Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

A PRACTICAL, LIGHTWEIGHT DEEP LEARNING SOLUTIONFOR DDOS ATTACK DETECTION

M.Pallavi[1] , G.S.D.Dharani[2] , D.Lakshmi Sahithi[3] , D.S.V. Sreevarshini[4] , D.Mounika[5]
[1]Assistant Professor, Department of Computer Science and Engineering, Vignan's Institute of Engineering for Women, Visakhapatnam, India
[2-5] Department of Computer Science and Engineering, Vignan's Institute of Engineering for Women, Visakhapatnam, India

ABSTRACT
Distributed denial of service (DDoS) attacks represent a rising danger to organizations and government offices. They hurt web organizations, limit admittance to data and administrations, and harm corporate brands. Assailants use application layer DDoS assaults that are not effectively recognizable in light of imitating bona fide clients. In this review, we address novel application layer DDoS attacks by examining the qualities of approaching bundles, including the size of HTTP outline parcels, the quantity of Internet Protocol (IP) addresses sent, consistent mappings of ports, and the quantity of IP tends to utilizing intermediary IP. We broke down client conduct openly goes after utilizing standard datasets, the CTU-13 dataset, genuine weblogs (dataset) from our association, and tentatively made datasets from DDoS attack apparatuses: Slow Nests, Mass, Brilliant Eyes, and Xerex. A multilayer perceptron (MLP), a profound learning calculation, is utilized to assess the viability of measurements based assault discovery. Recreation results show that the proposed MLP grouping calculation has an effectiveness of 98.99% in recognizing DDoS assaults. The exhibition of our proposed procedure offered the most minimal benefit of bogus up-sides of 2.11% contrasted with regular classifiers, i.e., Naïve Bayes, Decision Stump, Logistic Model Tree, Naïve Bayes Updateable, Naïve Bayes Multinomial Text, AdaBoostM1, Attribute Selected Classifier, Iterative Classifier, and OneR.

Keywords:
ML , Model , Script, DDoS, Traffic snippets.

INTRODUCTION
In the present high speed world, where the quantity of web associated gadgets is expanding and online applications are developing at a fast speed, data security is turning into an outright need. Starting from the start of the Internet, 1.2 billion sites have been created, and a colossal number and assortment of online applications are incorporated with different web administrations, like web based business, web based banking, web based shopping, online instruction, e-medical care, and industrial control systems (ICS) for basic foundation, and so on. These days, digital assailants are profoundly talented and exceptional to do effective assaults on organizations and legislatures. Cybercrime is huge business today, and the volume of taken data is gigantic. There are various classes of malware. This represents a gigantic gamble to legislatures, organizations, and customers all over the planet. We don't need to travel far once again into the memorable past the huge assault on a bank in Bangladesh, where USD 81 million was purportedly taken. This is a consistent indication of how viable these attacks can be; the bank's own PCs were utilized to move huge amounts of cash. No business is protected, regardless of how huge. Measurements show that 20% of impacted organizations fall into the independent company class, 33% into the SME classification, and 41% into the huge business class. The more broad the danger, the more significant it becomes to know about the issues and safeguard the significant data. 82% of associations have been presented to somewhere around at least one attacks in which information are taken and used to disable the casualty's administrations. The associations that

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

were impacted by DDoS attacks revealed a 26% drop in execution of their administrations and 41% detailed a blackout of the impacted administrations. Figure 1 shows a climate of DDoS attacks.

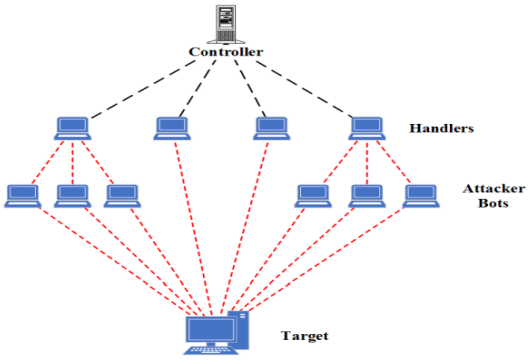


Fig . 1 Ddos Attack Environment.

REQUIREMENT ANALYSIS

Introduction of Requirement Analysis

This task has fundamental necessities to construct and improve the Jupyter ID E Application complete advancement which happens on Jupyter Scratch pad (Expected - Python Programming Language) worked with perfect and upgraded plan based client experience.

Hardware & Software Requirements

Software Requirements		Hardware Requirements	
Framework	Jupyter Notebook	Processor	I5 else Ryzen 5000 series or Above
Programming Language	Python	RAM	8Gb or Above
Editor	VS Code	SSD	128Gb
		GPU	(Optional)

Software Requirement Specifications

SRS is a report made by framework examiners after the necessities are gathered from different partners. SRS characterizes how the expected programming will communicate with equipment, outer connection points, speed of activity, and reaction season of framework, transportability of programming across different stages, practicality, speed of recuperation subsequent to crashing, security, quality, restrictions and so on. The necessity got from the client is written in normal language. It is the obligation of framework examiners to archive the necessities in specialized language so they can be understood and valuable by the product improvement group.

SRS should come up with following features:

Natural language is used to express the needs of the user. Specialized necessities are communicated in organized language, which is utilized inside the association.

System Design: This phase examines the requirements from the first phase and prepares the system design. System design helps define the system's overall architecture and specifies hardware and system requirements.

LITERATURE SURVEY

[1] Noe Marcelo Yungaicela-Naula , Cesar Vargas-Rosales ,And Jesus Arturo Perez-Diaz ,"SDN-Based Architecture for Transport and Application Layer DDoS Attack DetectionbyUsing Machine and Deep Learning" [1]. DoS/DDoS attacks are the most harmful threats affecting the networks. In this work, we presenteda solution using artificial intelligence to detect two types of threats: transport layer andapplication layer DDoS attacks. First, we proposed a modular SDN-based architecturewithcomponents that can be modified or improved separately, providing flexibility to test different intelligent methods to detect diverse attacks. Moreover, we explored four DL models andthreeML models that demonstrated an accuracy performance above 99% on the testing phase, usingtwo up-to-date datasets with real network traces. Our proposed solution was evaluatedinanemulated test bed, using Mininet and the ONOS controller. In this configuration, the detectionrate of the trained models remained high enough. We presented a ranking of the best modelsevaluated on the test bed, and we concluded that GRU and LSTM models maintained the highest detection rates in the inline model evaluation. These models achieved high detection rates, upto95% for slow-rate attacks and above

[2] Harish Kumar ,Yassine Aoudni ,Geovanny Genaro Reivan Ortiz ,Latika Jindal ,ShahajanMiah , and Rohit Tripathi, "Light Weighted CNN Model to Detect DDoS Attack over Distributed Scenario" [2]. Aiming at four types of minimal-degree DDoS assaults, this study obtains minimal degree DDoS assault data sets, analyzes and obtains 40 effective traits of minimal-degree DDoS assaults, and proposes a variable-kind minimal-degree DDoS based on CNN-RF hybrid learning. The attack detection method and online deployment of this model realize connected revealing of variable types of minimal degree DDoS assaults. Furthermore, an online detection time window is proposed, and the online detection performance is evaluated using false intervention degree and malicious network congestion revealing rate. Experiments show that the prototype based on CNN-RF hybrid deep learning algorithm can accurately detect different types of minimal-degree DDoS assaults. At the identical interval, the revealing method in this study is highly portable, and the minimal-degree DDoS assault data set is used close to the actual situation, which can be deployed and applied in practical environments when the hybrid deep learning model implements training and detection for multi-type

[3] Shahzeb Haider , Adnan Akhunzada , Iqra Mustafa, Tanil Bharat Patel, Amanda Fernandez, Kim-Kwang Raymond Choo , (Senior Member, Ieee), And JavedIqbal1,"ADeep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks" [3]. Contemporary innovative research and novel cyber security solutions are in dispensable to properly secure the new era of digitization. We proposed an efficient and scalable deep CNN ensemble framework to address the issue of the SDN.We evaluated our proposed framework with benchmark deep learning ensembles and hybrid state-of-the-art algorithms on a flow-based SDN dataset. The proposed algorithm demonstrates improvements both in detection accuracy and computational complexity. Finally, We endorse varied deep learning ensemble based detection and prevention mechanisms for the emerging large-scale distributed networks.

[4] Shi Dong And Mudar Sarem ,"DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks " [4]. The DDoS attack is currently the most serious threat to network security in the SDN network. The detection of the DDoS attack is critical to the defence against the DDoS attack. The recent DDoS attack detection methods still have low accuracy of identification and they are vulnerable to other factors. To address the above problems, we have completed the following achievements: Firstly, our proposed four features (i. e., flow length, the DDoS attack. Secondly, for the first time, a new concept called the degree of attack is proposed and presented to detect the DDoS attack. Based on this concept, a detection algorithm called DDADA algorithm is proposed.

[5] R. Doriguzzi-Corin , S. Millar, S. Scott-Hayward , J. Martínez-del-Rincón, andD. Siracusa,"A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection"[5]. The performance of LUCID in classifying unseen traffific flflows as benign or malicious (DDoS). As illustrated, the very high performance is maintained across the range of test datasets indicating the robustness of the LUCID design. These results are further discussed , where we compare our solution with state-of-the-art in the scientifific literature. The results show that thanks to the properties of its CNN, LUCID learns to distinguish between patterns of malicious DDoS behaviour and benign flows. Given the properties of convolutional methods, these patterns are recognized regardless of the position they occupy in a flow, demonstrating that our spatial representation of a flow is robust.

PROPOSED SYSTEM

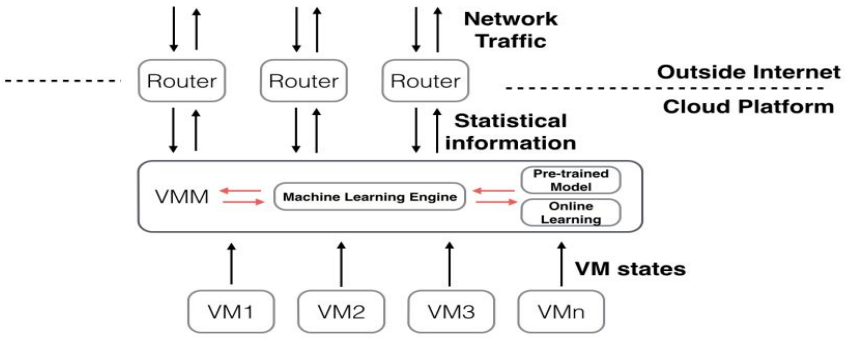
Method of Implementation:

This section contains the results of the experimental model using machine learning algorithms such as Random Forest, KNN, SVM algorithms. Jupyter tool is used for fetching the experimented results. Additionally, by subsampling the data, non-incremental learning methods can be used on large datasets. Weka also provides Hadoop and Spark compatibility as options for distributed data mining.

Attack Features Selection and Extraction

Before considering the feature selection and extraction, we first show the architecture of our system in Figure 1. The VMM monitors the status of the virtual machines and gathers statistical network traffic information. The VMM inputs the gathered information to a machine learning engine. The machine learning engine feeds back whether a suspicion sanction is detected. If the suspicious behaviour is only detected on a single VM, terminating this VM is a reasonable way to mitigate the attack

Architecture of Proposed Method



Data Collection

In our experiments, we collect network packages coming in and going out of the attacker virtual machine(s) for 2 hours. Four kinds of attacks are programmed to randomly start and end. Some of them may be started simultaneously.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

	dt	switch	pktcount	bytecount	dur	dur_nsec	tot_dur	flows	packetins	pktperflow	byteperflow	pktrate	Pairflow	tx_bytes	rx_bytes	tx_kbps	rx_kbps	t
46116	26657	8	20	1960	21	201000000	2.120100e+10	3	10	0	0	0	1	5327	5327	0	0.0	
46117	26657	8	20	1960	21	201000000	2.120100e+10	3	10	0	0	0	1	5457	3104	0	0.0	
46118	26657	8	20	1960	21	201000000	2.120100e+10	3	10	0	0	0	1	3227	3185	0	0.0	
46119	26657	8	20	1960	21	201000000	2.120100e+10	3	10	0	0	0	1	3357	1122	0	0.0	
46120	26657	8	20	1960	21	43000000	2.104300e+10	3	10	0	0	0	1	5327	5327	0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
104340	5262	3	79	7742	81	842000000	8.184200e+10	5	10	29	2842	0	0	15209	12720	1	1.0	
104341	5262	3	79	7742	81	842000000	8.184200e+10	5	10	29	2842	0	0	15099	14693	1	1.0	
104342	5262	3	31	3038	31	805000000	3.180500e+10	5	10	30	2940	1	0	3409	3731	0	0.0	
104343	5262	3	31	3038	31	805000000	3.180500e+10	5	10	30	2940	1	0	15209	12720	1	1.0	
104344	5262	3	31	3038	31	805000000	3.180500e+10	5	10	30	2940	1	0	15099	14693	1	1.0	

41321 rows × 19 columns

Table. 1

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103839 entries, 0 to 104344
Data columns (total 23 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   dt           103839 non-null  int64
 1   switch       103839 non-null  int64
 2   src          103839 non-null  object
 3   dst          103839 non-null  object
 4   pktcount     103839 non-null  int64
 5   bytecount    103839 non-null  int64
 6   dur          103839 non-null  int64
 7   dur_nsec     103839 non-null  int64
 8   tot_dur      103839 non-null  float64
 9   flows        103839 non-null  int64
 10  packetins    103839 non-null  int64
 11  pktperflow   103839 non-null  int64
 12  byteperflow  103839 non-null  int64
 13  pktrate      103839 non-null  int64
 14  Pairflow     103839 non-null  int64
 15  Protocol     103839 non-null  object
 16  port_no      103839 non-null  int64
 17  tx_bytes     103839 non-null  int64
 18  rx_bytes     103839 non-null  int64
 19  tx_kbps      103839 non-null  int64
 20  rx_kbps      103839 non-null  float64
 21  tot_kbps     103839 non-null  float64
 22  label        103839 non-null  int64
dtypes: float64(3), int64(17), object(3)
memory usage: 19.0+ MB
```

Table .2

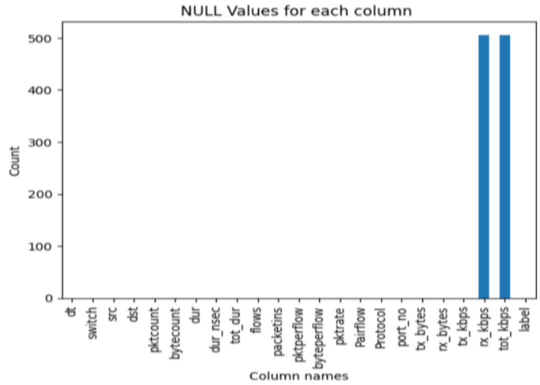
Data Processing

	dt	switch	pkt count	bytecount	dur	dur_nsec	tot_dur	flows	packetins
	pktperflow	byteperflow	pktrate	Pairflow	port_no	tx_bytes	rx_bytes		
	tx_kbps	rx_kbps	tot_kbps	label					
count	104345.000000	104345.000000	104345.000000	1.043450e+05					
	104345.000000	1.043450e+05	1.043450e+05	104345.000000	104345.000000				
	104345.000000	1.043450e+05	104345.000000	104345.000000					
	104345.000000	1.043450e+05	1.043450e+05	104345.000000	103839.000000				
	103839.000000	104345.000000							
mean	17927.514169	4.214260	52860.954746	3.818660e+07	321.497398	4.613880e+08			
	3.218865e+11	5.654234	5200.383468	6381.715291	4.716150e+06	212.210676			
	0.600987	2.331094	9.325264e+07	9.328039e+07	998.899756	1003.811420			
	2007.578742	0.390857							
std	11977.642655	1.956327	52023.241460	4.877748e+07	283.518232	2.770019e+08			
	2.834029e+11	2.950036	5257.001450	7404.777808	7.560116e+06	246.855123			
	0.489698	1.084333	1.519380e+08	1.330004e+08	2423.471618	2054.887034			

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

3144.437173 0.487945
min 2488.000000 1.000000 0.000000 0.000000e+00 0.000000 0.000000e+00
0.000000e+00 2.000000 4.000000 -130933.000000 -1.464426e+08 -
4365.000000 0.000000 1.000000 2.527000e+03 8.560000e+02 0.000000 0.000000
0.000000 0.000000
25% 7098.000000 3.000000 808.000000 7.957600e+04 127.000000 2.340000e+08
1.270000e+11 3.000000 1943.000000 29.000000 2.842000e+03 0.000000
0.000000 1.000000 4.743000e+03 3.539000e+03 0.000000 0.000000
0.000000 0.000000
50% 11905.000000 4.000000 42828.000000 6.471930e+06 251.000000 4.180000e+08
2.520000e+11 5.000000 3024.000000 8305.000000 5.521680e+05 276.000000
1.000000 2.000000 4.219610e+06 1.338339e+07 0.000000 0.000000
4.000000 0.000000
75% 29952.000000 5.000000 94796.000000 7.620354e+07 412.000000 7.030000e+08
4.130000e+11 7.000000 7462.000000 10017.000000 9.728112e+06 333.000000
1.000000 3.000000 1.356398e+08 1.439277e+08 251.000000 557.000000
3838.000000 1.000000
max 42935.000000 10.000000 260006.000000 1.471280e+08 1881.000000
9.990000e+08 1.880000e+12 17.000000 25224.000000 19190.000000 1.495387e+07
639.000000 1.000000 5.000000 1.269982e+09 9.905962e+08 20580.000000
16577.000000 20580.000000 1.000000

Null Values Extraction



Traffic for Normal and Malicious Traffic

Comparative results of all algorithms

RANDOM FOREST ALGORITHM

Random Forest is a popular machine learning algorithm used for both classification and regression tasks. It is an ensemble method that combines multiple decision trees and makes predictions by taking the average of the predictions made by each individual tree. The Random Forest algorithm works by creating a large number of decision trees, each of which is trained on a random subset of the training data and a random subset of the features. This randomness helps to reduce overfitting and improve the generalization of the model. During the prediction phase, each decision tree in the Random Forest makes a prediction, and the final prediction is obtained by taking the average of these predictions. This averaging process helps to smooth out the predictions and reduce the impact of individual noisy or biased decision trees

CONCLUSION

We propose a DDoS attack detection system based on deep learning to prevent attacks onthesourceside in the cloud. We extract statistical features of four DDoS attacks and launch real attacksinlabsettings for evaluation. Our proposed system is able to detect attacks with high accuracy(99.7%)andlow false positives (< 0.07%). By detecting DDoS attacks at the source virtual machines inthecloud,we can ”nip the attacks in the bud” and also protect the cloud provider’s reputation.

REFERENCES

[1] Noe Marcelo Yungaicela-Naula , Cesar Vargas-Rosales , (Senior Member, Ieee), AndJesus Arturo Perez-Diaz,”SDN-Based Architecture for Transport and Application Layer DDoSAttack Detection by Using Machine and Deep Learning”,Received July 9, 2021, acceptedJuly25, 2021, date of publication July 30, 2021, date of current version August 10, 2021.

[2] Harish Kumar ,Yassine Aoudni , Geovanny Genaro Reivan Ortiz ,Latika Jindal ,ShahajanMiah and Rohit Tripathi ,”Light Weighted CNN Model to Detect DDoS Attack over DistributedScenario”,Received 28 March 2022; Revised 8 May 2022; Accepted 17 May 2022; Published13June 2022.

[3] Shahzeb Haider , Adnan Akhunzada , Iqra Mustafa , Tanil Bharat Patel , Amanda Fernandez, “A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software DefinedNetworks”,Received January 31, 2020, accepted February 16, 2020, date of publicationFebruary27, 2020, date of current version March 26, 2020.

[4] Bin Jia, Yongquan Liang,”Anti-D Chain: A Lightweight DDoS Attack DetectionSchemeBased on Heterogeneous Ensemble Learning in Blockchain”.

[5] Shi Dong And Mudar Sarem,”DDoS Attack Detection Method Based on Improved KNNWiththe Degree of DDoS Attack in Software-Defined Networks ”,Received December 2, 2019, accepted December 22, 2019, date of publication December 30, 2019, date of current versionJanuary 8, 2020.

[6] R. Doriguzzi-Corin , S. Millar, S. Scott-Hayward , J. Martínez-del-Rincón , andD. Siracusa,”A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection”,IEEETRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 17, NO. 2, JUNE2020

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classifification withdeepconvolutional neural networks,” in Proc. 25th Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[8] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, “Deepanomaly: Fullyconvolutional neural network for fast anomaly detection in crowded scenes,” Comput. Vis. Image Understanding, vol. 172, pp. 88–97, Jul. 2018.

[9] Y. Kim, “Convolutional neural networks for sentence classifification,” in Proc. EMNLP, 2014, pp. 1746–1751.

[10] B. Alipanahi, A. Delong, M. Weirauch, and B. J Frey, “Predicting the sequencespecifificities of DNA- and RNA-binding proteins by deep learning,” Nat. Biotechnol., vol. 33, pp. 831–838, Jul. 2015

[11] D. Quang and X. Xie, “DanQ: A hybrid convolutional and recurrent deep neural networkforquantifying the function of DNA sequences,” Nucl. Acids Res., vol. 44, no. 11, p. e107, 2016.
[12] O. Janssens et al., “Convolutional neural network based fault detection for rotatingmachinery,” J. Sound Vib., vol. 377, pp. 331–345, Sep. 2016.
[13] A. Vilamala, K. H. Madsen, and L. K. Hansen, “Deep convolutional neural networksforinterpretable analysis of EEG sleep stage scoring,” in Proc. MLSP, 2017, pp. 1–6.
[14] N. McLaughlin et al., “Deep Android malware detection,” in Proc. CODASPY, 2017, pp. 301–308
[15] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, “A multimodal deep learning methodforAndroid malware detection using various features,” IEEE Trans. Inf. Forensics Security, vol. 14, no. 3, pp. 773–788, Mar. 2019.
[16] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, “Malware traffic classificationusingconvolutional neural network for representation learning,” in Proc. ICOIN, 2017, pp. 712–717.