Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Distributed Denial-of-Service (DDoS) Attack Detection using 1D Convolution Neural Network (CNN) and Decision Tree Model

Bhargavi Goparaju[1,*], Bandla Sreenivasa Rao[1]

[1] Acharya Nagarjuna University, Nagarjuna Nagar, Guntur, Andhra Pradesh 522510, India

ARTICLE INFO

Article history:
Received 5 April 2023
Received in revised form 26 June 2023
Accepted 10 July 2023
Available online 10 September 2023

Keywords:
Distributed Denial-of-Service; attack; convolution neural network; Decision tree; network security

ABSTRACT

The major problem of internet security is a Distributed Denial-of-Service (DDoS) attack, which can't be detected easily. This attack is said to have occurred when lots of service requests are simultaneously received at a server on the internet. This makes the server too busy to provide normal services for others. The Distributed Denial of Service (DDoS) attacks nature on large networks on the Internet demanding to develop the effective detection and response methods. The deployment of these technique should perform not only at the network core but also at the edge. A DDoS attack detection framework is presented based on transfer learning model consisting of 1D Convolution Neural Network (CNN) and decision tree classifier. The 1D CNN model utilizes for features extraction from the input network traffic data. This operation also reduces the dimension of the data thereby removing the redundancy in the data. These features are given to the decision tree model for classification. The proposed framework identified the DDoS attacks with good accuracy. This system could identify attacks in real-time and provide network security.

1. Introduction

The latest components used for network security are the intrusion detection systems or IDS. Their objective is to detect or prevent intrusions by analyzing the exchanges between or within systems [1]. For this, these systems must be able to: capture these exchanges, identify an attack in these captures and raise an alert and intrusion attempt after identification. It can also identify two main types of IDS: 1). IDS Networks: which monitor the exchanges carried out on the network, also called NIDS (or NIPS), which work with a specific probe positioned on the network [2]. 2). Host IDS / IPS: which monitor security at host levels, therefore directly within the systems, also called HIDS (or HIPS), which work using one or more probes implanted in the hosts [3].

The type of IDS that interests us in our case is the NIDS, since it is the one that will make it possible to monitor communications between connected objects in an environment. HIDS are often used in systems to verify that actions carried out within the system do not impact the security properties, for

* *Corresponding author.*
E-mail address: gbhargavi5007@gmail.com

https://doi.org/10.37934/araset.32.2.3041

example by checking the log files, the system calls made or by verifying access to certain resources on this system [4]. An example of HDS was developed within LAAS-CNRS. HIDS could be implemented in objects connected to ensure the safety properties internally. However, the same constraints that previously apply, and only collaboration with manufacturers can allow the installation of this type of component.

In the case of NIDS, existing solutions often cover traditional protocols such as Wi-Fi, or application protocols. However, the many wireless protocols of IoT are often not covered, or only partially. A potential solution would be to equip a NIDS with a large number of receivers suitable for all IoT protocols [5]. However, these protocols tend to evolve and new ones appear and disappear in environments, requiring reconfigurations regular IDS to be able to capture and monitor all exchanges, even those ad hoc. In addition, proprietary protocols whose specifications are unknown beforehand, which are characteristic of the IoT.

To respond to IoT issues, a similar specific component to a NIDS would therefore be an interesting solution, provided that it is sufficiently generic to limit the need for reconfiguration and to be able to monitor all protocols, especially proprietary ones [6].

For the identification of attacks, two different strategies are used: **Signature IDS**: an attack is identified as such if the elements of the capture (packet content, metadata) correspond to the signature of a known or already identified attack [7]. This operating mode is similar to that of antiviruses, which use an attack signature database to be able to identify a potential threat. **Behavioral IDS / IPS**: an attack is identified as such if the elements of the capture correspond to abnormal behavior [8]. In the case of these IDS / IPS, a normal behavior model is established beforehand, corresponding to all communications considered legitimate, therefore without malice. The capture of a communication is therefore compared to this model, and if it is significantly different from the legitimate model, this communication is identified as illegitimate.

Each type of identification has its advantages and disadvantages. The Signature IDS / IPS are more accurate, since they have comprehensive knowledge attacks identified as such. However, an attack not having never been encountered or not listed in the signature database will not be detected as an attack [9]. On the contrary, the Behavioral IDS are able to detect unknown attacks beforehand, since they will be able to identify such malicious communication in terms of its differences from the established behavioral model. These types of IDS are nevertheless less precise, since if a legitimate behavior was not integrated into the behavioral model it will be wrongly identified as an attack. In the case of connected objects and environments, the diversity of objects and the relative novelty of the protocols used make the behavioral approach more interesting, since the risk of zero-day vulnerabilities is higher in the case of recent technologies. The figure 1 is the intrusion detection systems or IDS in the below,

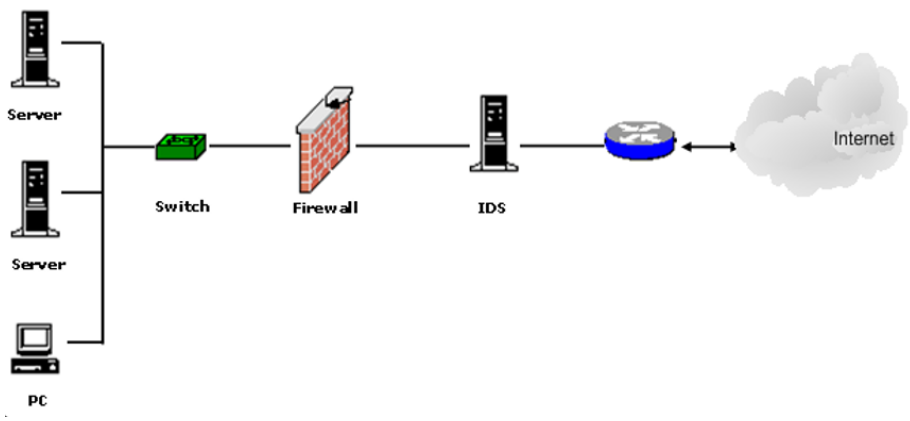

Fig. 1. The Intrusion Detection Systems Or IDS

Finally, these different traditional approaches therefore cover a certain number of malicious acts that can be perpetrated in connected environments, especially those coming from outside. Indeed, the implementation of solutions network security is based on network entry and exit points, such as a firewall, may be sufficient to detect and protect systems within the local network attacks from the Internet [10]. However, in the case of communications taking place within the local network, these solutions are very incomplete, and do not allow not effectively protecting or detecting attacks using protocols decentralized wireless communications. Thus, believe that only a solution generic, having the ability to monitor all protocols at the same time, even those with no known specifications, would be able to provide a mechanism fault tolerance viable in these connected environments. Section one presents introduction about the attacks and intruder detection systems. Section two depicts literature. Section three presents the proposed framework followed by results and conclusion.

2. Literature

As DDoS attacks increase in scale, frequency and sophistication, traditional methods based on out-of-band scrubbing centers and inter manual breakdowns are as inefficient as they are costly [11]. In the particular case of volumetric attacks, the deviation suspicious traffic to scrubbing centers weighs not only in terms of latency but also of cost, the latter being directly linked to the volume of data to be neutralized. Furthermore, this traditional approach requires analysis and intervention human, which only increases the latency and increases the costs remediation. With these methods, no less than 30 minutes may elapse between detection and neutralization, a delay unacceptable when we know that it often does not take more than a few minutes to cripple a website.

In a hyper connected world where the slightest interruption of service is paid in cash, it is essential to review its strategy anti DDoS and set up faster defenses, more efficient and more economical [12]. IP networks must constitute the first line of defense against volumetric attacks. Telemetry, machine analysis and network programmability then intervene to promote the development of a process smarter, more automated detection and neutralization and more adaptable.

ASERT (2012) and Prolex Security Engineering & Response Teams Curt Wilson and Arbor Networks (2013) [3, 4], [13, 14] DDoS attacks from compromise computer networks called botnets can be initiated. Many online tools allow botnets to be used. Online DDoS services have been introduced in recent years, usually called booters or stress [15]. These services provide rates that enable individuals to use them and permit a consumer to attack the target of his choice. Some booters also provide free testing of the service for a few minutes. Many of these attacks are increased by the wide variety of instruments and service that enable distributed denial of services attacks to be launched.

Warif *et al*., [16] has presented the Reflection-based attacks that utilize the accessible machines on the Internet and provide responses to the requests from any source. The packets transmission to these reflectors based on the IP address of victim as the source IP known as IP address spoofing is involved in a reflex attacks. The unwanted traffic generates by the victim to the latter’s destination is resulted by these responses of reflectors. This traffic may be big enough to saturate the network connections for the victim, which results in a service denial. By spoofing the victim's IP, the intruder queries the servers. The servers then submit the answer to the attacker's requests. The service denial is successful when the traffic volume caused by such replies exceeds the network bandwidth available to the victim.

Reflective attacks also include UDP transport protocol-based protocols. The UDP protocol actually lets the root task identification of the application layer that allows the spoofing of the IP address. By the way, UDP would not have to set up a session before sending data (unlike the TCP

protocol). A query is sent to a service based on a UDP packet is allowed by an attacker in this function and a reflector response is provided [17].

In order to make one or more networks unavailability, a volumetric attack aims to exhaust the available network bandwidth [18]. This form of attack is also carried out using the characteristics of such protocols to optimize traffic. Moreover, volumetric attacks are designed to saturated processing resources of a target by generating an extremely high number of packets per second. Some protocols have far larger answers than the request.There may also be more than the number of packets caused by the response than those needed to submit the request. The amplification of these protocols can be used for volumetric attacks.

Chen *et al*., [19] presented the Amplification attacks, some implementations of NTP, a protocol for time synchronization machines on an IP network, can also be used to conduct amplification attacks. The amplification can come from sending a control message (for example, *read-var*) allowing to retrieve information on the state of a server and possibly, to modify this state. The amplification resulting from this type of query varies depending on system and implementation. Observations show a fac-significant amplification factor in terms of bandwidth of about 30. Thus, a attacker can generate traffic to a target whose volume is 30 times larger than that needed to query vulnerable servers.

In addition, a query implemented in *ntpd* and intended for supervision purposes, used for retrieving the IP addresses list that have queried server. This list also includes various information relating to requests, such as the dates of terrogation or the number of packets received. Implementations with this feature keeps in memory the data of the last 600 requests that the server has received. By exploiting this functionality, tests make it possible to that it is possible to obtain an amplification factor in terms of bandwidth of about 1290, and a packet amplification factor of 99 [20].

As mentioned by Lee *et al*., [21], to detect an incident, it is essential to have means of supervision infrastructure, both in terms of the network and the services operated. In particular, monitoring enables monitoring of the evolution of the use of resources, such as bandwidth, memory and processor or space disk. Significant changes seen in these services may imply an operating problem and possible service denial. Mobile Ad-hoc networks are distributed and made up of different individual devices that can communicate with one another. Attacks on the network can take many different forms due to its scattered nature and weak infrastructure [26, 27].

Network traffic can be monitored using protocols such as Net- Flow or IPFIX (IP Flow Information Export) [22]. These protocols make it possible to obtain information on network exchanges in the form of flows described by IP addresses source or destination, the transport protocol used, the source or destination ports, as well as other characteristic elements of the traffic [23]. NetFlow or IPFIX are today available on a large number of network devices, and allow export flows to collectors who can thus aggregate, in real time, information relating to network traffic. Analysis of data collected using analysis tools NetFlow or IPFIX can be used to detect significant variations in traffic.

3. Proposed DDOS Attach detection model

The proposed framework is designed to take the network data as input and the 1D CNN model is used to extract the features from the data. The extracted features are given to the decision tree model for classification. The transfer learning model consists of 1D CNN and decision tree classification model. The figure 2 shows the proposed transfer learning-based DDoS attack detection model.

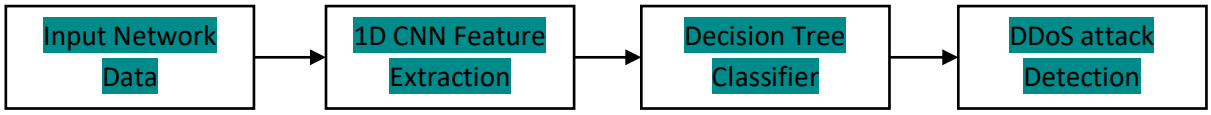

Fig. 2. Proposed Transfer Learning Based DDoS Attack Detection Model

3.1 1D Convolutional Neural Network (CNN)

CNNs are one of the most frequently used computer vision neural networks to identify image patterns and artifacts. In general, three different types of layers shape a convolution network. This consists of the convolution layer, the pooling layer (POOL) and the fully connected layer (FC). The completely connected layer(s) are also used to exit the network. Normally, an activated layer follows a convolution layer, and then a pooling layer will repeat this sequence many times before the layer is completely connected to the network which is sometimes referred to under the notation CONVNET.Main Model will therefore have an architecture of type ENTRANT-CONV-ACTIVATION-POOL-FC where the three intermediate steps CONV-ACTIVATION-POOL can be repeated several times before we arrive at a fully connected FC layer. Before leaving the network, it is normal to use more than a completely connected FC layer. Figure 3 is Classical architecture of a convolutional neural network.

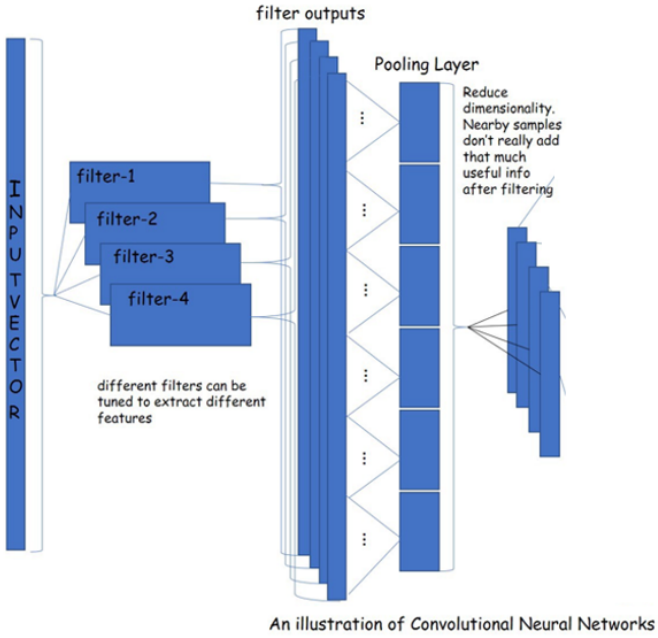

Fig. 3. Classical Architecture of a Convolutional Neural Network.

A stack of independent computing layers constitutes a CNN architecture:

- The convolutive layer (CONV) that processes the receiving field data.
- The pooling layer (POOL), allowing the compression of information by reducing the image size (often by downsampling).

Convolution layer

Convolutional neural networks have their own layer called the convolutional layer, which is a clear difference from recurrent neural networks and other neural networks. The convolution layer uses a filter to transform the input before it is passed to the next layer.

Pooling layer

A convolutional neural network repeats convolution and pooling. Pooling is to reduce the learning size according to a set rule, and in pooling, subsampling is performed using the average value and maximum value.

There are three types of pooling. Average pooling extracts the average value of the specified easy pixel values, but it has the disadvantage that it takes time for learning to converge. The second maximum pooling compresses to the maximum value in the specified range. Maximum pooling can eliminate the effects of small values, but there is a risk of falling into a local solution. The figure 4 shows the Max Pooling. Finally, Lp pooling is a technique that multiplies the surrounding values to the p-th power and takes their standard deviation, including both maximum and average pooling. It can focus on large values and reduce the effects of small values.

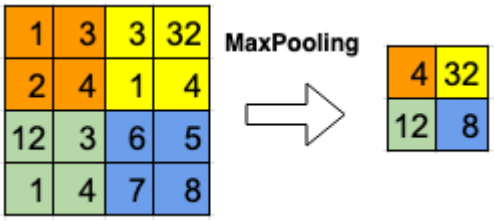

Fig. 4. MaxPooling

In earlier section, the traditional deep CNNs have designed and presented for operating on 2D data such as videos and images. That's the reason, they are called as "2D CNNs". Alternatively, they have modified and developed as 1D Convolutional Neural Networks (1D CNNs) [47-56]. The investigations have shown that 1D CNNs are beneficial for certain applications and 1D signals are more preferable to 2D because of the below reasons such as:

In 1D CNNs, simple array operations require for BP and FP instead of matrix operations. Compared to 2D CNNs, the computational complexity is lower for 1D CNNs.

Based on the recent studies, the challenging tasks that involve 1D signals can be learnt by 1D CNNs with shallow architectures including small number of neurons and hidden layers. To deal with these tasks, deeper architectures are needed for 2D CNNs. It's an easy way to train and implement the networks with shallow architectures.

Special hardware setup is needed for training deep 2D CNNs. For example, GPU or cloud computing. To train the compact 1D CNNs with neurons (<50) and hidden layers (2 or less), GPU implementation is robust and feasible unlike a standard computer.

The compact 1D CNNs are easily adapted for low cost and real-time applications specifically mobile or hand-held devices owing to the low-level computational requirements.

3.2 Decision Tree Algorithm Classification

For both regression and classification issues, Decision Tree is considered as a supervised learning method, but it is mostly used for solving classification issues. It represents as a tree-structured classifier, in which each leaf node refers to the outcome, branches refer to the decision rules, and internal nodes refer to the dataset features. Two different nodes include in a decision tree such as Leaf Node and Decision Node. Here, decision nodes can use for making any decision and they have multiple branches. Leaf nodes represent those decisions' output, and any further branches are not contained. Based on the features of given dataset, the decisions or tests have processed. To obtain all possible solutions to a decision or problem, it is a graphical representation based on given constraints. It's called as a decision tree, which displays as a tree starting with the root node and expanding further branches and a tree-like structure is constructed. The figure 5 shows the CART algorithm or Classification and Regression Tree algorithm is used for building a tree.

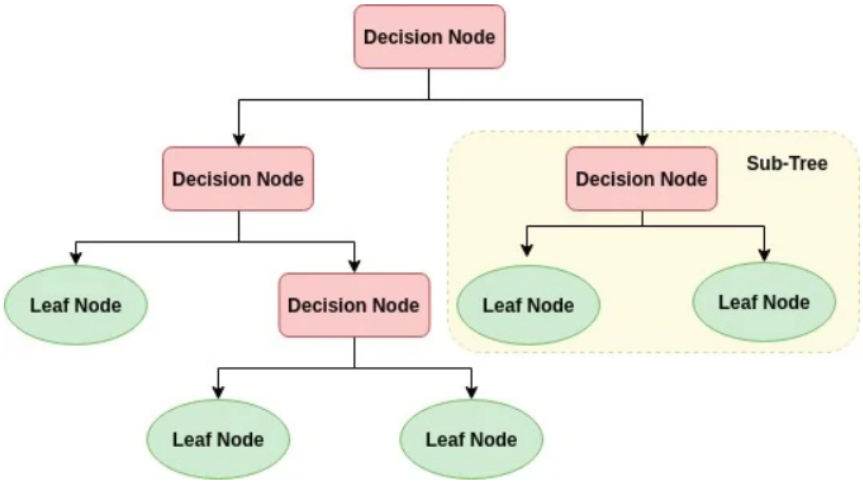

Fig. 5. Decision Tree for Machine Learning

Working of Decision Tree algorithm

The algorithm is started from the tree's root node to predict the given dataset class in a decision tree. The root attribute values with the record (real dataset) attribute have compared by this algorithm. The branch follows and jumps to the next node based on the comparison.

Again, the algorithm has compared the attribute value with the other sub-nodes for the next node and move further. Until it reaches to the tree's leaf node, the process has continued. The below algorithm can be used to understand the complete process:

- **Step-1:**Start the tree with the root node, known as S, which includes the complete dataset.
- **Step-2:** Determine the best attribute based on the **Attribute Selection Measure (ASM)** in the dataset.
- **Step-3:** Categorize S into subsets that consist of possible values for the best attributes.
- **Step-4:** The decision tree node generates, that contains the best attribute.

- **Step-5:** The dataset subsets that created in Step-3 have used to make new decision trees recursively [24]. This process continues up to reaching the phase where you can't classify the nodes further and the final node is known as a leaf node.

4. Experimental Results

In this section, the experimental analysis is presented to verify the proposed technique. It's often used to monitor the training progress when training networks for deep learning [25]. In training options, the 'training-progress' specifies as the 'Plots' value and network training is started. A figure creates by the trained networks and the training metrics have displayed at each iteration. The gradient is estimated and network parameters are updated for each iteration.

- On each individual mini-batch, Training accuracy and classification accuracy estimate.
- Smoothed training accuracy – A smoothing algorithm applies to the training accuracy for obtaining the smoothed training accuracy. Compared to the unsmoothed accuracy, it is less noisy that makes easier for spot trends.
- Validation accuracy – classification accuracy based on the validation set (specified based on training options)
- Smoothed training loss, training loss, and validation loss – the loss on smoothed version of mini-batch, the loss on each mini-batch, and the loss on the validation set, respectively. The loss function is the cross-entropy loss if the network final layer is a classification layer.

Figure 6 shows the training accuracy plot of 1DCNN network. While figure 7 represents the training loss plot.

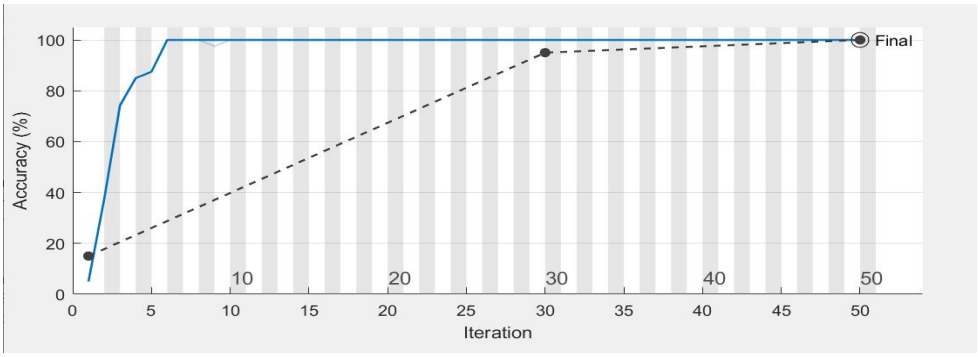

Fig. 6. Training Accuracy Graph

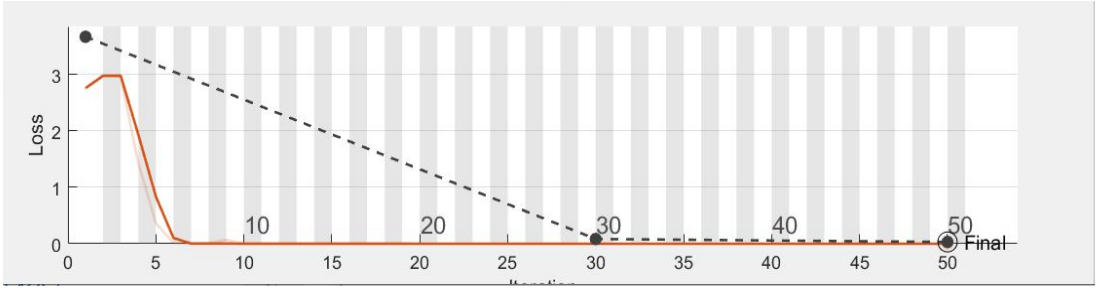

Fig. 7. Training Loss Plot

Training processing

The data set contains 2000 entries, of which 1000 are attacks and 1000 genuine user requests. From this data the total of 1700 (850 attacks, 850 normal) is given for training, remaining 300 (150 attacks, 150 normal) are given for testing.

The table 1 shows the training confusion matrix. In this matrix attacks accuracy is 96.70, Non-attacks accuracy is 97.29 and the model accuracy is 96.99. The table 2 shows the training parameter values and the model accuracy is 0.97.

Table 1
Training Confusion Matrix

Actual class / Predicted class	Attacks	Non attacks	Accuracy
Attacks	822	28	96.70
Non attacks	23	827	97.29
Model accuracy			96.99

Table 2
Training Parameters

Actual class / Parameters	Attacks	Non attacks
TP	822.0	827.0
FP	28.0	23.0
FN	23.0	28.0
TN	827.0	822.0
Precision	0.97	0.97
Sensitivity	0.97	0.97
Specificity	0.97	0.97
Model Accuracy	0.97	

To evaluate the classification model performance, a confusion matrix N*N matrix is used, where N refers to the number of target classes. The actual target values compare with those predicted values by the matrix in the machine learning model.

Testing

Table 3 shows the testing confusion matrix. In this matrix attacks accuracy is 97.33, non-attacks accuracy is 95.33 and the model accuracy is 96.33. The table 4 shows the testing parameter values and the model accuracy is 96.33.

Table 3
Training Confusion Matrix

Actual class / Predicted class	Attacks	Non attacks	Accuracy
Attacks	146	4	97.33
Non attacks	7	143	95.33
Model Accuracy	96.33		

Table 4
Testing Parameter Values

Actual class / Parameters	Attacks	Non attacks
TP	146.0	143.0
FP	4.0	7.0
FN	7.0	4.0
TN	143.0	146.0
Precision	97.0	95.0
Sensitivity	95.0	97.0
Specificity	97.0	95.0
Model Accuracy	96.33	

5. Conclusion

Based on a transfer learning model, a DDoS detection system design is presented in this work. For cloud computing, attack detection is a research area to focus on making the cloud as a trusted and secure platform for the future Internet of Things. The proposed transfer learning model consists of 1D CNN and decision tree model. The 1D CNN extracts the features from the input data and these features are given to the transfer learning model for classification. The training accuracy is recorded as 97 percent and the testing accuracy is recorded as 96.33 percent.

References

[1] Borkar, Amol, Akshay Donode, and Anjali Kumari. "A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS)." In *2017 International conference on inventive computing and informatics (ICICI)*, pp. 949-953. IEEE, 2017. https://doi.org/10.1109/ICICI.2017.8365277

[2] Hodo, Elike, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, and Robert Atkinson. "Shallow and deep networks intrusion detection system: A taxonomy and survey." arXiv preprint arXiv:1701.02145 (2017).

[3] Haider, Waqas, Gideon Creech, Yi Xie, and Jiankun Hu. "Windows based data sets for evaluation of robustness of host-based intrusion detection systems (IDS) to zero-day and stealth attacks." *Future Internet* 8, no. 3 (2016): 29. https://doi.org/10.3390/fi8030029

[4] Mehmood, Amjad, Mithun Mukherjee, Syed Hassan Ahmed, Houbing Song, and Khalid Mahmood Malik. "NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks." *The Journal of Supercomputing* 74, no. 10 (2018): 5156-5170. https://doi.org/10.1007/s11227-018-2413-7

[5] Verma, Abhishek, and Virender Ranga. "ELNIDS: Ensemble learning based network intrusion detection system for RPL based Internet of Things." In 2019 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU), pp. 1-6. IEEE, 2019. https://doi.org/10.1109/IoT-SIU.2019.8777504

[6] Li, Daming, Lianbing Deng, Minchang Lee, and Haoxiang Wang. "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning." *International journal of information management* 49 (2019): 533-545. https://doi.org/10.1016/j.ijinfomgt.2019.04.006

[7] Cortés, Francisco Muñoz, and Natalia Gaviria Gómez. "A hybrid alarm management strategy in signature-based intrusion detection systems." In 2019 IEEE Colombian Conference on Communications and Computing (COLCOM), pp. 1-6. IEEE, 2019. https://doi.org/10.1109/ColComCon.2019.8809121

[8] Arrington, Briana, LiEsa Barnett, Rahmira Rufus, and Albert Esterline. "Behavioral modeling intrusion detection system (bmids) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms." In 2016 25th International Conference on Computer Communication and Networks (ICCCN), pp. 1-6. IEEE, 2016. https://doi.org/10.1109/ICCCN.2016.7568495

[9] Barki, Lohit, Amrit Shidling, Nisharani Meti, D. G. Narayan, and Mohammed Moin Mulla. "Detection of distributed denial of service attacks in software defined networks." In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2576-2581. IEEE, 2016. https://doi.org/10.1109/ICACCI.2016.7732445

[10] Gelenbe, Erol, and Yasin Murat Kadioglu. "Energy life-time of wireless nodes with network attacks and mitigation." In 2018 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE, 2018. https://doi.org/10.1109/ICCW.2018.8403561

[11] Tan, Zhiyuan, Aruna Jamdagni, Xiangjian He, Priyadarsi Nanda, Ren Ping Liu, and Jiankun Hu. "Detection of denial-of-service attacks based on computer vision techniques." IEEE transactions on computers 64, no. 9 (2014): 2519-2533. https://doi.org/10.1109/TC.2014.2375218

[12] Cui, Yunhe, Lianshan Yan, Saifei Li, Huanlai Xing, Wei Pan, Jian Zhu, and Xiaoyang Zheng. "SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks." *Journal of Network and Computer Applications* 68 (2016): 65-79 https://doi.org/10.1016/j.jnca.2016.04.005

[13] Curt Wilson, Arbor Networks ASERT, « Attack of the Shuriken: Many Hands, Many Weapons ». <http://www.arbornetworks.com/asert/2012/02/ddos-tools/ >, fév. 2012.

[14] Prolexic Security Engineering & Response Team, « Prolexic Quarterly Global DDoS Attack Report Q3 2013 ». <http://www.stateoftheinternet.com/ resources-web-security-2013-q3-global-ddos-attack-report.html>, oct. 2013.

[15] Krebs, Brian. "DDoS services advertise openly, take paypal." *Krebs on Security* (2013).

[16] Warif, Nor Bakiah Abd, Ainuddin Wahid Abdul Wahab, Mohd Yamani Idna Idris, Rosli Salleh, and Fazidah Othman. "SIFT-symmetry: a robust detection method for copy-move forgery with reflection attack." *Journal of Visual Communication and Image Representation* 46 (2017): 219-232. https://doi.org/10.1016/j.jvcir.2017.04.004

[17] Zhang, Menghao, Guanyu Li, Lei Xu, Jun Bi, Guofei Gu, and Jiasong Bai. "Control plane reflection attacks in SDNs: New attacks and countermeasures." In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pp. 161-183. Springer International Publishing, 2018. https://doi.org/10.1007/978-3-030-00470-5_8

[18] Rudman, Lauren, and B. Irwin. "Characterization and analysis of ntp amplification based ddos attacks." In 2015 Information Security for South Africa (ISSA), pp. 1-5. IEEE, 2015. https://doi.org/10.1109/ISSA.2015.7335069

[19] Chen, Chih-Chieh, Yi-Ren Chen, Wei-Chih Lu, Shi-Chun Tsai, and Ming-Chuan Yang. "Detecting amplification attacks with software defined networking." In 2017 IEEE conference on dependable and secure computing, pp. 195-201. IEEE, 2017. https://doi.org/10.1109/DESEC.2017.8073807

[20] NTP development team, « NTP Software Downloads ». <http://www.ntp.org/downloads.html>.

[21] Lee, Keunsoo, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, and Sehun Kim. "DDoS attack detection method using cluster analysis." *Expert systems with applications* 34, no. 3 (2008): 1659-1665. https://doi.org/10.1016/j.eswa.2007.01.040

[22] Claise, Benoit. *Cisco systems netflow services export version 9*. No. rfc3954. 2004. https://doi.org/10.17487/rfc3954

[23] Claise, Benoit, Brian Trammell, and Paul Aitken. *Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information*. No. rfc7011. 2013. https://doi.org/10.17487/rfc7015

[24] San, Gan Shu, Siana Halim, Debora Anne Yang Aysia, and Jessie Lestari. "Predicting Willingness to Donate Smartphones as a Reuse Option Using Decision Tree Analysis." PhD diss., Petra Christian University, 2023. https://doi.org/10.37934/araset.30.3.315324

[25] Zamri, Nurul Farhana Mohamad, Nooritawati Md Tahir, Megat Syahirul Megat Ali, Nur Dalila Khirul Ashar, and Ali Abd Almisreb. "Real Time Snatch Theft Detection using Deep Learning Networks." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 31, no. 1 (2023): 79-89. https://doi.org/10.37934/araset.31.1.7989

[26] Subburayalu, Gopalakrishnan, Hemanand Duraivelu, Arun Prasath Raveendran, Rajesh Arunachalam, Deepika Kongara, and Chitra Thangavel. "Cluster based malicious node detection system for mobile ad-hoc network using ANFIS classifier." *Journal of Applied Security Research* 18, no. 3 (2023): 402-420. https://doi.org/10.1080/19361610.2021.2002118

[27] Gopalakrishnan, S., and A. Rajesh. "Cluster based Intrusion Detection System for Mobile Ad-hoc Network." In *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, vol. 1, pp. 11-15. IEEE, 2019. https://doi.org/10.1109/ICONSTEM.2019.8918871