Library Progress International
Vol.44 No. 3, Jul-Dec 2024: P. 15925-15935

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

An Advanced design of Intrusion Detection System using Machine learning Architecture

Abhishek Gandhar [1], Prakhar Priyadarshi [2]*, Shashi Gandhar[3], S B kumar [4] Arvind Rehalia [5], Mohit Tiwari [6]

[1]Professor, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[2]Professor, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[3]Associate Professor, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[4]Associate Professor, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[5] Associate Professor, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[6]Assistant Professor, Bharati Vidyapeeth's College of Engineering, New Delhi, India
***Corresponding author email:** prakharpriya@gmail.com

How to cite this article: Abhishek Gandhar, Prakhar Priyadarshi, Shashi Gandhar, S B kumar Arvind Rehalia , Mohit Tiwari (2024) An Advanced design of Intrusion Detection System using Machine learning Architecture. *Library Progress International,* 44(3), 15925-15935

Abstract:
The Internet is advancing over the years, computer networks and its applications are having an exponential growth. An increased chances of getting attacked and major risk of potential damage caused by it, therefore many Intrusion Detection Systems(IDS) and Intrusion Prevention Systems (IPS) are instantiated to analyze networks andprevent any threats. Due to the limited number of datasets, such intrusion detection systems cannot be deployed accurately as well as are providing full proof solutions. In this paper, a comprehensive evaluation regarding the network features and outline the issue pertained by CICIDS Datasets is presented. This paper also presents an advanced design of Intrusion Detection System which can be used to classify any given network flow and declare Malicious or Benign system. The proposed system can evaluate different real-world models using deep learning techniques resulting in better responses
Keywords: CICIDS2017, CICIDS2018, Deep Learning, Feature Selection, Network Features.

1. Introduction:

With the exponential growth of information and communication technology, a significant challenge to provide security and prevention arises for today's network engineers and data researchers. Intrusion Detection Systems are important to defend networks and prevent attacks. It is one of the dynamic parts of the network which monitors large amounts of information and analyzes it by searching for any abnormality or deviation in network policies to prevent any attacks or system failure. The correct meaning of intrusion is illegal to access or an attempt to access a protected environment. Various IDS models have high accuracy and low false rate, but a few are ready for production use. Therefore, the shortcomings of the presented dataset i.e., redundant data and imbalance dataset can further cause irregularities have been removed. Thus the aim of this paper to build an effective IDS model which can be ready for production.

For an IDS to work efficiently, it must be trained with a proper dataset, but many of the dataset available have certain dependencies which cause a greater amount of problem to most of the classification and thus results in inaccurate models to predict outcome.

The architecture of the manuscript consists of following sections: Section 2 presented a brief literature survey, Section 3 shown the detailed analysis of the CICIDS 2017 and CICIDS 2018 dataset followed by corrections, Section 4 presented the implementation of IDS using proposed dataset, Section 5 provided the different evaluation techniques for improving results followed by Section 6 exhibited the conclusion.

2. Literature Survey:

A data breach could happen to any organization today. As information is stored digitally, someone can retrieve sensitive data if it is not secured. Due to the pandemic, cyber security incidents have increased. There have been more than 445 million cyber-attacks reported, which is double when compared to 2019.The Kaspersky Security Network (KSN) reported detecting and blocking 52 million local cyber threats in India this year[1]. Significant risks include Information leaks, Phishing attacks etc

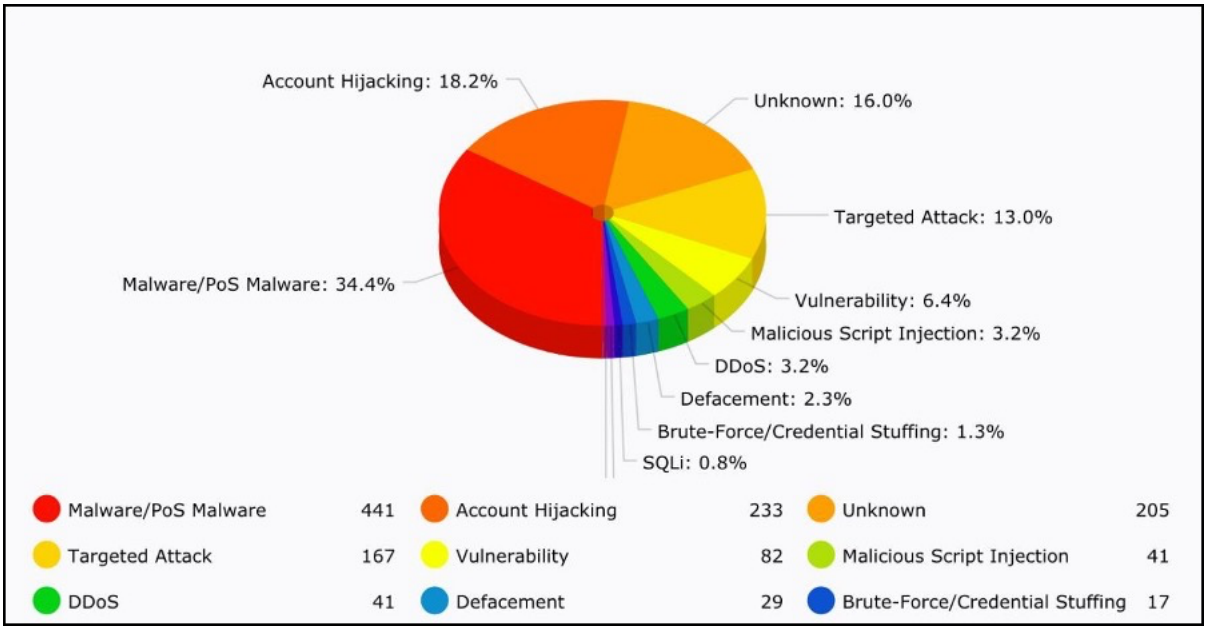
The entropy-based IDS counter-measure methods do not rely on the particular attack features. The main implementation is to detect various attacks by grasping the network traffic updates. An intrusion detection algorithm does the pre-processing on network packets and can identify a contingency reasonable by trustworthy websites or any other type of cyber-attacks. A hybrid system that joins anomaly detection with a signature-based IDS in a series manner achieves almost double the IDS-only system's detection accuracy. The proposed system will provide a solution using error detection and signature-based identification qualities simultaneously [2]. This methodology increased the performance of the offered algorithm, by using the exponential smoothing machine as a forecasting architecture. [3].

Numerous application fields and a broad range of categorization problems have made use of deep learning. Deep Learning is a "black-box" technique that can adapt to the underlying system model, unlike traditional classification methods like Neural Networks and discriminant analysis, which demand a thorough understanding of the underlying assumptions of the probability model of the system that produced the data. Their capability to modify the data, especially in high dimensional datasets, makes them especially useful in fields like decision support for hidden weapon identification, anticipating and categorizing of Internet traffic, and signature verification, where they overcome many of the challenges in model building associated with traditional classification techniques and algorithms[4].

There is a rapid increment in web attacks globally , In last five years, a sharp rise in phishing attacks are recorded year by year, which summed to a loss of billions of capital. Therefore, a foolproof system which must be error-resistant. The accuracy of the model in "A Hybrid Network Intrusion Detection Model Based on CNN-LSTM and Attention Mechanism" was 99.41%. The model was tested using the imbalanced CIRA-CIC-DoHBrw-2020 dataset. Modules ACNNBN and LSTM make up the model[5]. A hybrid intrusion detection system (IDS) is developed in "A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs" by combining the convolutional neural network (CNN) and long short-term memory network (LSTM). This method enhances the intrusion detection performance of zero-day attacks and integrates the CNN with LSTM to improve intrusion detection performance and achieve an accuracy of 96.32%. The calculated accuracy surpasses the accuracy of each individual model[6].

Authors demonstrate exceptional proficiency in network flow analysis in "HYBRID-CNN: An Efficient Scheme for Abnormal Flow Detection in the SDN-Based Networks," where they apply Long Short-Term Memory (LSTM). These techniques, however, are inaccurate since they are unable to extract the deep features from network traffic. We suggest using a Hybrid Convolutional Neural Network (HYBRID-CNN) technique to solve the aforementioned issues. To be more precise, the HYBRID-CNN uses a CNN to generalize local features using two-dimensional (2D) data and a Deep Neural Network (DNN) to efficiently memorize global features using one-dimensional (1D) data[7].Compared to the accuracy of each individual model, the estimated accuracy is higher[6].

The CICIDS2017 intrusion detection dataset, which is up to date and covers all frequent intrusions and cyberattacks, is used to analyze DL-IDS in the paper "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system" [2]. DL-IDS achieved an overall accuracy of 98.67% in the multi-classification test, with an accuracy of over 99.50% for each type of attack[7].
Lastly, CNN is a supervised deep-learning approach among other neural network types in "A Survey on Network Intrusion Detection using Convolutional Neural Network" [3]. R. Upadhyay and D. Pantiukhin used it for the first time in intrusion detection in 2017[8–9].



Details of different attacks occurred with their distribution.

Figure 1. Attack Distribution in 2018[10-11].

3. Analysis of CICIDS 2017/2018 Datasets:

The dataset of CICIDS contains 5000000 instances and 84 features. It has 14 attack labels like Brute - force, DDoS, Heartbleed, etc. and 1 benign label which resembles the true - real world data. On examining the dataset, 72000 instances of missing values were found, which after removing, 4928000 instances were left in the dataset. The characteristics of CICIDS 2017/2018 dataset and count of labels is present in Table 1 and Table 2, respectively.

Table 1: Features of CICIDS 2017/2018 dataset

Name	CICIDS 2017/2018
Dataset Category	Multi Class
Year of Launch	2017/2018
Available different instances	4928000
Available features	84
Available different classes	15

Table 2: Count of labels in CICIDS 2017/2018 dataset

Class Labels	**Number of instances**
Benign	22,000,000
DOS Hulk	230124
Port Scan	158930
DDOS	128027

FTP - BruteForce	201292
SSH - BruteForce	193486
DOS Slowloris	16786
DOS Slowhttptest	5499
Botnet	288157
Web Attack - Brute Force	2118
Web Attack - XSS	882
Infiltration	161132
Web Attack - SQL injection	108
Heartbleed	11
DOS GoldenEye	51801

Shortcomings of CICIDS 2017/2018 dataset:

- **Missing Values:** It has been noted that the CICIDS dataset comprises 72000 instances with missing counts which need to be removed from the dataset.
- **Huge Volume of Data:** The combined dataset of 2017 and 2018 becomes so huge such that it requires more resources such RAM & CPU for data loading and processing it for classification.
- **Dispersed Data:** CICIDS data is present across numerous files which makes the processing of files a difficult task.
- **Class variation:** The CICIDS dataset contains improper distribution of classes which makes the data imbalance.

Changes in CICIDS dataset:

To minimize the inaccuracies caused by the dataset, we performed following operations on the dataset:

1. Merge Dataset:

To reduce the file processing and loading time of the dataset, we merged all the different files of the into single CSV file to ease the workflow and increase the efficiency.

2. Clean the Dataset:

To further reduce the space complexity of the dataset, we dropped all the rows containing NaN values and further reduce the storage need by editing the datatype of the column required by adjusting to the needs.

3. Balance the Dataset:

After combining the two datasets.i.e. (CICIDS 2017 and CICIDS 2018), we have a dataset with more than 40 million rows and over 80 columns for features. This is a huge disadvantage for building a Binary Classification Model. Therefore, the requirement to balance the dataset is vital. As a result, 180000 Benign Entries and 150000 Malicious Entries are made into a dataset to balance the classification. The attack labels of CICIDS dataset were combined into a single class i.e. (Malicious) to do a binary classification.

4. Removing Invalid Entries:

Entries with incorrect values are removed and some values of the dataset are reaching infinity both positive and negative which are also removed.

5. Implementation:

For the implementation, A binary classification model is designed to check the flow of the network, whether is it benign or malicious. Following different Machine Learning Models are evaluated to get efficient results.

1. Logistic Regression.
2. K-Nearest Neighbors.
3. Random Forest Classifier.
4. Decision Tree Classifier.

These are the best binary classification model in Sklearn Machine Learning Library in Python-3.7. We have taken a part of the original dataset with around 500000 samples and categorized them in two categories i.e., Benign & Malicious. The implementation of the model is presented in Fig. 2

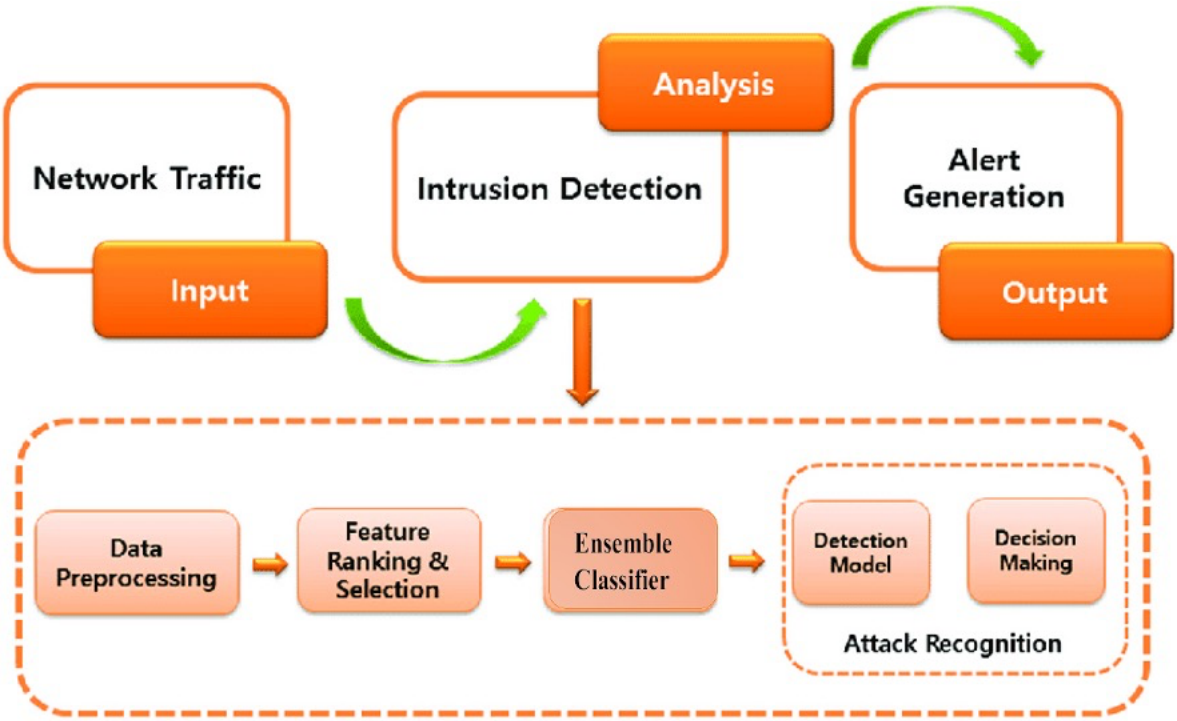


Figure 2. Architecture of the proposed model

Customary, the first step is to clean the dataset. So, in the first step all row containing null values for all attributes, removed the outliers are removed and then rows containing the infinite values from our dataset are also removed .After that proceed for typos, of the categorical label values and other parameters like connection type, etc. In the next step, combine all the subcategories of an attack into a Single major attack type, as considering the abstract model formation. In the next set of implementations, transformation operations in the dataset is performed, so first, the code is assigned to each category label using Pandas Categories, as function will not move with string manipulations. Next, will give appropriate data types to some attributes, like “Timestamp”. Then reached to dataset information gathering, where it has been found that 60% records are of Benign and the left 40% is of Malware which further divides in several categories. Also, the different set of values for an attribute, like their min, max, mean, etc. are examined thoroughly , So an overview of range of a parameter is achieved, then move forward for attribute correlations, but must keep an eye on the attributes which has high correlations, plotted various correlations graph. Now the data splitting will performed for inputs and outputs, like the label as our target is set, and other attributes which can be a major factor to determine the label as variables. Distribution of the label is given in Figure 3.

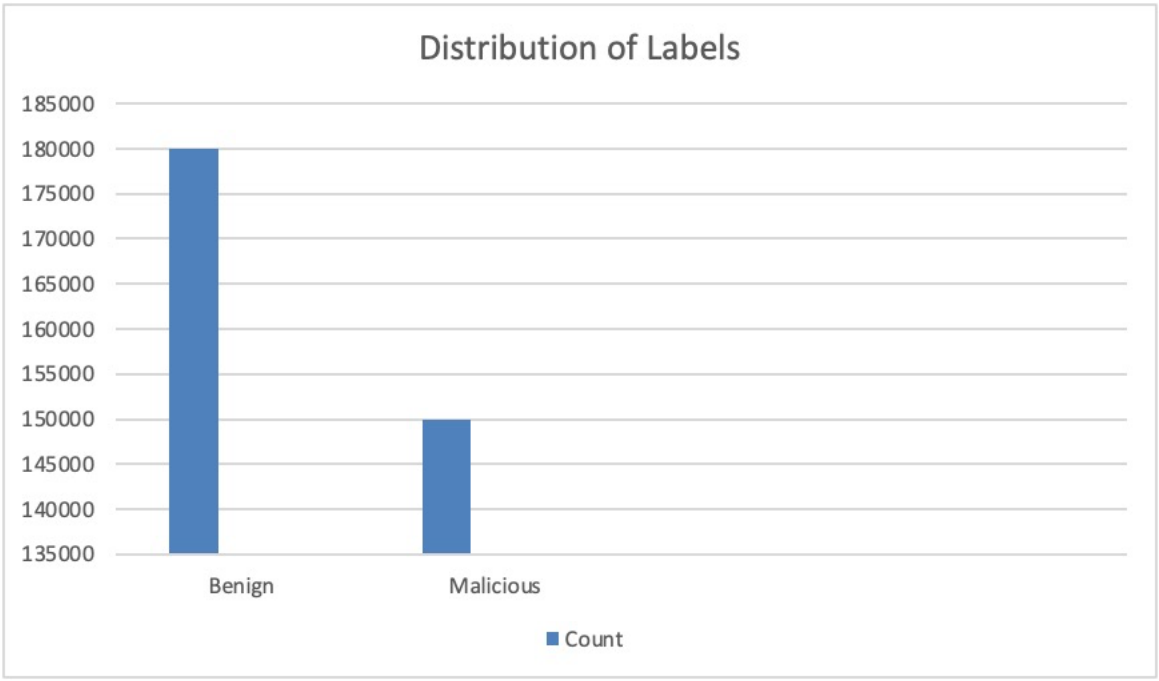


Figure 3. Distribution of Labels.

Now some attributes like timestamp, connection type, src ip, dst, ip etc. are removed as they are negotiable correlations with the target variable. Then the data is segregated for training and testing module, the ratio of 40:60 is considered for training and testing, respectively. Then move for data fitting on different models and different results are achieved. After combining all the predicted value, a model is designed to conclude the final label of the presented details.

Feature Selection and Scaling:

In this section the dataset is manipulated to get an optimized result on the proposed model. Therefore, the dataset is scaled using Standard Scaler, which reduces the processing and storage time significantly.

For the Feature Selection process, it has been sorted and collected the top 45 features from Logistic Regression feature_importance function, Decision Tree Classifier feature_importance function and Random Forest Classifier feature_importance function. After gathering all the feature, a dataset comprising of 27 features is formulated. This selection process reduced the overall processing time and storage required significantly and thus resulting in an optimized model which will be more efficient than previous one. Details of the selected features are shown in Table 3.

Table 3: Selected Feature for Final Evaluation [12].

Initial Window Bytes Forward	No. of bytes sent in initial window in the Forward (fwd) direction
Active Maximum	Max. time a flow was active before becoming idle
Flow IAT Mean	Avg. time between two flows
Subflow fwd Bytes	Describe mean of bytes in a sub-flow in the fwd path
Flow IAT Std	Std. deviation time two flows
RST(Reset) Flag Count	No. of Packets (Pkts) with RST
Sub-flow Backward(bwd) Pkts	Avg. no. of Pkts in a sub flow in the bwd direction
Backward Header Length	Total bytes used for headers in the fwd direction

ECE (ECN echo) Flag Count	No. of Pkts with ECE
Initial Window Bytes bwd	No. of bytes sent in initial window in the bwd direction
Backward Packet LengthStandard	Std. deviation size of Pkt in bwd direction
Forward Packet LengthMinimum	Avg. size of Pkt in fwd direction
Packet Length Variance	Min. inter-arrival time of Pkt
Total Backward Packets	Total Pkts in the bwd direction
Backward Packets/s	No. of bwd Pkts / second
Active Minimum	Min. time a flow was active before becoming idle
Average Packet Size	Avg. size of Pkt
Flow Packets/s	Flow packets rate that is no. of Pkts transferred / second
Forward IAT Max	Max. time between two Pkts sent in the fwd direction
Forward IAT Std	ation time between two Pkts sent in the fwddirection
Forward Header Length	Total bytes used for headers in the fwd direction
Forward IAT Total	Total time between two Pkts sent in the fwd direction

6. Evaluation:

Machine Learning Model:

Before Feature Selection and Scaling:

In this section, a comparative analysis of the different models is discussed. It can be specifically inspected that without any parameter specification, the model is producing a quite high output , but some models are under performing like Logistic Regression.

Table 4: Evaluation of Binary Classification Model With default parameters

	Model	Fitting time	Scoring time	Accuracy	Precision	Recall	F1_score	AUC_ROC
2	Random Forest	53.107196	1.957395	0.995646	0.995417	0.995813	0.995647	0.999688
1	Decision Tree	8.944193	0.104310	0.995215	0.994928	0.995442	0.995216	0.997695
3	K-Nearest Neighbors	4.591267	29.934623	0.970444	0.970361	0.970008	0.970439	0.988622
0	Logistic Regression	6.792112	0.117802	0.784499	0.792460	0.774780	0.781167	0.821156

Further a Voting Classifier Ensemble model is designed, which comprises of all the mentioned model (Logistic Regression, k-NN, Decision Tree Classifier, Random Forest Classifier) and considered its soft and hard voting classification. Evolution details are given in Table 5.

Table 5: Evaluation of Ensemble Model with default parameter

	Model	Accuracy	Precision	Recall	F1_score	AUC_ROC
1	Ensembling_soft	0.994906	0.991877	0.996941	0.994403	0.995078
0	Ensebling_hard	0.984497	0.992026	0.973673	0.982764	not applicable

After Feature Selection and Scaling:

After the feature selection and scaling, an increase in overall metrics is observed, which further entails the proposed evaluation that the CICIDS dataset can be optimized for getting much higher efficiency. Details of the Evaluation Metrics are given in Table 6.

Table 6: Evaluation of Different Models after dataset manipulations

	Model	Fitting time	Scoring time	Accuracy	Precision	Recall	F1_score	AUC_ROC
2	Random Forest	35.854499	1.325677	0.995917	0.995681	0.996096	0.995918	0.999676
1	Decision Tree	2.876061	0.093805	0.995203	0.994920	0.995424	0.995204	0.997683
3	K-Nearest Neighbors	16.167837	36.496153	0.990643	0.990488	0.990644	0.990644	0.996611
0	Logistic Regression	4.286083	0.126241	0.852816	0.856336	0.847323	0.851875	0.935387

From the previous, Voting Classifier Ensemble model comprising of all the above model (Logistic Regression, k-NN, Decision Tree Classifier, Random Forest Classifier) and considered its soft and hard voting classification. The model using the edited dataset is evaluated. Evaluation details are given in Table 7.

Table 7: Evaluation of Ensemble Model after dataset manipulation

	Model	Accuracy	Precision	Recall	F1_score	AUC_ROC
1	Ensembling_soft	0.995471	0.992383	0.997679	0.995024	0.995657
0	Ensebling_hard	0.993489	0.992640	0.993017	0.992829	not applicable

Deep Learning Model:

The proposed sequential model in keras/tensorflows, a feedforward ANN without numerous inputs and/or outputs, defining layers one by one. An input layer, that is, the layer that gets input from outside sources. The thick or completely connected layers are considered for classification. The rationale is that, despite the best efforts to pool and convolute the input, it remains multidimensional and requires linearity in order to be sent through a densely connected layer. The continuity is applied via imposing constraints either on the ANN's parameters or modifying the cost function. The application of dropout layers that randomly and temporally set nodes in proposed layers to 0, i.e., deleting them during the training. To train the ANN functioning between various network attacks. Thus, the final layer, will be a densely connected layer again, which has equal number of outputs and classes. Additionally, the activation function is modified to indicate the probability of relations to either class. Then it will be compared to the true labels.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 72, 128)	896
activation_1 (Activation)	(None, 72, 128)	0
conv1d_1 (Conv1D)	(None, 67, 256)	196864
activation_2 (Activation)	(None, 67, 256)	0
flatten (Flatten)	(None, 17152)	0
dense_4 (Dense)	(None, 256)	4391168
dropout_1 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 15)	3855
activation_3 (Activation)	(None, 15)	0

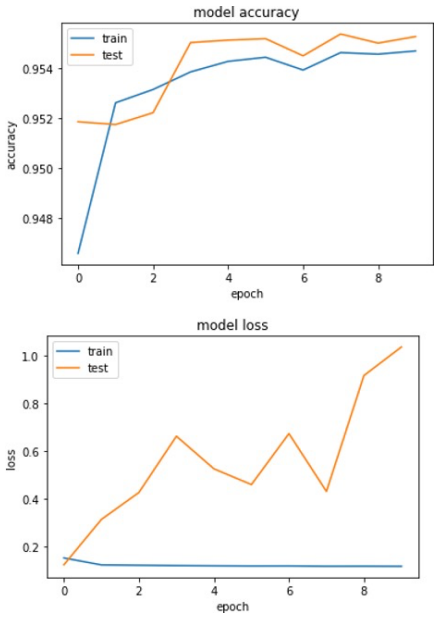
Total params: 4,592,783
Trainable params: 4,592,783
Non-trainable params: 0

Fig 7. Classifier Training for Model 1

Results:

```
Epoch 1/10
7181/7181 [==============================] - 58s 8ms/step - loss: 0.1539 - accuracy: 0.9466 - val_loss: 0.1254 - val_accuracy: 0.9519
Epoch 2/10
7181/7181 [==============================] - 61s 8ms/step - loss: 0.1248 - accuracy: 0.9526 - val_loss: 0.3151 - val_accuracy: 0.9517
Epoch 3/10
7181/7181 [==============================] - 57s 8ms/step - loss: 0.1233 - accuracy: 0.9531 - val_loss: 0.4280 - val_accuracy: 0.9522
Epoch 4/10
7181/7181 [==============================] - 56s 8ms/step - loss: 0.1219 - accuracy: 0.9539 - val_loss: 0.6634 - val_accuracy: 0.9550
Epoch 5/10
7181/7181 [==============================] - 57s 8ms/step - loss: 0.1210 - accuracy: 0.9543 - val_loss: 0.5269 - val_accuracy: 0.9551
Epoch 6/10
7181/7181 [==============================] - 56s 8ms/step - loss: 0.1203 - accuracy: 0.9544 - val_loss: 0.4610 - val_accuracy: 0.9552
Epoch 7/10
7181/7181 [==============================] - 56s 8ms/step - loss: 0.1204 - accuracy: 0.9539 - val_loss: 0.6744 - val_accuracy: 0.9545
Epoch 8/10
7181/7181 [==============================] - 56s 8ms/step - loss: 0.1192 - accuracy: 0.9546 - val_loss: 0.4324 - val_accuracy: 0.9554
Epoch 9/10
7181/7181 [==============================] - 56s 8ms/step - loss: 0.1195 - accuracy: 0.9546 - val_loss: 0.9175 - val_accuracy: 0.9550
Epoch 10/10
7181/7181 [==============================] - 56s 8ms/step - loss: 0.1190 - accuracy: 0.9547 - val_loss: 1.0370 - val_accuracy: 0.9553
Best loss: 0.12535648047924042
Model: CNN-Model1.h5
Balanced Acc loss: 78.40240998013573
{'loss': 0.12535648047924042, 'status': 'ok', 'model_name': 'CNN-Model1.h5'}
```

Epoch Report for Model



Graph for Epochs for Model

7. Conclusion:

A detailed analysis of CICIDS 2017 and CICIDS 2018 dataset with its shortcomings and their removal is presented in this research article. A comparative analysis of different dataset that are present for the training of IDS model and their shortcomings were also discussed. The manuscript also presented a error resistant solution to the real word issue implementing the nearly perfect IDS. The proposed model exhibited the peck of efficiency that can be achieved from this evaluation technique and resulted in using the Ensemble Model with soft voting parameter and got a 99.5024% of F1-Score.

Such IDS can be utilized for checking the file system for adulterations. These IDS are able to detect the modifications in the files subject to cyber-attacks. The proposed IDS can protect the machines from various types of security threats. Therefore the presented IDS can boost up the security in today's computer world.

Constraints

System Memory might be limited due to very large dataset input, with around 50 million rows and 84 features.

Parsing raw data to get the feature may create a bottleneck for preprocessing and predictions.

References:

1. I. Dutt, S. Borah, and I. K. Maitra, "Immune System Based Intrusion Detection System (IS-IDS): A Proposed Model," in IEEE Access, vol. 8, pp. 34929-34941, 2020, doi: 10.1109/ACCESS.2020.2973608.
2. Özge Cepheli, Saliha Büyükçorak, Güneş Karabulut Kurt,, "Hybrid Intrusion Detection System for DDoS Attacks" Journal of Electric and Computer Engineering, 03 April 2016, https://doi.org/10.1155/2016/1075648
3. K. Pradeep Mohan Kumar, M. Saravanan, M. Thenmozhi, K. Vijayakumar, "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks" Journal of Concurrency and Computation Practice and Experience,13 March 2019,https://doi.org/10.1002/cpe.5242
4. Alex Shenfield, David Day, Aladdin Ayesh, "Intelligent intrusion detection systems using artificial neural networks",ICT Express,Volume 4, Issue 2,2018,Pages 95-99,ISSN 2405-9595, https://doi.org/10.1016/j.icte.2018.04.003.
5. Sasanka Potluri, Shamim Ahmed,Christian Diedrich,"Convolutional Neural Networks for Multi-class Intrusion Detection System", Proceedings of 6th International Conference, MIKE 2018, Cluj-Napoca, Romania, December 20–22, December 2018 DOI: 10.1007/978-3-030-05918-7_20
6. Stephen Kahara Wanjau,Geoffrey Wambugu,Aaron Mogeni Oirere,Geoffrey Muchiri Muketha,"Discriminative spatial-temporal feature learning for modeling network intrusion detection systems",Journal of Computer Security

32(322):1-30, February 2023,DOI:10.3233/JCS-220031

7. R. Upadhyay and D. V. Pantiukhin, “Application of Convolutional neural networks to intrusion type recognition,” 2017

8. W. Zhong, N. Yu and C. Ai, "Applying big data based deep learning system to intrusion detection," in Big Data Mining and Analytics, vol. 3, no. 3, pp. 181-195, Sept. 2020, doi: 10.26599/BDMA.2020.9020003.

9. Antanios Kaissar, Ali Bou Nassif and MohammadNoor Injadat, “A Survey on Network Intrusion Detection using Convolutional Neural Network”,ITM Web Conf., 43 (2022) 01003, DOI:https://doi.org/10.1051/itmconf/20224301003.

10. Moisés Toapanta, Enrique Mafla, Bryan Benavides, Dario Huilcapi,”Approach to Mitigate the Cyber-Environment Risks of a Technology Platform”, The International Conference on Information and Computer Technologies (ICICT-2020), USA, March 2020,DOI:10.1109/ICICT50521.2020.00069.

11. Hanan Hindy, David Brosset, Ethan Bayne, Amar Seeam, Christos Tachtatzis, Robert Atkinson, and XavierBellekens. 2018. A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets. 1, 1 (June 2018).

.