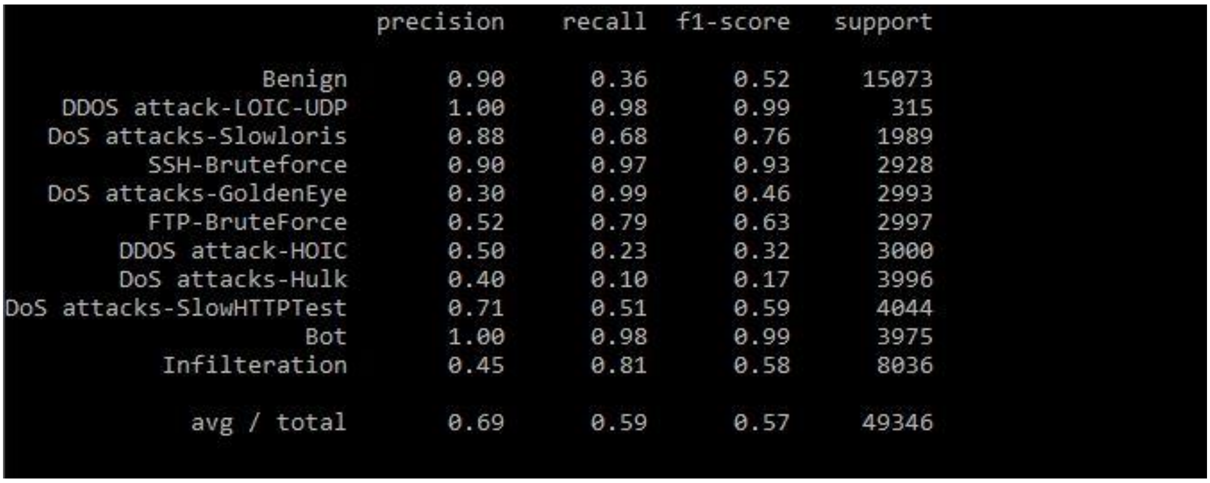


Fig 10. Visual representation of ANN accuracy

6.2 Precision, Recall and f1-score:

The precision is the ratio of the number of true positive classifications to the sum of the true positive and false positive classifications, as shown below. The recall is the ratio of the number of true positive classifications to the sum of the true positive and false negative classifications represented in the below figure. The f1-score is the harmonic mean of the recall and precision as shown in Fig 11. The precision, f1-score and recall achieved overall by the system and the precision, f1-score and recall associated with specific attack label or class is shown in figure(11).



```
                         precision    recall  f1-score   support

                 Benign       0.90      0.36      0.52     15073
  DDOS attack-LOIC-UDP       1.00      0.98      0.99       315
 DoS attacks-Slowloris       0.88      0.68      0.76      1989
        SSH-Bruteforce       0.90      0.97      0.93      2928
 DoS attacks-GoldenEye       0.30      0.99      0.46      2993
        FTP-BruteForce       0.52      0.79      0.63      2997
      DDOS attack-HOIC       0.50      0.23      0.32      3000
      DoS attacks-Hulk       0.40      0.10      0.17      3996
DoS attacks-SlowHTTPTest       0.71      0.51      0.59      4044
                    Bot       1.00      0.98      0.99      3975
          Infilteration       0.45      0.81      0.58      8036

            avg / total       0.69      0.59      0.57     49346
```

Fig 11. Precision, Recall and f1-score obtained

6.3 Confusion Matrix:

The confusion matrix is a visual representation of the classification performance of the trained ANN model. The performance of the ANN model in correctly predicting each of the 11 classes based on the network features is represented in the confusion matrix as shown in Fig 12.

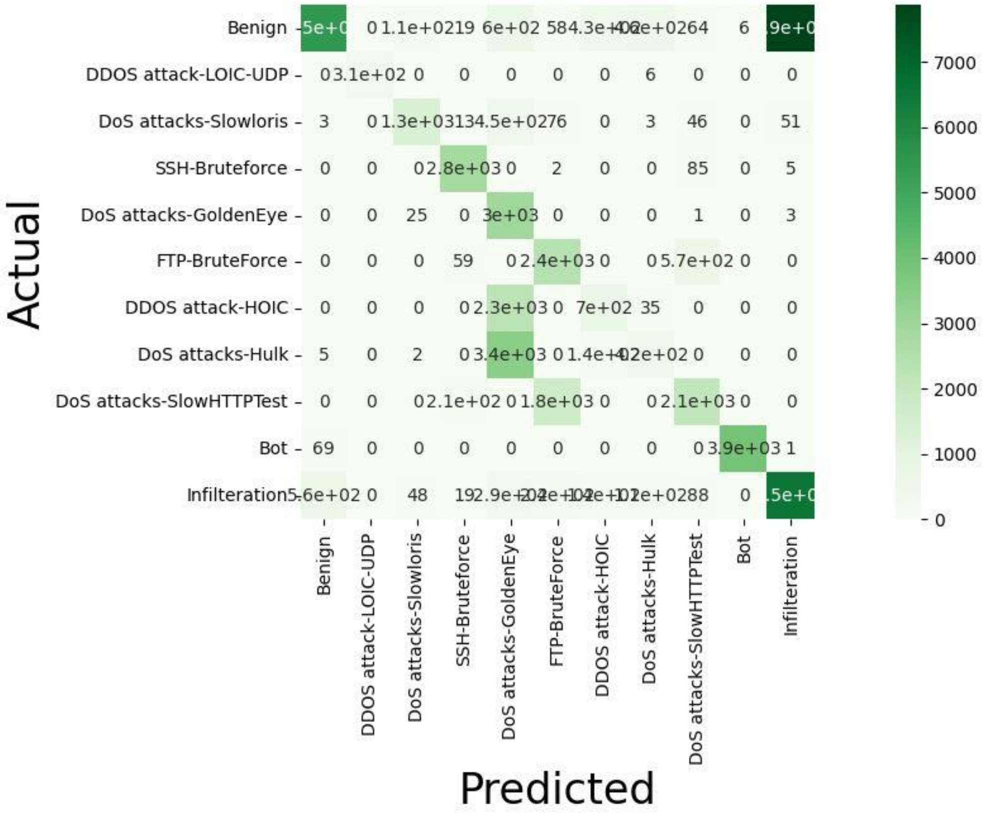


Fig 12. Confusion Matrix

6.4 Discussion:

A system that detects and classifies network breaches or intrusions was successfully created. The system was able to perform intrusion detection and classification rather effectively. It was observed from the existing literature that the deep learning models are able to effectively detect and classify intrusions; this observation is supported by the outputs of the system developed here, as the system is able to effectively detect and classify network intrusions. From the existing literature, it was also observed that feature selection methods made the performance of the deep learning models more effective; this observation was also supported by the findings made here, as feature selection enhanced the performance of the model. It was observed from the existing literature that the data associated with networks was imbalanced, and a balancing technique had to be used. This was in line with the development of the system proposed here, as the dataset was imbalanced and had to be balanced. The observation about the feature scaling from the literature study was also supported here, as feature evaluation was also performed here for improving the performance of the ANN model. The accuracy achieved by the system was 76.22%. From Table (1) and Table (2) it can be observed that this value of accuracy cannot be considered very high. This reduction in accuracy may be due to the system classifying the data into 11 different classes. But the exact reason for the reduction in accuracy could not be determined.

System	Accuracy(in %)
(Kasongo and Sun, 2020)	99.77%
(Ashiku and Dagli, 2021)	95.4%
(Maithem and Al-sultany, 2021)	99.98%
System proposed here	76.22%

Table(1): Accuracies of the existing systems based on deep learning that performed both detection, classification and the system developed here

System	Accuracy(in %)
(B.Singh and A.Anil Kr. , 2016)	97.97%
(Shenfield, Day and Ayesh, 2018)	98%
System proposed here	76.22%

Table (2): Accuracies of the existing systems that performed both detection using ANN and the system developed here.

The system is able to perform intrusion detection and classification effectively, but it has some limitations. The main limitation of the system is the low accuracy achieved by the system compared to the existing IDS. The reason for this could also not be conclusively determined.

7. Conclusion and Future work:

A system for detecting and classifying network breaches was successfully developed. The system was developed using a trained ANN model. The ANN was trained using the CSE-CIC-IDS 2018 dataset. The data was pre-processed, and the feature selection was performed using the ANOVA technique. The dataset was imbalanced, and it was balanced using the SMOTE technique. Feature scaling was also carried out using the Standard Scalar. Finally, the system was implemented in the form of a desktop application that could detect and classify network breaches based on the network features given as input. The system was able to achieve an accuracy of 76.22%. The main research questions were answered. The 1st research question, “Will an IDS developed using deep learning methods be able to detect and classify intrusions in a network?” was answered as a system that accurately detects and classifies intrusions in networks was developed using the deep learning-based ANN. The 2nd research question, " How will the performance of the deep learning based IDS in detecting and classifying intrusions be evaluated?" was answered as the performance of the IDS was evaluated using the performance metrics like accuracy, f1-score, precision, recall and the confusion matrix.

A different deep learning algorithm can be used to improve the performance of the system developed here. Along with a new deep learning algorithm, a new and more effective feature selection technique can also be used in the future to improve the performance of the system.

References

[1] L. Ashiku and C. Dagli, “Cybersecurity as a Centralized Directed System of Systems Using SoS Explorer as a Tool,” in *2019 14th Annual Conference System of Systems Engineering (SoSE)*, May 2019, pp. 140–145. doi: 10.1109/SYSOSE.2019.8753872.

[2] S. Duque and Mohd. N. bin Omar, “Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS),” *Procedia Comput. Sci.*, vol. 61, pp. 46–51, Jan. 2015, doi: 10.1016/j.procs.2015.09.145.

[3] S. Naeem, N. Jamil, H. U. Khan, and S. Nazir, “Complexity of Deep Convolutional Neural Networks in Mobile Computing,” *Complexity*, vol. 2020, p. e3853780, Sep. 2020, doi: 10.1155/2020/3853780.

[4] A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, “Deep Learning for Intrusion Detection and Security of Internet of Things (IoT): Current Analysis, Challenges, and Possible Solutions,” *Secur. Commun. Netw.*, vol. 2022, p. e4016073, Jul. 2022, doi: 10.1155/2022/4016073.

[5] L. Ashiku and C. Dagli, “Network Intrusion Detection System using Deep Learning,” *Procedia Comput. Sci.*, vol. 185, pp. 239–247, Jan. 2021, doi: 10.1016/j.procs.2021.05.025.

[6] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A Deep Learning Approach to Network Intrusion Detection,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[7] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A Critical Review of Recurrent Neural Networks for Sequence Learning.” arXiv, Oct. 17, 2015. doi: 10.48550/arXiv.1506.00019.

[8] A. Guezzaz, S. Benkirane, M. Azrour, and S. Khurram, “A Reliable Network Intrusion Detection Approach Using Decision Tree with Enhanced Data Quality,” *Secur. Commun. Netw.*, vol. 2021, p. e1230593, Aug. 2021, doi: 10.1155/2021/1230593.

[9] A. Karami, “An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities,” *Expert Syst. Appl.*, vol. 108, pp. 36–60, Oct. 2018, doi: 10.1016/j.eswa.2018.04.038.

[10] A. Yahyaoui, H. Lakhdhar, T. Abdellatif, and R. Attia, “Machine learning based network intrusion detection for data streaming IoT applications,” in *2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter)*, Jan. 2021, pp. 51–56. doi: 10.1109/SNPDWinter52325.2021.00019.

[11] Y. Kayode Saheed, A. Idris Abiodun, S. Misra, M. Kristiansen Holone, and R. Colomo-Palacios, “A machine learning-based intrusion detection for detecting internet of things network attacks,” *Alex. Eng. J.*, vol. 61, no. 12, pp. 9395–9409, Dec. 2022, doi: 10.1016/j.aej.2022.02.063.

[12] D. R. Patil and T. M. Pattewar, “Majority Voting and Feature Selection Based Network Intrusion Detection System,” *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 9, no. 6, Apr. 2022, Accessed: Mar. 08, 2023. [Online]. Available: https://eudl.eu/doi/10.4108/eai.4-4-2022.173780

[13] J. Fakirah, L. M. Zishan, R. Mooruth, M. L. Johnstone, and W. Yang, “A Low-Cost Machine Learning Based Network Intrusion Detection System With Data Privacy Preservation”.

[14] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, “A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks,” *IEEE Trans. Emerg. Top. Comput.*, vol. 7, no. 2, pp. 314–323, Apr. 2019, doi: 10.1109/TETC.2016.2633228.

[15] V. Hajisalem and S. Babaie, “A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection,” *Comput. Netw.*, vol. 136, pp. 37–50, May 2018, doi: 10.1016/j.comnet.2018.02.028.

[16] M. Siraj, “Analyzing ANOVA F-test and Sequential Feature Selection for Intrusion Detection Systems,” *Int. J. Adv. Soft Comput. Its Appl.*, vol. 14, no. 2, pp. 186–194, Aug. 2022, doi: 10.15849/IJASCA.220720.13.

[17] M. P. Bharati and S. Tamane, “NIDS-Network Intrusion Detection System Based on Deep and Machine Learning Frameworks with CICIDS2018 using Cloud Computing,” in *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, Oct. 2020, pp. 27–30. doi: 10.1109/ICSIDEMPC49020.2020.9299584.

[18] S. Moualla, K. Khorzom, and A. Jafar, “Improving the Performance of Machine Learning-Based Network Intrusion Detection Systems on the UNSW-NB15 Dataset,” *Comput. Intell. Neurosci.*, vol. 2021, p. e5557577, Jun. 2021, doi: 10.1155/2021/5557577.

[19] S. M. Kasongo and Y. Sun, “A deep learning method with wrapper based feature extraction for wireless intrusion detection system,” *Comput. Secur.*, vol. 92, p. 101752, May 2020, doi: 10.1016/j.cose.2020.101752.

[20] M. Maithem and G. A. Al-sultany, “Network intrusion detection system using deep neural networks,” *J. Phys. Conf. Ser.*, vol. 1804, no. 1, p. 012138, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012138.

[21] M. Naveed *et al.*, “A Deep Learning-Based Framework for Feature Extraction and Classification of Intrusion Detection in Networks,” *Wirel. Commun. Mob. Comput.*, vol. 2022, p. e2215852, Aug. 2022, doi: 10.1155/2022/2215852.

[22] B. I. Farhan and A. D. Jasim, “Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 2, Art. no. 2, May 2022, doi: 10.11591/ijeecs.v26.i2.pp1165-1172.

[23] R. I. Farhan, A. T. Maolood, and N. F. Hassan, “Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, Art. no. 3, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1413-1418.

[24] B. Singh and A. K. Ahlawat, “Innovative Empirical Approach for Intrusion Detection Using ANN.” Rochester, NY, 2016. Accessed: Mar. 08, 2023. [Online]. Available: https://papers.ssrn.com/abstract=3535008

[25] A. Shenfield, D. Day, and A. Ayesh, “Intelligent intrusion detection systems using artificial neural networks,” *ICT Express*, vol. 4, no. 2, pp. 95–99, Jun. 2018, doi: 10.1016/j.icte.2018.04.003.

[26] D. Devaraj, J. P. Roselyn, and R. U. Rani, “Artificial neural network model for voltage security based contingency ranking,” *Appl. Soft Comput.*, vol. 7, no. 3, pp. 722–727, Jun. 2007, doi: 10.1016/j.asoc.2005.11.010.

[27] H. Nasiri and S. A. Alavi, “A Novel Framework Based on Deep Learning and ANOVA Feature Selection Method for Diagnosis of COVID-19 Cases from Chest X-Ray Images,” *Comput. Intell. Neurosci.*, vol. 2022, p. e4694567, Jan. 2022, doi: 10.1155/2022/4694567.

[28] H. Lin and H. Ding, “Predicting ion channels and their types by the dipeptide mode of pseudo amino acid composition,” *J. Theor. Biol.*, vol. 269, no. 1, pp. 64–69, Jan. 2011, doi: 10.1016/j.jtbi.2010.10.019.