ISSN 2278-3091

Volume 11, No.6, November - December 2022

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse011162022.pdf

https://doi.org/10.30534/ijatcse/2022/011162022

A Efficient Method to Detect DDos Attack in Cloud Computing

Asma Harbi Alashjaee[1], Dr Randa Ahmed Jabeur[2]

[1] Jouf University, SA, 421204012@ju.edu.sa

[2] Jouf University, SA, rjabeur@ju.edu.sa

Received Date: October 15, 2022 Accepted Date: November 20, 2022 Published Date: December 06, 2022

ABSTRACT

Electronic networks and cloud servers face many security problems since they are vulnerable to DDOS attacks. As cloud servers are exposed to attacks similar to legitimate requests on servers, it turns out that they are healthy, but they carry an attack on the data on the servers. DDOS attacks not only affect networks, but also the data carried by servers. In this paper, we proposed a new scheme to discover DDOS attacks. By using a Principal Component Analysis (PCA) scheme for network state analysis on traffic packet data, we divide and segment the network for reducing the overall computation. Comparing the results to the sample entropy, we were able to detect DDOS attacks more accurately.

Key words : DDOS attacks, Principal Component Analysis, Electronic networks .

1. INTRODUCTION

Cloud computing technology offers an all-inclusive collection of computer resources. It is accessible anytime, anywhere via the Internet at a minimal or no cost. Numerous business owners have increased the performance of their businesses and reduced IT costs by utilizing cloud computing. While cloud computing has numerous advantages over those on-premise However, they are also susceptible to internal as well as external attacks [1]. Therefore cloud developers have to implement security measures to guard their users' personal data from cyber attacks. This paper is written for cloud service providers and developers who want to increase the security of their cloud-based offerings**.**

The following are the contributions of this paper:

- We present a source-based DDoS defense technique used to reduce DDoS assaults in both fog and cloud environments.
- We employ SDN technology. It includes the DDoS defense module installed to stop network and transport-level DDoS attacks.
- The proposed system provides the deep-learning (DL)-based detection technique that is able to detect DDoS-infected traffic and can keep the same packet from reaching the cloud.

This work incorporates various machine learning algorithms like Random Forest, and Naïve Bayes for classification. In this paper, we look at solutions for detecting DDOS attacks and distinguishing them from legitimate requests to servers.

The remaining part of the paper is as follows: Section 2 we will pave the way for understanding what DDOS attacks are, the types of attacks that can penetrate cloud servers, what cloud servers are, and what is the harm of attacking them. In the 3 section, we will clarify some previous studies. In the 4 section, we will explain the methods of machine learning and deep learning to solve the problem of detecting DDOS attacks, and then we will clarify in the 5 section the results obtained. We will conclude with the 6 section of the paper.

2. DANGERS TO DATA

Data centers keep a variety of data types, and a sizable amount of that data contains sensitive information about individuals or businesses. However, due to human mistakes, programming defects, and unforeseen circumstances, this data is prone to lose, hacking, or corruption [2]. Cloud developers must apply cutting-edge encryption techniques to protect the integrity of data carried from the user to the cloud, even if it is clear that the cloud service provider cannot prevent all data dangers. [3].

2.1 Cloud computing attack vectors

Cloud computing attacks focus on accessing user data and hindering users from accessing cloud computing services. Both could cause significant harm to cloud users and reduce confidence in the security of cloud computing services [4]. Hackers typically monitor the communications between cloud users and apps when they are preparing cloud-based attacks by:

- Exploiting cloud computing vulnerabilities;

Obtaining users' credentials elsewhere other than the cloud

After the user passwords have been compromised, use the authentic pre-access that was provided to the cloud.

There are a variety of ways to hack Cloud computing platforms. Additionally, hackers are always trying to develop more complex cloud computing services [5].

However, understanding at least the most frequent of them will assist cloud developers in designing better secure applications. The research presents some of the most prevalent forms of cyber-attacks against cloud users.

2.1.1 Malware injection attacks in the cloud

Attacks from malware are used to alter user data stored in the cloud. Hackers achieve this by inserting a VM into an IaaS solution or an infected Service Execution Module into a SaaS or PaaS system. [6]. If the cloud server can be successfully impersonated, the user requests will be forwarded to the hacker instance or module, which then will begin to run malicious software. In addition, the attacker could begin damaging actions such as theft, data manipulation, or even eavesdropping. The two most popular types of malware attacks are cross-site Scripting and SQL injection. Cross-site scripting attacks involve hackers introducing malicious software (such as Flash, JavaScript, and other types of software) on a vulnerable website. An XSS attack on the cloud computing architecture of Amazon Web Services was planned in 2011 by German researchers. Attackers employed vulnerable database apps to target SQL servers while using SQL injection. An SQL injection attack was launched against the PlayStation website of Sony in 2008.

2.1.2 Excessive usage of cloud services

Hackers can use cloud services to conduct denial-of-service and brute-force attacks against specific people, companies, and different cloud providers. In 2010, for instance, security experts Brian Anderson and Brian Anderson coordinated a DoS attack by using the Amazon EC2 cloud architecture [7]. As a consequence, by investing only $6 to hire virtual services, they were able to make their clients unreachable on the Internet. At the 2011 Black Hat Technical Security Conference, Thomas Roth engaged in a brute force attack(8). By renting servers from cloud service providers, hackers can employ sophisticated cloud capabilities to send thousands of potential passwords to a target user's account.

2.1.3 Distributed Denial of Service (DDoS) attacks

DoS attacks aim to overburden the system and prevent users from accessing its services. These attacks pose a serious threat to cloud computing systems since even a single cloud server might result in issues for a large number of users [9]. When the workload grows, cloud systems begin to use additional virtual machines and service instances to give greater processing capacity. The cloud infrastructure makes a cyber assault more devastating while attempting to avoid it. Finally, the cloud infrastructure slows down, preventing genuine users from accessing their cloud services. DDoS assaults may be more harmful in a cloud setting if hackers deploy more zombie devices to target a large number of systems [10]

2.1.4 Attacks from the side channel

By putting a fake virtual machine in the same hosting environment as the targeted virtual machine, attackers create the possibility of a side-channel attack. Hackers target applications on systems using encryption techniques in the attack via side channels. If you build a secure and effective system, attacks of this kind could be prevented.

2.1.5 Attacks through the detour

In the context of cloud computing,[11] the wrapped attack could be a case of a man-in-the-middle attack. Since cloud users typically connect to their services through the web cloud computing is vulnerable to be a target for attempts to circumvent [12]. The use of an XML signature protects users' credentials from unwanted access, but it does not secure document locations. An attacker can modify the XML document by enveloping the XML signature element. In 2009, for example, a SOAP interface flaw was discovered in Amazon Elastic Cloud Computing (EC2) [13]. As a result of a successful signature-wrapping attack, attackers were able to change the eavesdropping message.

2.1.6 Attacks by the Man in the Cloud

By taking advantage of holes in the sync token process, hackers can intercept and modify cloud services by replacing the sync token with a new one that gives them access to the cloud during the subsequent sync. Users might not even be aware that their account has been compromised because an attacker can always restore the original sync tokens. Additionally, it's possible that accounts that have been compromised won't be recovered [14].

2.1.7 Insider attack

An insider attack is initiated by an authorized user who violates security regulations. A cloud attacker system could be commanded by the provider or an employee of the corporate client with powerful rights. To avoid this harmful behaviour, cloud providers must design secure systems with access to different levels of cloud services [15].

2.1.8 Theft of service or account

Account or service theft happens after gaining access to the user's credentials. This can be done in several methods, including hunting, spyware, and cookie poisoning. Attackers can access a user's personal information or business data when a cloud account is breached, jeopardizing cloud computing services. For instance, a 2007 phishing scheme that targeted a Salesforce employee disclosed all of the company's client accounts. [16].

2.1.9 Advanced Persistent Threats (APTs)

Without the awareness of authorized users, APT assaults provide hackers the ability to steal sensitive data stored in the cloud or to abuse cloud services. Since these attacks last so long, hackers can alter security precautions to block them. Once they have gained unauthorized access, hackers can peruse data center networks and manipulate network traffic.

2.1.10 Specter and Collapse are two new attacks.

This year saw the debut of these two types of cyberattacks, which swiftly emerged as a fresh danger to cloud computing. By taking advantage of a design weakness in the majority of modern CPUs, adversaries may access data encoded from memory using malicious JavaScript code. Attackers can access information from the kernel thanks to both Specter and Meltdown, which blur the line separating software and the operating system. Because not all cloud users have installed the most recent security upgrades, this is a significant problem for cloud developers [17].

2.2 DDOS attack in the cloud

Cloud computing is growing in both research in academia as well as industrial technology. DDoS is one of the security threats that could compromise availability. As per Cloud Security Alliance, DDoS is one of the top nine dangers to cloud computing. Cloud Security Alliance. DoS attacks represent 14 percent of all attacks within the cloud [18]. Many famous websites, including Yahoo, were hit by DDoS attacks in the early 2000s. In May 2001, grc.com was struck by a large DDoS attack [19]. The corporation relied on the internet for manufacturing, and its business suffered considerably as a result. VeriSign hired Forrester Consulting in March 2009 to research DDoS risks and security. The poll included 400 people from the United States and Europe. In their organization, 74% have been subjected to one or more DDoS assaults. Attacks cause service interruption in 74% of cases, according to 31%, and do not impact services in 43% of cases. According to a study of DoS attacks in the cloud, as the cloud's usage increases, so does the frequency of DDoS attacks. If the demand for services is increased in a cloud-based platform then it will be able to supply the required processing power to handle the added stress. In other words, the cloud system can work against the attacker while permitting the attacker to cause as much damage as is feasible when the cloud service is in place, beginning with just one assault entry point. A cloud service is made up of various services that are delivered on the same hardware servers, which may be overburdened due to flooding. Therefore that if a particular service tries to run within the exact same system that is overwhelmed the availability of that service could be affected. Another result of the flood is that cloud use fees will skyrocket. The problem is that there isn't an "upper limit" to refer to. Cloud attacks are neighbor attack, in which the cloud device is targeted by its counterpart within the same physical infrastructure and stops the other from offering services. These attacks could impact cloud performance, lead to financial losses, as well as damage to other servers that are part of the same cloud structure.

2.3 Comparison of PCA and sample entropy

Model of entropy: Collecting flow or feature traffic statistics from switches and computing randomness to assess unpredictability in packets that arrive into the network is a standard way for detecting DDoS in SDN. The entropy increases as the unpredictability increases, and vice versa. Depending on the scheme, an attack is detected by defining a threshold and observing if the entropy is above or below it. The sample's entropy is one metric that represents the degree of dispersion or concentration.

Considering one observed, the overall number of traffic is S, where NOD couples (source and target couples) exist and ni represents the amount of OD-Piri traffic. In this observation, the OD pair times will occur. The score H(X) is in the middle of the pack (0; log 2N). When the distribution is maximumly concentrated, the score will take the value 0. When the distribution is maximally scattered, the score will take the value 2N.

2) Contrast

To compare PCA and sample entropy, we continually set up the experiment and changed several settings.

DDoS risk assessment

Both the distribution and the amount of OD pairs present have an impact on the sample entropy outcome. Regular network traffic begins at interval 11. In this experiment, DDoS assaults begin at interval 152, and normal traffic doubles at interval 172. The interval is set to 1 second. In this stage, We're curious to observe how changing normal traffic impacts the result. We analyze the data in each switch using sample entropy, conventional PCA, and our PCA partitioning approach. The entropy value of the sample slightly rises when the volume of typical traffic is doubled**.**

3. RELATED WORKS

Landoll [20] mentioned in his research “The security risk assessment handbook” that, When done correctly, information security risk assessments give managers the input they need to manage risk by recognizing threats to company assets, identifying existing control weaknesses, and selecting effective measures. They can create a false sense of security when they perform poorly, allowing potential risks to develop into devastating losses of sensitive information cash, and business value. The Security Risk Assessment Handbook: A Complete Guide to Conducting Security Risk Assessments, 3rd Edition picks up [21] The book picks up where its most popular predecessors ended with detailed instructions on how to carry out an effective security risk assessment and efficiently. Additionally, it offers extensive coverage of Security risk analyses, mitigation plans and risk assessments. To conduct security risk assessments efficiently, we equip managers with the necessary information to reduce risk by identifying threats to the company's assets, identifying control weaknesses, and then deciding on the appropriate measures to counter. If they are not implemented correctly, they could create false security, allowing the potential risk to turn into a catastrophic loss of sensitive data, cash, and business value.

One-sided network communication solutions like RDMA and NVMe-over-Fabrics are increasingly becoming popular in production applications and data centers. Despite their enticing low CPU use and high performance, they create new security problems that might substantially weaken data center software systems built on top of them. At the same time, they provide one-of-a-kind chances to assist improve security. In terms of security, one-sided network communication is a double-edged sword. The purpose of this study is to provide insights into the security consequences and potential of one-sided communication. Security flaws and faults are the results of badly written software, which may be easily exploited by cyber thieves [22]. Wireless communication channels also allow remote control of vehicles as well as location tracking and audio exfiltration in the cabin and even theft. Remote exploits are possible through an array of attacks (including mechanical equipment audio players with CDs, Bluetooth as well as cellular radio). A lot of Cloud Software systems can be susceptible to security risks that modern security instruments and techniques cannot identify. This issue is a major one that requires the supervision and monitoring of the development process as well as maintaining it. Security is seen as one of the requirements that are not functional that have a significant influence on the design and development Cloud Software as a Service (SaaS). In addition there is a high percentage of different views between two different concepts of software engineering, i.e., traditional and modern, which causes an enormous challenge for the team responsible for software development to address security concerns during the implementation and maintenance levels in the SDLC. A real-world study comprised 103 actual failures that were created manually or automatically by real applications using a variety of methods of testing and presented some preliminary findings. The results of the study identified the emergence of many security weaknesses in the beginning stages in the cloud Software/Service Development Life Cycle (CSDLC)) [23]. As a result, this must be kept in mind ahead of time. Based on these findings, this study proposes a general paradigm for dealing with such security throughout the early phases of the CSDLC. This framework intends to offer an extra degree of security in the early phases of the CSDLC, as demonstrated through a case study demonstrating the framework's applicability [24].

Other researchers [25] discussed cloud computing security as an important side of our technical life. Modern data center networks have lately grown dramatically as a result of increasing demand for always-on and fast-response internet services. These networks frequently rely on commodity technology to achieve massive scalability while keeping capital expenditures to a minimum. The disadvantage is that commodity devices are prone to failure, posing a difficult task for network operators to address these failures quickly and with minimal delays to hosted services. Recent research [26] has concentrated on autonomous failure localization. However, resolving problems still necessitates substantial human interaction, resulting in a protracted failure recovery period. Unlike prior efforts, NetPilot tries to mitigate rather than fix faults as rapidly as possible. NetPilot reduces errors in the same way that operators do: by disabling or restarting suspected harmful components. By using an intelligent trial-and-error technique, NetPilot avoids the requirement to determine the specific root cause of a failure. The core that powers NetPilot includes an Impact Estimator that guards against mitigation methods that are too disruptive as well as a specific mitigation planner, which decreases the number of times it is attempted. We demonstrate how NetPilot can effectively reduce the number of major failures common in data center production networks.

Saxena et, al. [27]discussed In “Communication cost-aware resource-efficient load the balancing (carelb) platform for cloud-based data centers” Load balancing for communication-intensive applications, which allows for more effective resource use and lower power consumption. In addition, the overloaded network communications between virtual machines that are networked enhances network traffic, reduces the experience of cloud clients and impacts the overall performance. Cloud computing has evolved into an integral component in information technology (IT) which is the basis of the global digitalization. It offers a standardized set of IT resources that are constantly available, accessible at all times, and offered as a service. Cloud computing's scalability and pay-per-use advantages have pushed the entire globe toward on-demand IT services that allow for more efficient use of virtual resources. The increasing cloud computing has led to increased traffic on networks entering as well out of data centers. As per the Cisco Global Cloud Index, connectivity between devices within the data center will grow by an annual rate of 23.4 percent average annual growth rate (CAGR) by 2021 [28]. A connection cost-aware framework and a resource-efficiency load balancing framework (CARE-LB) have been created to solve these concerns, reducing connection costs and power consumption while maximizing resource utilization. Virtual machines with high convergence and dependency are purposely positioned near each other to decrease communication costs. The suggested integration of particle swarm optimization and genetic algorithm based on unrelated sorting is used to develop VM mode, which entails tailoring the PSOGA method to encode VMs as molecules as well as chromosomes. The framework's efficiency is evaluated by conducting numerous tests in a virtual data center and comparating the results with the most current techniques, like Genetic Algorithms, First-Fit, Random-Fit and Best-Fit algorithmic heuristics. Comparatively to the genetic algorithm-based load balancing framework, the CARE-LB framework increases the utilization of resources of 11%. It also lowers the power usage in the range of 4.4 percentage, cuts the cost of communication by 20.3 percent, while reducing the time to execute by 49.7 percent. The suggested CARE-LB architecture offers a viable option for implementing data-intensive applications more quickly while maximizing resource utilization and lowering power consumption. Experiments with various numbers of users were undertaken to determine the number of users for the connection cost analysis. For example, for 100 VMs, 10, 20,..., and 80 users are chosen at random and their VM requests (number and kind of VMs) are produced at random, ensuring that the overall number of VMs requested does not exceed the number of available VMs.

4. IMPLEMENTATION

Before we start the implementation, we explain the system model. Here was the following notation:
N-> the network where the DDoS attack happens
FND -> Fog Environment Nodes
S-> switches
Ln -> set of links exist between S and FN
Target Fog-Controller (TF): TFnd is The SDN controller, which serves as the central controller and is placed on the target node. During the attack, the console's resources will be used up, and the CPU will be used up more, which will slow down the system's throughput. The assault was initiated by a group of zombies using

the Attack function of the botnet BtNt, which causes bti to send infected request packets to the target TFith to manipulate the fog. There are two prerequisites for a DDoS attack in the system model are as follows:

- A data route P <fn source; fn dest> should exist between Bt and Tf. A direct or indirect path can be connected. It is direct if BN and TF I are connected by a single link (ln), and if e ln = 1, it is indirect if there are several linkages (ln > 1).
- The bots can have several zombies inside of them to carry out the attack. e, g bn = fbn1; bn2;...::; bnn, where (n > th), (this is the threshold value for the number of attackers such that the FN resources are consumed by more than 50%)

Set up environment and implementation
we set up the test environment and configure a network of devices to see how PCA performance, partitioning, and sampling entropy are in different cases.
Experiment configuration: By creating a cloud environment with all the required elements in place, the system model is established. Three layers make up the complete ecosystem. The open-source software used to create the cloud environment makes up the top layer. Own cloud is a cloud computing platform linked to the Apache web server. PHP and MySQL are prerequisites for setting up "Own cloud". Cent OS7 has been installed on the cloud server. MySQL uses "MariaDB" as its database. A popular Linux system like CentOS uses MariaDB, a derivative of MySQL. It is comprised of multiple virtual machines, comprising an SDN which is also an Apache server, on where an SDN console is created. Different forms of lawful and malicious virtual threats are present in the application layer. Different sorts of valid and malicious virtual machines installed with Linux and Windows can be found at the application layer. Through the use of random virtual machines and HPing-3, the primary attacks against the TCP, UDP, and ICMP protocols are conducted. For numerous VMs on the application layer, a topology is built using the Mininet emulator.

The fog server's "Floodlight" controller served as our choice for the SDN controller. It is a Java-based console that is licensed by Apache and used to establish communication links (Ln=Ln1, Ln2,...) between client computers and Fn-fog nodes. The console is used to create and configure the DDoS Defense module. The free source Python module "Keras," which utilizes the "Tensorflow" backend, was used to create the deep learning model. The loss function is the "binary entropy," the "adam" optimizer is utilized, and there is a leak probability of 0.2. 128 hidden neurons are employed to explore the LSTM. The output layer of the model makes use of the tanh() activation function and has two hidden layers with 128 hidden neurons that employ Sigmoid. The model may have vanishing gradient issues since it employs the sigmoid activation function. The small batch descent (GD) technique is employed in place of batch gradient descent to prevent this. In mini-batch GD, learning is restarted with a fresh small batch after iterating to a fixed small batch size, lowering the likelihood of gradient burst. The number of iterations for this mini-batch is set at 512. Running on Windows 10 is an Anaconda distribution, which houses the Python execution environment.

A program has been prepared for the Mininet test environment to create a small unit of a ring topology network of three switches and 11 nodes, which can be another network or a station that is directly connected to the switch. There are 11 hosts in this network, so there can be 11 parent nodes and 11 destination nodes, although the parent node cannot be the same as the destination node.
For each pair (fn source,fn dest) we can represent the following: (fn source ,fn dest),o = 1,2,…….,11 and fn source ≠ fn dest
Figure 1 depicts Long Short-Term Memory (LSTM) networks, A type of neural network that is able to learn order dependence in cases that require sequence prediction. This feature is vital for solving complex issues like speech recognition and machine translation and many more. One area that is challenging in deep learning is the LSTMs. The understanding of LSTMs along with how concepts such as sequential and bidirectional are related to the field may be complicated.

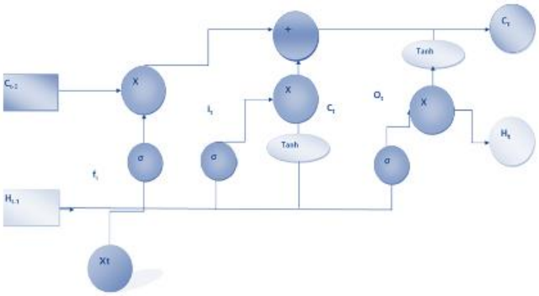

Figure 1: Long Short-Term Memory Neural Network Model Architecture

4.1 Analysis

Training a DDoS Defense Model

It is the Hogzilla Dataset is used during this investigation to test and train the proposed model for a deep-learning-based DDoS defense module. This data set pulls data of ISCX 2012's ISCX 2012 IDS and the CTU-13 Botnet. [29]. Each stream in this data has behavioral traits in it. The ISCX 2012 IDS dataset comprises data related to typical packets, whereas the CTU-13 botnet dataset covers all aspects connected to attacking packets. Three different fields are included in the dataset. Three different fields are included in the dataset.They are logical, numerical, and deterministic. The use of a single hot coding approach yields binary strings as the representation of categorical variables [30]. Each category characteristic is transformed into 16-bit equivalent binary strings in this technique. The specifics of the associated characters are displayed in Table 1.
The deep learning model is running on the CTU-13 Bot et al. [31] and ISCX 2012 IDS datasets generated the findings shown in this section. The proportion of the training sample to the test sample is 90:10. This indicates that 90% of the total data sample is utilized for training, while the remaining 10% is used for both validation and testing. To verify the results for validity, we applied a technique of 10-cross-validation. The totality of data samples is split equally into 10 sections nine of the sections are used as samples for training and the remaining portion is used to test. The

process is repeated ten times before the final outcome is calculated by averaging all of the repetitions. In addition, efforts are to modify the formula's requirements. The model was tested using one hidden layer as well as two layers of hidden layer and also by changing how many hidden nodes, first from 32 nodes to 64 before moving up from 128 up to 128 of them. Similar to that, the probability of leakage was initially put at 0.1 and later it was determined as 0.2. The visible and the hidden neural networks' modules are removed temporarily, together with their inbound and outgoing connections, the dropout probability is used to stop the issue of overfitting as well as quick response within the neural network that is recurrent. [32]. Initially, the network was trained using dropouts from zero. Then, it was tested with 0.1 to 0.3. The model, however, is set to 0.2. The two hidden layers and 0.2 leakage rate of the LSTM model are effective. By examining the outcomes after modifying the parameters and running the trials again, the model's parameters are fine-tuned. The accuracy % concerning the training and test examples is shown in Figure 3. Figure 3 creates a comparison graph of the error rate of the LSTM variables using only the test data. Figure 4 displays the test data correctness of the LSTM variables. As you can see, LSTM2.2 is more effective in terms of precision than the other. The test data set performance was similarly impressive with a score of 98.88 percent, and the performance on this learning dataset was around 99.12 percent.

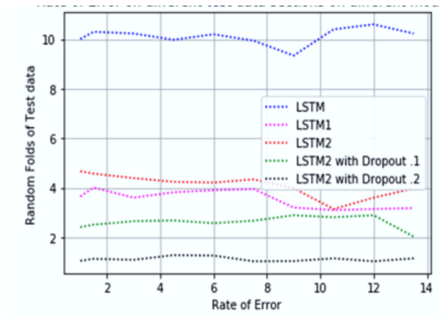

Figure 2: Random Folds of Test Data

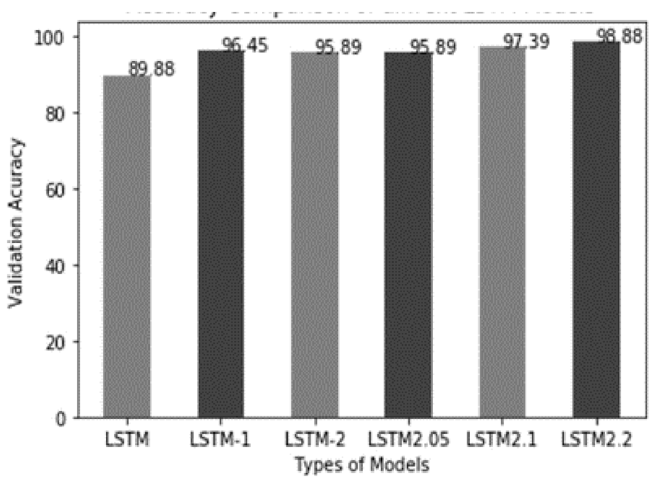

Figure 3: Test validation

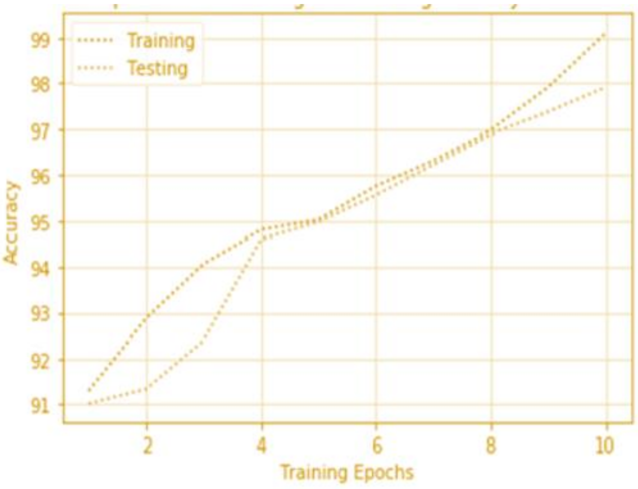

Figure 4: Graphic representation of test

Testing the DDoS Defense Model

A model created using two different forms of data is tested. 10% of the total data set is used as test samples in the initial testing of the model on the Hogzilla data set. Additionally, several actual DDoS attacks are put into practise, and a test environment is developed to verify the model. TCPDump is used to extract network traffic. TCPDump is a programme that automatically tracks network statistics. Using a programme called Hping3, which separates malicious from benign data, a DDoS assault simulation is performed. The open source Hping3 utility is used by some random virtual machines to launch DDoS attacks against the TCP, UDP, and ICMP protocols. After applying a deep-learning model to these threatening packets Results are presented in Table 2. The performance measurements are based on percentage precision over the test data for all LSTM variables, with the possibility of a variety of hidden neural networks. In addition, LSTMs are tested with leakage and non-leaking potentials in addition to 0.1 as well as 0.2 leakage probabilities. As can be seen in the diagram, the LSTM model is comprised of three layers hidden and 128 input nodes and leakage rates of 0.2 which is lower than other models. Table 1 contrasts the DDoS Defense model with other models that have previously used DL and SDN. As can be observed, the test data show some encouraging outcomes for LSTM 2.2.

Table 1: Presenting test results with and without layers

Model Type	**LSTM with no hidden layers**	**LSTM with 1 hidden layers (LSTM-1)**	**LSTM with 2 hidden layers (LSTM-2)**	**LSTM with 3 hidden layers (LSTM-3)**
Activation Function	**Sigmoid, Tanh**	**Sigmoid, Tanh**	**Sigmoid, Tanh**	**Sigmoid, Tanh**
No hidden neurons = 128	89.88	96.45	95.89	97.21
No hidden neurons = 64	87.96	96.98	95.67	97.45
No hidden neurons = 32	87.67	91.33	94.68	93.67
Dropout = 0.2	90.98	92.89	98.88	98.34
Dropout = 0.1	90.13	91.57	97.39	96.78
Dropout = 0.0	89.88	96.45	95.89	93.29

Model Type	***Training Accuracy***	***Testing Accuracy***	***Dataset Used***	***Used in Cloud and Fog***
Stacked Auto Encoder	NA	95.65	Captured Data	No
LSTM	99.00	98.00	ISCX 2012	No
LSTM-2 Dropout = 0.2	99.48	98.88	ISCX 2012, Real Data	Yes

5. RESULTS

1. Training a DDoS Defense Model

DDoS defense model test

The model that was developed is evaluated with two types of sources. The model is initially tested using the Hogzilla dataset which includes 10percent of the dataset being used as test samples. In addition, numerous real DDoS attacks are executed in the test bed is built to verify the idea. TCP Dump is used to extract network traffic. TCP Dump is an automatic network statistics monitoring tool. To mimic a DDoS assault, a program called Hping3 is employed, which collects both harmful and non-malicious data independently. DDoS assaults against TCP, UDP, and ICMP protocols are carried out using the free source program Hping3 and several random virtual machines. LSTMs are evaluated using and without dropout probabilities between 0.1 or 0.2. It can be seen how the LSTM model that includes three layers hidden with an input total of 128 nodes and a drop-out probability of 0.2 is superior to the other models. As you can see, LSTM 2.2 provides some fascinating results from test data.
Information about the attributes of the dataset that are used in the proposed model.

Table 2 : Details of the dataset attributes used in the proposed model.

Total No. of fields	192	No. of classes	3
No. of Categorical fields	4	No. of bits required to represent categorical fields	16
No. of Numerical field	9	No. of bits required to represent numerical fields	Nil
No. of Boolean fields	179	No. of bits required to represent Boolean fields	2

Table 2 represents the dataset attributes used in the proposed model. The table shows that there are 192 total fields and 3 classes, 4 categorical fields, and 9 numerical fields in the proposed model. We need 2 Boolean fields in the proposed model as we have 179 present.

We are continually setting up the experiment and adjusting specific settings to compare the PCA and the entropy sample.

We aim to investigate the following points:

(1) if the amount of regular traffic affects the outcome, (2) whether this scheme detects when a DDoS assault ceases, and (3) introducing a mutant DDoS attack targeting the SDN and determining whether this system is capable of detecting this attack.

DDoS evaluation the sample entropy value is determined not only by the distribution but also by the number of OD pairs that emerged.

The interval in this experiment is set to 1 second, regular network traffic begins at interval 11, DDoS assaults begin at interval 152, and the quantity of normal traffic doubles at interval 172.

We're interested in how modifying the usual traffic will impact the outcome. We examine the data using sample entropy, classic PCA plot, and our PCA partitioning technique on each switch, and the results show that DDoS attacks have captured sample entropy, PCA, and PCA partitioning. The sample entropy value significantly rose when the typical traffic volume doubled. While this is going on, the results of the PCA plot and PCA split plot are separated with the regular interval and are unaffected by the typical multiplicative traffic.

A DDoS assault may be identified by the fact that the destination IP address is static and the entropy falls when a DDoS attack takes place, according to an evaluation of the Entropy model of a new form of DDoS attack. However, we can easily use a DDoS assault to give the target IP address at random, preventing the entropy sample from detecting such attacks. Therefore, in this experiment, we begin regular traffic at interval 8 and a modified DDoS assault at interval 43.

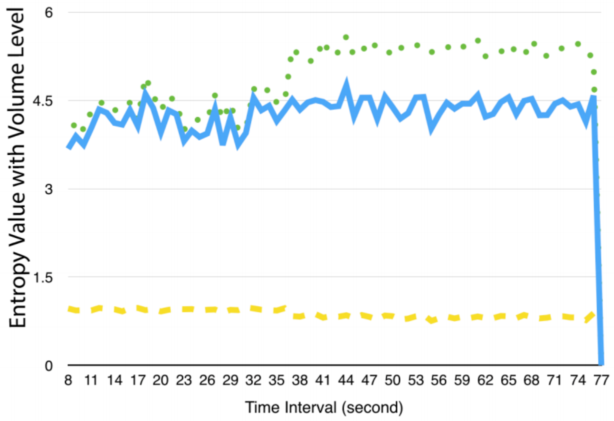

Figure 5: Changes in Volume Entropy with time Interval

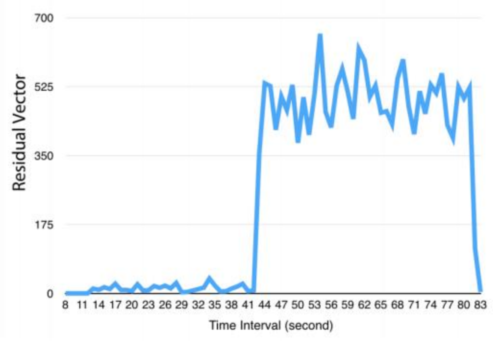

Figure 6: Changes in Residual Vector concerning time interval

Entropy will fluctuate as a result of changes in volume. There are more ways to distribute molecules in a bigger volume, and the more distribution options there are, the higher the entropy. The entropy will rise as the volume rises.

The figure above shows one independent variable on the horizontal, and that of the remaining variables on the vertical. The linear model can be used for the data if the points on the residual plot are distributed randomly across the horizontal axis. In other cases, a nonlinear model is more suitable.

6. CONCLUSION

To identify DDoS attacks on the cloud computing environment, we present a DDoS strategy employing principal component analysis. Also, we put it to the test and compared it against a widely utilized approach called sample entropy. We demonstrate that this technique has clearer results than the alternative. In the meanwhile, we discover a unique DDoS assault targeting Cloud Network environments that might do greater harm to cloud networks. We tested this novel attack using the two detection methods and discovered that it is difficult to be identified by sample entropy while still being spotted by PCA.

REFERENCES

[1] Duncan, A., Creese, S., & Goldsmith, M. (2015). An overview of insider attacks in cloud computing. Concurrency and Computation: Practice and Experience, 27(12), 2964-2981.

[2] Goyal, D., & Rajput, R. S. (2020). Cloud Computing and Security. In The Evolution of Business in the Cyber Age (pp. 293-319). Apple Academic Press.

[3] Tabrizchi, H., & Kuchaki Rafsanjani, M. (2020). A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing, 76(12), 9493-9532.

[4] Amara, N., Zhiqui, H., & Ali, A. (2017, October). Cloud computing security threats and attacks with their mitigation techniques. In 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 244-251). IEEE.

[5] Amara, N., Zhiqui, H., & Ali, A. (2017, October). Cloud computing security threats and attacks with their mitigation techniques. In 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 244-251). IEEE.

[6] Tank, D., Aggarwal, A., & Chaubey, N. (2019). Virtualization vulnerabilities, security issues, and solutions: a critical study and comparison. International Journal of Information Technology, 1-16.

[7] Barcomb, K. E. (2011). Taking the High Ground: A Case for Department of Defense Application of Public Cloud Computing. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT.

[8] Chou, T. S. (2013). Security threats on cloud computing vulnerabilities. International Journal of Computer Science & Information Technology, 5(3), 79.

[9] Balobaid, A., Alawad, W., & Aljasim, H. (2016, December). A study on the impacts of DoS and DDoS attacks on cloud and mitigation techniques. In 2016 International Conference on Computing, Analytics and Security Trends (CAST) (pp. 416-421). IEEE.

[10] Ahmad, W., Rasool, A., Javed, A. R., Baker, T., & Jalil, Z. (2021). Cyber security in iot-based cloud computing: A comprehensive survey. Electronics, 11(1), 16.

[11] Singh, S., Jeong, Y. S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. Journal of Network and Computer Applications, 75, 200-222.

[12] Gruschka, N., & Iacono, L. L. (2009, July). Vulnerable cloud: Soap message security validation revisited. In 2009 IEEE International Conference on Web Services (pp. 625-631). IEEE.

[13] Danezis, G., Domingo-Ferrer, J., Hansen, M., Hoepman, J. H., Metayer, D. L., Tirtea, R., & Schiffner, S. (2015). Privacy and data protection by design-from policy to engineering. arXiv preprint arXiv:1501.03726.

[14] Iqbal, S., Kiah, M. L. M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R. (2016). On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. Journal of Network and Computer Applications, 74, 98-120.

[15] Borazjani, P. N. (2017, May). Security issues in cloud computing. In International conference on green, pervasive, and cloud computing (pp. 800-811). Springer, Cham.

[16] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless computing: Current trends and open problems. In Research advances in cloud computing (pp. 1-20). Springer, Singapore.

[17] Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE communications surveys & tutorials, 18(1), 602-622.

[18] Arora, K., Kumar, K., & Sachdeva, M. (2011). Impact analysis of recent DDoS attacks. International Journal on Computer Science and Engineering, 3(2), 877-883.

[19] Landoll, D. (2021). The security risk assessment handbook: A complete guide for performing security risk assessments. CRC Press.

[20] Hanson, R. K., & Thornton, D. (2000). Improving risk assessments for sex offenders: A comparison of three actuarial scales. Law and Human behavior, 24(1), 119-136.

[21] Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., ... & Kohno, T. (2011). Comprehensive experimental analyses of automotive attack surfaces. In 20th USENIX security symposium (USENIX Security 11).

[22] Le Goues, C., Nguyen, T., Forrest, S., & Weimer, W. (2011). Genprog: A generic method for automatic software repair. Ieee transactions on software engineering, 38(1), 54-72.

[23] Aljawarneh, S. A., Alawneh, A., & Jaradat, R. (2017). Cloud security engineering: Early stages of SDLC. Future Generation Computer Systems, 74, 385-392.

[25] Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., & Inácio, P. R. (2014). Security issues in cloud environments: a survey. International Journal of Information Security, 13(2), 113-170.

[26] Hancock, P. A., Nourbakhsh, I., & Stewart, J. (2019). On the future of transportation in an era of automated and autonomous vehicles. Proceedings of the National Academy of Sciences, 116(16), 7684-7691.

[27] Saxena, D., & Singh, A. K. (2021). Communication cost aware resource efficient load balancing (carelb) framework for cloud datacenter. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 14(9), 2920-2933.

[28] Saxena, D., & Singh, A. K. (2020). Communication cost aware resource efficient load balancing (care-lb) framework for cloud datacenter. Recent Advances in Computer Science and Communications, 12, 1-00.

[29] Priyadarshini, R., & Barik, R. K. (2019). A deep learning based intelligent framework to mitigate DDoS attack in fog environment. Journal of King Saud University-Computer and Information Sciences.

[30] Goyal, R. (2018). Building a Machine Learning Based Recommendation Engine for the Virtual Academic Advisor System (Doctoral dissertation).

[31] Pektaş, A., & Acarman, T. (2019). Deep learning to detect botnet via network flow summaries. Neural Computing and Applications, 31(11), 8021-8033.

[32] Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003, August). Best practices for convolutional neural networks applied to visual document analysis. In Icdar (Vol. 3, No. 2003).