


Article

An Adaptive Distributed Denial of Service Attack Prevention Technique in a Distributed Environment

Basheer Riskhan ¹, Halawati Abd Jalil Safuan ¹, Khalid Hussain ^{1,*}, Asma Abbas Hassan Elnour ², Abdelzahir Abdelmaboud ³ , Fazlullah Khan ⁴ and Mahwish Kundi ⁴

¹ School of Computing and Informatics, Albukhary International University, Alor Setar 05200, Keddah, Malaysia; b.riskhan@aiu.edu.my (B.R.); halawati@aiu.edu.my (H.A.J.S.)

² Computer Science Department, Community College-Girls Section, King Khalid University, Abha 62529, Muhayel Aseer, Saudi Arabia; aaselnour@kku.edu.sa

³ Department of Information Systems, King Khalid University, Abha 61913, Muhayel Aseer, Saudi Arabia; aelnour@kku.edu.sa

⁴ Computer Science Department, Abdul Wali Khan University, Mardan 23200, Pakistan; fazlullah.mcs@gmail.com (F.K.); mkundi@awkum.edu.pk (M.K.)

* Correspondence: khalid.hussain@aiu.edu.my

Abstract: Cyberattacks in the modern world are sophisticated and can be undetected in a dispersed setting. In a distributed setting, DoS and DDoS attacks cause resource unavailability. This has motivated the scientific community to suggest effective approaches in distributed contexts as a means of mitigating such attacks. Syn Flood is the most common sort of DDoS assault, up from 76% to 81% in Q2, according to Kaspersky's Q3 report. Direct and indirect approaches are also available for launching DDoS attacks. While in a DDoS attack, controlled traffic is transmitted indirectly through zombies to reflectors to compromise the target host, in a direct attack, controlled traffic is sent directly to zombies in order to assault the victim host. Reflectors are uncompromised systems that only send replies in response to a request. To mitigate such assaults, traffic shaping and pushback methods are utilised. The SYN Flood Attack Detection and Mitigation Technique (SFaDMT) is an adaptive heuristic-based method we employ to identify DDoS SYN flood assaults. This study suggested an effective strategy to identify and resist the SYN assault. A decision support mechanism served as the foundation for the suggested (SFaDMT) approach. The suggested model was simulated, analysed, and compared to the most recent method using the OMNET simulator. The outcome demonstrates how the suggested fix improved detection.

Keywords: DDoS attack; SYN attack; attack mitigation; security



Citation: Riskhan, B.; Safuan, H.A.J.; Hussain, K.; Elnour, A.A.H.; Abdelmaboud, A.; Khan, F.; Kundi, M. An Adaptive Distributed Denial of Service Attack Prevention Technique in a Distributed Environment. *Sensors* **2023**, *23*, 6574. <https://doi.org/10.3390/s23146574>

Academic Editor: Mikael Gidlund

Received: 2 May 2023

Revised: 19 June 2023

Accepted: 22 June 2023

Published: 21 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction and Background

1.1. Introduction

The introduction of the attack through machines is known as “Supplementary Victims”, while the in-attack routing protocols are “Main Victims”. In this case, tracking the attacker is becoming difficult taking into account legitimate customers.

Network security has become a complex challenge for companies with a data centre or network configuration. Various hardware and software resources are used to unencrypt passwords from several attacks. The push-back procedure is widely used to support distributed denial-of-service attacks. DDoS activities are viewed as a traffic control issue using a router, even though the deceptive controller causes disruption and therefore does not manage congestion after the conventional edge. The newest generation of the router is sensitive enough to detect and lower suspicious packets. Onshore routers have been informed of the decrease in suspicious packets and advised to use router services for legitimate content instead. A further unique and reliable countermeasure for handling DDoS attacks is the implementation of user riddles [1,2]. Throughout this strategy, the

victim system challenges a system that sends traffic to recognise the attacker. If the client solves the mystery, the traffic is viewed as valid and thus expected to move to the client; unless most of these people solve the riddle, the difficulty of the riddle is enhanced. Such a strategy ensures the continuous flow of network traffic across the intermediary routers before it reaches its destination (Figure 1) [3].

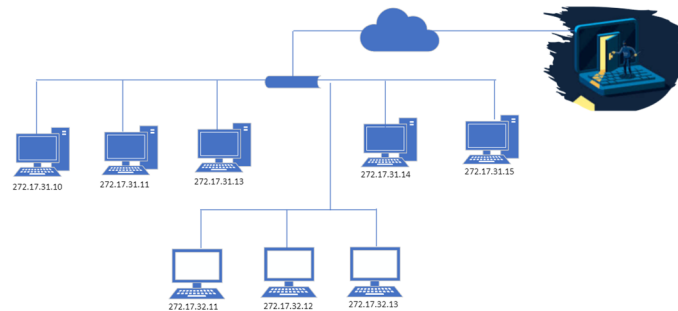


Figure 1. DDoS attack scenario.

A single-layer rational system is a circular base feature system used to diagnose irregularities and classify regular traffic. These proposals add ingenuity to Denial-of-Service prevention, although information-based positioning is often used. The simple advancement of signature-type attack identification is commonly used when inbound traffic is likened to accessible, recognised strikes called white-list patterns (information). It efficiently detects potential threats throughout the signature data set [4].

One strategy for detecting DDoS is attitude-based identification, which can distinguish DDoS-attack traffic from sanctioned traffic irrespective of different ways of attacking content and techniques. Currently, DDoS strikes are conducted using tools, worms, and botnets to victimise entirely different packet transmission rates and packet aspects of the defence strategy [5,6].

As a result, these different types of attacks contribute to defence systems requiring other encryption methods on-site. DDoS attacks aim to render traffic unavailable, including Flash crowd cases. The findings of experimentations with several databases and tests suggest that the predicted techniques can separate DDoS threats from lawful traffic [7]. Denial of Service has emerged as a major threat to several companies nationwide. DoS attacks are resolved through the series number encryption technique, and the hop sequence filtration approach efficiently filters attack packets, providing the database with appropriate security [8].

To secure two-layer protection strategy resources, it is suggested that the MAC generator be isolated legally from the encrypted one, through which the client services are distributed to legitimate lanes and lawful customers efficiently [9,10]. Traditional methods for detecting a distributed denial of service (DDoS) attack have also been unsuccessful. Throughout this journal, artificial neural systems and clustering algorithms have been suggested for a new compact tracking strategy. At the same time, the ANN Multi-layer Perceptron has been used to enhance conviction rate and precision [6,11]. The outcome of the whole analysis is influential compared to earlier studies and a fantastic way forward towards future research [12]. The aim is to identify and prevent specific DDoS attack trends and strategies from occurring in a decentralised setting. It is a remedy for the identification and mitigation process, wherein the SFaDMT methodology works in a single-node activity. SFaDMT can be used efficiently to identify sequence recognition and signatures that already occur throughout the SFaDMT system [13–15].

Once a DDoS intrusion is performed on a system, the application of resources to potential users cannot be successfully achieved. In order to address this problem, it is suggested that DDoS identification, as well as prevention techniques, be referred to as SFaDMT. The whole strategy describes the SYN Flood attack on the system and minimises it to execute streamlined behaviour for the system [10,16,17].

One crucial method for stopping cyberattacks is intrusion detection, which may be divided into three categories: hybrid detection, misuse detection, and anomaly detection. For example, anomaly detection uses network data and connection traffic to find threats and typical access behaviours. However, traditional behaviour identification-based anomaly detection is unable to meet the demands due to the large-scale, dispersed, and non-standard physical components present in ICPS and IIoT.

The heavy computational burden of cloud data centres and the monitoring of anomalous access to physical units with set communication cycles are two technological issues that require attention from a federated learning technique that decentralises the detection work to the edge, considering the former [18]. Knowing how a cyberattack is designed is the most crucial factor in a CPS's security. Knowing the structure of such a cyberattack is a crucial component of a successful mitigation plan for the security of CPS.

A variety of cyber-attacks were developed against CPS components to explore this, and the impact on cyber, physical, and collaborative control components was assessed. Stuxnet [16] and the Aurora assault [19] raised awareness of and sparked widespread worry about cyberattacks that may harm physical infrastructure. As previously said, since most current security measures were created for cyber-only systems, they cannot be easily applied to CPS in a collaborative network. New strategies are required to stop CPS failure. The interface is a crucial node where cyber components enable a wide range of assaults due to the differences in the physical and cyber layers' features inside CPS. The PC, in comparison, is rigid and straightforward, with very few attack alternatives [20].

1.2. Literature Review

Pushback is a strategy used to defend against DDoS attacks. DDoS attacks are mostly successful because traffic can be carried out with malware hosting in the decentralised system, and end-to-end traffic management cannot be conducted and can be managed by a function in the new router. The packets related to the intrusion must be identified but most likely contributed to the strike [21–23].

To complete just the lawful traffic's progress upward, routers will inform of the cancellation of the deceptive traffic. In certain cases, the user question has been used as a common strategy for the past few years to help alleviate the DDoS attack [21]. The target system assigns a riddle to the end user to define and discriminate between legitimate and deceptive traffic. If the user effectively solves the riddle, it is presumed that the user is a legal end-user, and permission to access the database will also be given. Unless the highest possible number of clients can overcome the riddle, the system may increase the difficulty of the riddles. When it hits the end state, it is a crossroads for malicious information [24].

The strategy for detecting DDoS using actions-based identification can distinguish between distributed denial-of-service (DDoS) traffic and legal traffic, irrespective of the various types of intrusion transmissions, including techniques [14,25,26]. Today, DDoS attacks use software, worms, and botnets to victimise entirely different transmission rates and packet types to defeat defensive systems. Accordingly, these different types of attacks contribute to protection systems offering other detection systems for ground attacks. DDoS attacks go through traffic like Flash population cases [27].

DDoS attacks include options for reproducible variations that unite the area separately from the normal crowd flow of traffic. In this journal, similar detection approaches have been used to endorse Pearson's statistics. Techniques can derive reproducible options from packet deliveries within the DDoS traffic, not from quick crowd congestion. Comprehensive models have been conducted to enhance detection systems [22].

The results of the experimentation have been shown regarding many databases, and our findings support the predicted techniques by which DDoS attempts could be distinguished from legal traffic [23,28]. Denial-of-Service attempts are a significant downside for the tech community, given that the research group has also developed a comprehensive scope of security strategies.

Throughout this journal, we aim to implement information technology's rapidly hopping, easily remotely operated, and efficient channel-layer architecture against DDoS attacks. Our solution provides a clear method for potential buyers to protect the functionality and target database of the correspondence activities. We tend to describe the Dynamic Database Server Address Alteration technique, but each component implements the approach [12].

DDoS flood-based packet strikes are a very common technique and are successful against the accessibility of facilities and apps on the system. They are quite hard to detect and avoid due to the decentralised framework. The new technology addressed throughout this journal is Stop-It. Throughout this methodology, combative processes premised on filters are prepared to prevent attacks from happening. Big DDoS floods are centred on assaults. Nevertheless, this could be unsuccessful unless the concentration connection is communicated to the survivor. The journal shows a clear variation of the Vary system within the Stop-It methodology. Directly and indirectly, attacks can be controlled to minimise DDoS attacks [13].

Throughout this journal, the author points out how GET Flood's interaction mechanism is incorporated into distributed denial-of-service attacks for rapid attack identification in a decentralised setting. By contrast, interval simulations are performed to align efficiency with the trend identification of attack alternatives and Snort identification of approved communications protocol stream trends, including log data from a network server. Experimental data indicate that the proposed strategy is safer than the identification of Snort because the previous period was smaller for that traffic. Furthermore, the whole strategy will ensure the ability of the target computer to be associated with the preventative and dependable identification of endorsed information and communication procedures [14].

DDoS strikes send large amounts of network traffic to the target system through the victimhood of various systems. Flow-based object detection strategies have performed significantly better than fingerprint-based attack identification techniques in these tests. Flow-focused DDoS attack identification methods were separated into two classes, i.e., packet-header-predicated and numerical-implementation-based. In that job, the goal is to examine each computational principle to investigate the DDoS attack mechanism and to maintain false pros and cons focused on problematic control bench victimhood advanced systems.

The journal has also been evaluated and tested in terms of precision, including the ability to perceive, and its development is recommended to produce even better outcomes than the two algorithms initially proposed as different strategies:

- Signers based;
- Anomaly related;
- DNS related;
- Mining cantered.

A comparison, including an examination of the benefits and drawbacks of the approaches alluded to here, can be made. Throughout the current situation, however, no one discussed the issue of why it is hard to detect current botnets and how we might utilise fluxing strategies to detect them. Throughout this research, two more sophisticated botnet-level strategies are mentioned: Fast-Flux-Single-Flux and Double Flux-Domain-Flux-Torpig (FFSN), which passive and active strategies could identify.

First, the author suggested a DNS-based RDNS monitoring strategy for detecting unauthorised flux system networks throughout this journal. Second, the flux agent surveillance system consists of four elements. To obtain information and add new IPs to the IP track repository, a new technique was created throughout the title of the Dig-Tool; the key element was the tracking agent, which delivers the HTTP server to the IP track repository, and that same reaction is reported. The final aspect is an IP lifetime records server for recording the system's condition, i.e., "1" for the system being available, whereas "0" is for the service not being accessible.

2. Proposed Solution and Methodology (SFaDMT)

The Internet Protocol is a cloud-interface communications server that is a delivery service for packets. UDP is not that efficient, which implies that perhaps the distribution of packages is not guaranteed. However, network-less implies that certain packets could keep their own records and be independent of transmissions. The Transmitting Control Interface can be improved with a one-sided network layer and the Protocol on another. Accurate contact between applications and different networks is assured. TCP allows the efficient transmission of data streams in sequence with no duplication or malfunction.

2.1. Establishment of TCP Connection—(Three-Sided Shake of TCP)

TCP connectivity is formed with a three-sided handshake. First, the user delivers link queries by submitting the SYN message for the host; the host acknowledges this by sending the SYN to acknowledge packets back to the recipient node, which also distributes the contact space throughout the queue. Eventually, the user accepts the ACK packets and the communication phase is finished.

2.2. The Technique for DoS/DDoS Outbreak (TCP-SYN Flooding Intrusion)

A TCP (Transmission-Control Protocol)-synchronised flood intrusion is a harmful DDoS/DoS attack initiated by an assailant via several connections. In these connections, SYN-ACK and SYN packages are swapped frequently, resulting in a shortage of ACK messages that are not sent to the server. As a result, the system leaves demilitarised space dedicated to all unfinished links, so there is no space available for anti-malicious link queries that prevent end-users from using the survivor scheme or channel. The SYN Flood strike is based on a three-way sequence of technical handshakes that starts transmitting control procedure associations. Throughout this grouping, the third package indicates the initiator's capacity to retrieve packets at the IP address and its initial message, which utilises the origin or restores the retrieval capability. Figure 2 describes the start of a standard TCP link transmitted at the packaging chain. Link details are preserved by a collection of mechanisms throughout the operating system and the transmission mechanism code framework in the Transmission Controls Box (TCB).

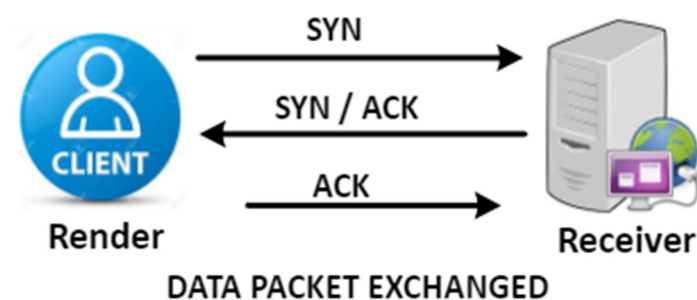


Figure 2. Three-way handshake of TCP connection.

The TCB storage capacity depends on the TCP settings, the functionality offered by the configuration, and the communication allowed. The Transfer Control Box had 282 bytes and 1300 bytes throughout the new OS. The transmitting control procedure's collected status synchronisation suggests that the contacts are just half-open. At the same time, the validity of the demands remains a question to be answered. The main point would be that the transmission-controlled box is distributed based on the synchronising package obtained before the link is created or the initiator's accessibility is verified.

It is a good indication of the denial of service that the receiving synchronisation is now the distribution that triggers the distribution of the transmission controls procedure. This can also deplete the capacity of the server processor. The aim of the Transfer Control Policy flood synchronisation intervention is to reduce the delay via the synchronisation sections, which occupies the full bottleneck. Link-encrypted communication domains are

used by synchronisation assailants and do not trigger any reaction to line the Transmitting Control Box when using the obtained synchronisation classification. The transmitting Control Procedure aims to be credible, so servers keep their Transfer Control Blocks in synchronisation for a longer period until the two-thirds connection has been released. Meanwhile, the network is denied entry demands for legally permitted transmission control procedure connections.

Figure 3 specifies a set-up of synchronization inundating attacks and provides a general clue for such interventions.

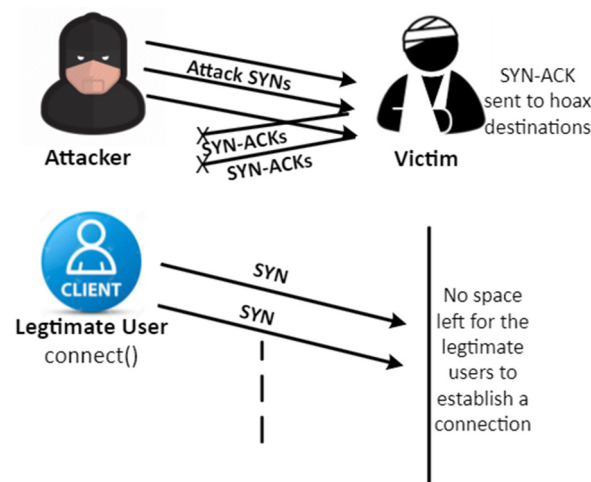


Figure 3. Denial of request to legitimate user.

Figure 4 shows some dissimilarities which have been observed.

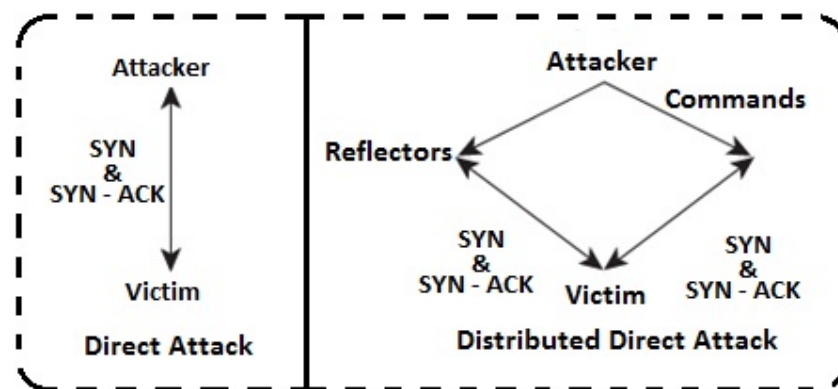


Figure 4. Types of basic DDoS attack.

3. Proposed Technique (SFaDMT)

SFaDMT will filter the SYN packets from incoming traffic throughout the projected strategy, and rules are implemented in the SFaDMT to detect the DDoS attack signatures.

3.1. Flowchart

Once traffic comes up, the signatures are compared to the current repository. If paired, the traffic has been considered malignant and would be obstructed from obtaining access to the system. If the traffic sequence does not suit the signatures still in existence in the computer system, it will be entered in the SFaDMT, where a contrast is made between the current signatures and the traffic that comes in the SFaDMT. If the signatures have a correspondence here between two traffics of more than 71%, the inspection of the system will be allowed and will be deemed harmful. If the correspondence is lower than 69%, this will be viewed as a legal flow of traffic and will be able to obtain access the network

effectively. An adapted solution for the identification of DDoS intrusions in a centralised setting (Figure 5):

- Application of a multi-deposit preceptor deposit in a centralized environment.
- Pattern and signature-centred strategies are used to identify DDoS intrusion.

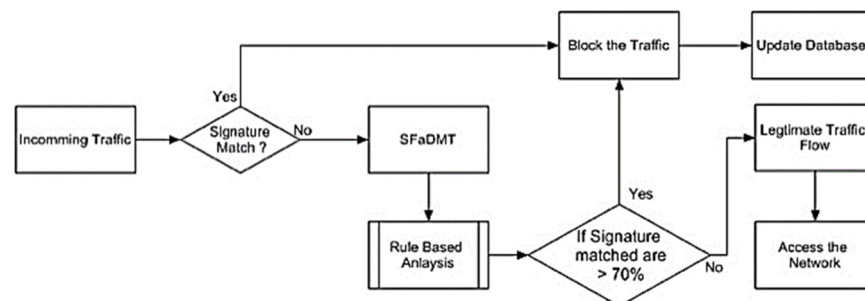


Figure 5. Flow chart of SFaDMT technique.

3.2. The Model of SFaDMT

An inspection was carried out on the traffic coming to extract SYN messages. Subsequently, as in the suggested solution, the signatures could be compared with the traffic to decide whether the traffic is authentic or deceptive. When unauthorised communication has been identified, the SFaDMT will be informed and traffic will be stopped from entering the system. It will alert, review the repository, and create logs at any time, but if the exact type of attack happens on the system, it will be identified after the contrast. If there are fingerprints, if it does not meet the SFaDMT fingerprints or if it has less than 71% correspondence, it will be regarded as legitimate traffic. This will upgrade the SFaDMT and allow you to obtain a connection effectively. In the context of traffic, the behaviour is unclear. A layer packet examination identifies traffic and determines whether it is deceptive or legal. The channel would be reached whether the transmission is legal or not, while the deceptive traffic would be restricted (Figure 6).

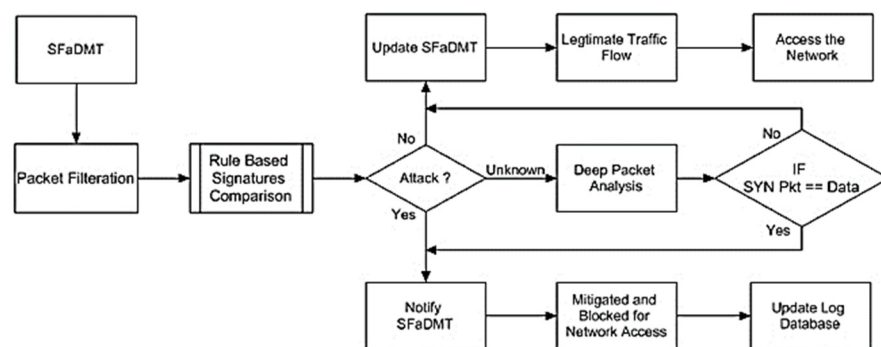


Figure 6. Framework of SFaDMT technique.

3.3. Proposed Solution and Methodology

The recommended strategy introduces a method when the transmission collects at the channel portal, as seen in the illustration described. Afterwards, the recommended strategy protects it from disruptive activity. Malignant traffic may be distinguished through signature-centred detection. A multi-layer strategy is used throughout this method to suit the identities currently existing on the server. If the signature indicates some similarities, a detailed description will be carried out in the SFaDMT system, and even if the outcome suits 71% of the accessible signatures, the fraudulent traffic will be mitigated and isolated from the system (Figure 7).

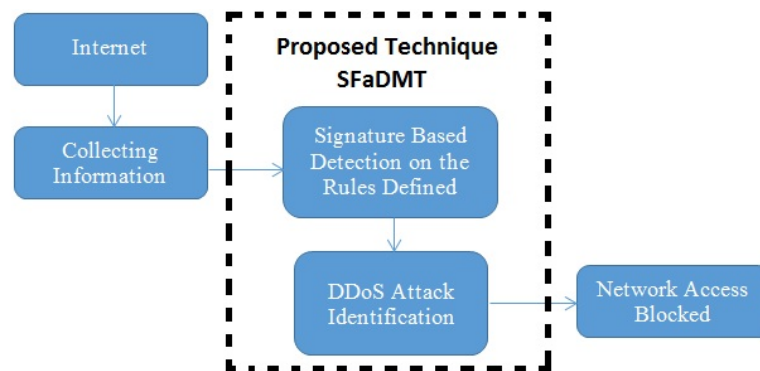


Figure 7. Proposed SFaDMT solution for detection of DDoS attacks.

3.4. Guidelines for the Categorization of SYN Flood Malevolent Instructions/Apps

In the proposed technique, a hybrid solution is presented, as shown in the above image. We have deployed a solution that will protect the packet from malicious traffic as it arrives at the network gateway. The malicious traffic can be isolated using signature-based detection. In this solution, a multi-layered approach was used in which the signatures were matched with the already-present signatures in the database.

A detailed analysis will be performed in the SFaDMT module if the signature is similar. If the result matches 70% of the signatures available, it will mitigate that malicious traffic and isolate it from the network. The filtration rules of SYN flood packet are described below, whereas Algorithm 1 is presented the procedure to detect half-open TCP connection, Algorithm 2 presented how to detect malicious traffic based on signatures in already archived attacks signatures and finally Algorithm 3 the mitigation mechanism is described the working of SFaDMT.

3.5. Rules for the Filtration of SYN Flood Malicious Packets

Rule 1 \rightarrow {SRC_IP \neq DST_IP}

Rule 1 allows only packets with different source IP addresses and destination IP addresses. However, it will consider attack traffic if it finds the same source and destination IP address.

Rule 2 \rightarrow {SRC_PORT \neq DST_PORT}

Rule 2 allows only TCP packets under the condition that the source port must not be equal to the destination port. This is because it will be treated as attack traffic if it finds the same source and destination port.

Rule 3 \rightarrow {tcp.FLAG = SYN}

Rule 3 investigates the TCP packets that are the SYN packets for further analysis by SFaDMT.

Algorithm 1. Check for Half Open TCP Connection.

```

while
read present connection;
if (connection attempt is not successful ||
TCP connection is not synchronized at both ends ||
TCP connection is aborted || connection cannot be closed)
then
the TCP connection is malicious;
else
the TCP connection is legitimate;
end
end
  
```

Algorithm 2. The following algorithm implements the above rules and detects malicious traffic based on signatures from the already-saved attack signature database.

Input: **Packet Pkt**
Output: **Generates logs when DDoS Attacks are performed**
while (true)
 if (a packet is not equal to null) **do**
 if (TCP packet arrives) **do**
 TCPCount++;
 if (TCP Packets % threshold \geq 70) **do**
 Alarm "TCP Flow Attack has been detected!";
 else if (IP Packets % threshold \geq 70 && source_ip == destination_ip) **do**
 Alarm "IP Address Pattern has been detected!";
 else
 Show "Error"
 end if;
 end if;
 end if;

Algorithm 3. The following algorithm is the mitigation of attack traffic performed using SFaDMT.

Input: **Data Traffic Dt**
Output: **Analyze Dt**
while (true)
 if (Dt arrives) **do**
 Dt++;
 if (Packet Header == Legitimate Dt) **do**
 Display "Legitimate Traffic and can access the network";
 else if (Packet Header == Malicious Dt) **do**
 Display "Malicious Traffic has been detected and mitigated";
 else if (Packet Header == Unknown Dt) **do**
 Display "Deep Analysis needs to be performed on this packet";
 else if (Packet Analysis == Dt Attack Pattern Detected) **do**
 Display "Restrict traffic from the network";
 else
 Display "Legitimate traffic has been detected and mitigated";

4. Result and Analysis

A simulator was built on a high-end system accessible in a computer lab to predict congestion for analysis and study the suggested structure (SFaDMT). The topology was built in a modelling tool called OMNET++, with nodes and clusters ranging from 20 to 300 mobile users. A distinction was made between the suggested adaptive approach and the pushback. During the execution of both strategies, a traffic dive was created repeatedly by different situations. The empirical study of simulation-generated outcomes could be seen in the net parts of the whole section.

The OMNET++ simulator was used to produce tests. A digital topology was developed, where nodes varying from 20 to 300 were only used to produce traffic flashes in the system, and the suggested dynamic methodology was used to identify SYN Flood attacks. The outcomes attained can be seen in the graphs of the output graph, which show the identification contrast between pushback and the suggested SFaDMT method. This indicates that the SFaDMT technique demonstrates that the results are faster and more efficient than those using the pushback methodology. The produced digital topology can be seen in the preceding images, below.

4.1. Case I

During the first test, as seen in Figure 8, a topology of eight nodes or modules was generated to implement the suggested SFaDMT methodology. Then, a traffic burst was

produced and the recommended methodology was used to identify a DDoS SYN Flood strike. The suggested strategy recognised the SYN Flood invasion at one megabyte, while the pushback strategy recognised the SYN Flood strike at 9–10 megabytes.

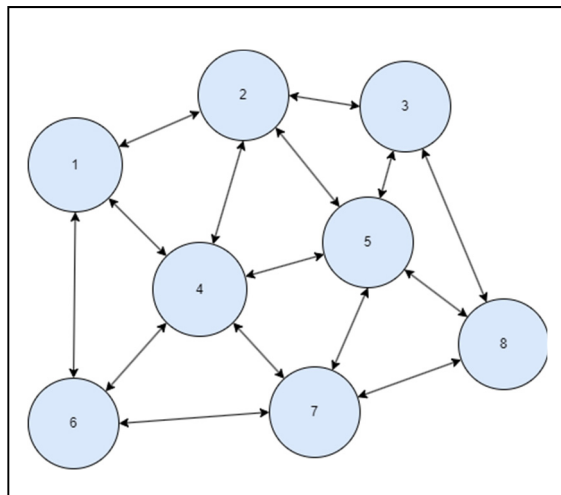


Figure 8. Topology of SFaDMT during the first run.

4.2. Case II

In Case II, as seen in Figure 9, a topology of 40 nodes and modules was developed for the simulated world of the SFaDMT methodology. With even more entities throughout the topology, a greater blast of traffic was produced relative to Case I. The suggested methodology identified a DDoS intrusion at 3–4 MB, while the pushback strategy detected DDoS attacks at 14–15 MB.

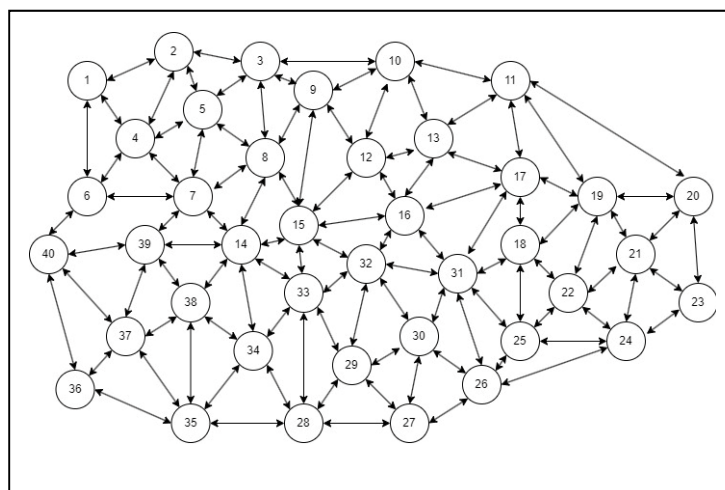


Figure 9. Topology of SFaDMT during the third run.

4.3. Case III

As seen in Figure 10, a topology of 120 modules or nodes was generated for the simulations of the suggested SFaDMT methodology during most of the fifth cycle. A traffic burst was produced, and the applied methodology was used to diagnose a DDoS SYN Flood strike. Because the nodes increased for each situation throughout the topology, a greater congestion burst was created compared to the prior case. The suggested methodology identified a DDoS intrusion at 14–15 MB, while the pushback strategy detected a DDoS attack at 40–43 MB.

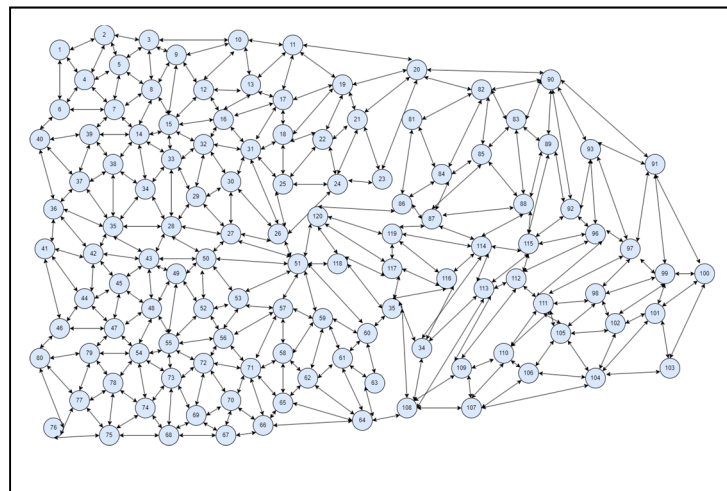


Figure 10. Topology of SFaDMT during the fifth run.

4.4. Case IV

A topology of 300 nodes or modules was developed, as seen in Figure 11, for the simulated model of the suggested SFaDMT methodology. A traffic burst was produced, and the proposed procedure was used to identify a DDoS SYN Flood invasion. The level of traffic burst improved, given the number of points and nodes. The recommended strategy recognised a DDoS attack at 11–12 MB, while the pushback strategy recognised a DDoS invasion at 60–63 MB.

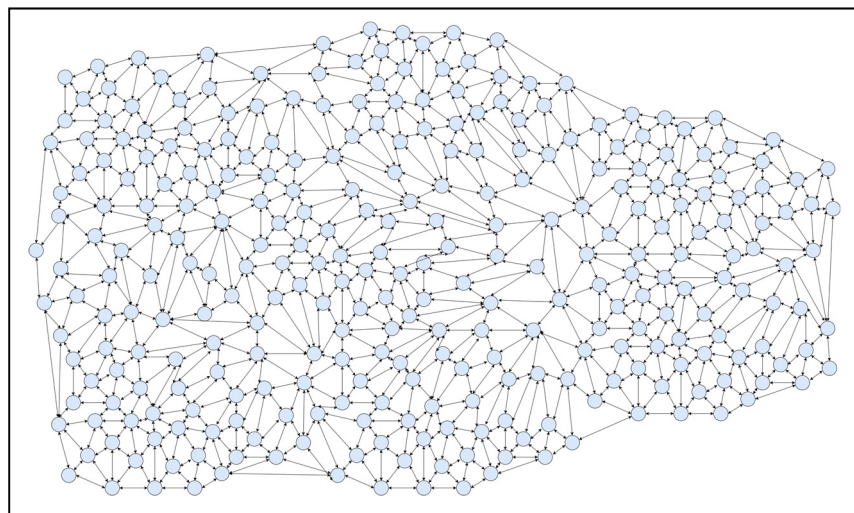


Figure 11. Topology of SFaDMT during the sixth run, with 300 nodes.

5. Evaluation Parameters

To evaluate the proposed SFaDMT Model simulation, a test bed was formulated according to the following parameter details described in Table 1.

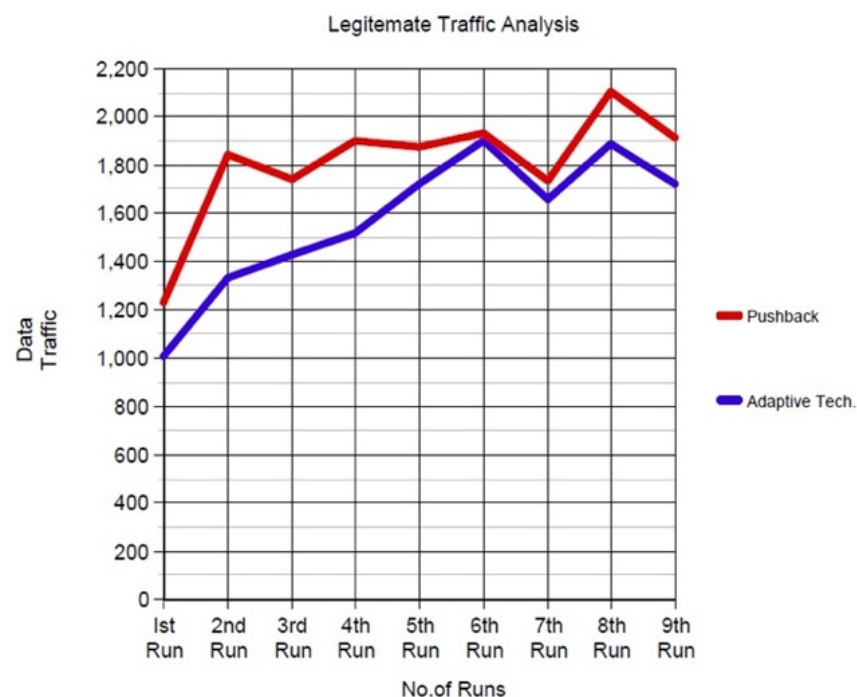
Analysis of the Adaptive Technique with the Previous Technique

The legitimate traffic analysis graph generated traffic to analyse the good- and bad-will packets. It is shown in the graph during the first run that the push-back technique detected the traffic, including malicious packets at 1.2 MB. In contrast, the adaptive SFaDMT technique detected at 1.0 MB. Therefore, whereas legitimate traffic was 0.8 Mb, the traffic detection ratio of the adaptive SFaDMT technique was better in the performance analysis than the pushback technique.

Table 1. Evaluation Parameters for OMNET Simulation.

Evaluation Parameters	
Nodes	10–200
N/W Type	Static
Traffic Burst	1.0–2.2 Gb
Malicious Node	Unknown
No. of Run	1–9

Figure 12 shows that during the second run, while generating the traffic pushback detection, the malicious traffic in the graph was at 1.8 MB while the SFaDMT identified it at 1.3 MB. Similarly, on the third run, the pushback technique detected at 1.75 MB, and SFaDMT identified at 1.4 MB. In the fourth run, pushback detected at 1.9 MB, whereas the adaptive proposed technique detected at 1.5 MB. In the fifth run, the pushback detected at 1.9 Mb, and the adaptive technique detected at 1.7 Mb, clearly showing the performance increase ratio between the previous and adaptive techniques. Finally, at the sixth run, both techniques almost detected at 1.9 MB. Still, the SFaDMT technique detected the traffic, including malicious packets, earlier than the previous technique used in the graph for comparison. For contrast, the details of those runs are shown in Table 2.

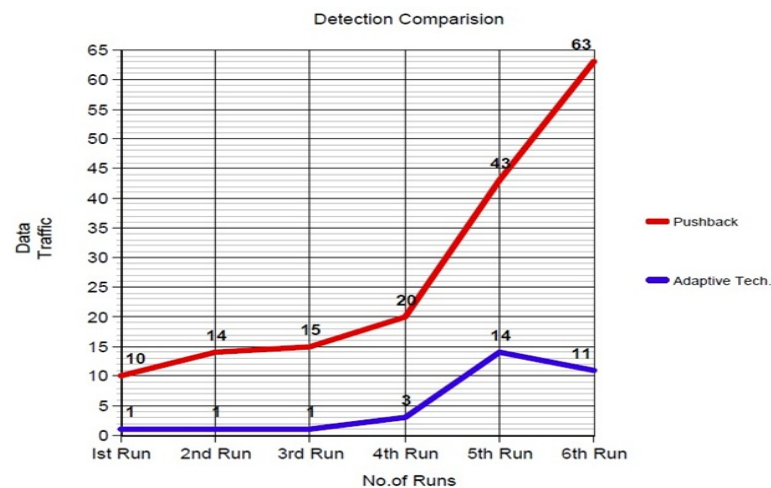
**Figure 12.** Graph comparisons of SFaDMT and pushback technique.

The seventh run pushback technique detected at 1.75 MB, and the proposed technique detected at 1.65 MB. The eighth-run pushback technique detected the good- and bad-will packets at 2.1 MB, while the SFaDMT identified them at 1.9 MB. In the ninth run, SFaDMT detected the malicious traffic that flows with the legitimate traffic at 1.7 MB, whereas the pushback technique detected it at 1.9 MB. Hence, it was concluded that there were fewer false positives in the proposed SFaDMT technique than in the previous technique, known as the pushback technique.

The detection comparison between the pushback and SFaDMT techniques shown in Figure 13 shows that during the first run, the pushback technique detected the DDoS attack at 10 MB. In contrast, the SFaDMT detected the attack when only 1 MB of traffic was generated.

Table 2. Traffic comparison analysis of both techniques.

Legitimate Traffic Analysis		
No. of Runs	Data Traffic (Mb)	
	Adaptive SFaDMT Technique	Pushback Technique
1st Run	1.0	1.2
2nd Run	1.3	1.8
3rd Run	1.4	1.75
4th Run	1.5	1.9
5th Run	1.7	1.9
6th Run	1.9	1.9
7th Run	1.65	1.75
8th Run	1.9	2.1
9th Run	1.7	1.9

**Figure 13.** Detection comparisons of pushback and SFaDMT technique.

On both the second and third runs, the SFaDMT technique similarly detected the traffic at 1 Mb, whereas the pushback technique rose to 14–15 Mb. The complete details with runs are shown in Table 3. As the burst size of traffic increased in each run, the SFaDMT detection technique detected the attack at 3 MB, while the pushback rose and reached 20 MB. In the fifth run, the SFaDMT detected at 14 MB and the pushback detected at 43 MB.

Table 3. Detection comparison of both techniques.

Detection Comparison		
No. of Runs	Data Traffic (Mb)	
	Adaptive SFaDMT Technique	Pushback Technique
1st Run	1.0	10.0
2nd Run	1.0	14.0
3rd Run	1.0	15.0
4th Run	3.0	20.0
5th Run	14.0	43.0
6th Run	11.0	63.0

In the sixth and last run, the SFaDMT detected at 11 MB, whereas the pushback technique detected at 63 MB, which shows that the adaptive technique can detect an attack at an earlier stage and increase the performance of the detection technique.

6. Conclusions

Among the most prevalent DOS (Denial of Service) attacks is the Syn Flood strike. It can affect the business side and services for legitimate customers, and it is impossible to eradicate such an invasion. However, the proposed strategy can considerably decrease the risk and harm caused by such attacks by taking the initiatives outlined in the journal. In a paper, the authors suggest an adaptive DDoS tracking mechanism that activates and minimises the attacks of Syn Flood. The envisaged SFaDMT methodology streams the TCP packets and the tracking contrast, which is decided based on the policies based on the registrations characterised throughout the SFaDMT identification system. The attack is observed if the network traffic ratio is greater than 70% relative to the regulations, and the signatures are processed. The traffic is deemed suspicious relative to the host network and is prohibited from accessing system resources. When the contrast is less than 75% of the traffic from a different host, it will be treated as encrypted traffic. If the traffic sequence is unclear, a layer packet examination can be conducted on the SYN packet interface to determine if it includes suspicious content. A prototype was built in OMNET for the application of the SFaDMT methodology with 20–300 mobile users, as well as a contrast with the pushing-back strategy. Traffic loads of varying sizes were created for identification, including network traffic assessment using the suggested adaptation and pushback techniques. Tests have shown that the new methodology has a 26 percent better identification rate than the current push-back strategy.

Author Contributions: Conceptualization, B.R. and H.A.J.S.; Methodology, B.R. and K.H.; Software, A.A.H.E.; Validation, H.A.J.S., A.A. and M.K.; Formal analysis, M.K.; Investigation, K.H. and F.K.; Data curation, A.A.H.E.; Writing—original draft, A.A.; Writing—review & editing, F.K.; Visualization, F.K. All authors have read and agreed to the published version of the manuscript.

Funding: Deanship of Scientific Research at King Khalid University for funding this work through the large Research Groups Project under grant number (RGP.2/127/44).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the large Research Groups Project under grant number (RGP.2/127/44).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hwang, R.H.; Peng, M.C.; Huang, C.W.; Lin, P.C.; Nguyen, V.L. An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access* **2020**, *8*, 30387–30399. [[CrossRef](#)]
2. Novaes, M.P.; Carvalho, L.F.; Lloret, J.; Proença, M.L. Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. *IEEE Access* **2020**, *8*, 83765–83781. [[CrossRef](#)]
3. Lima Filho, F.S.D.; Silveira, F.A.; de Medeiros Brito Junior, A.; Vargas-Solar, G.; Silveira, L.F. Smart detection: An online approach for DoS/DDoS attack detection using machine learning. *Secur. Commun. Netw.* **2019**, *2019*, 1574749. [[CrossRef](#)]
4. Li, Z.; Rios, A.L.G.; Xu, G.; Trajković, L. Machine learning techniques for classifying network anomalies and intrusions. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
5. Lin, P.; Ye, K.; Xu, C.Z. Dynamic network anomaly detection system by using deep learning techniques. In *International Conference on Cloud Computing*; Springer: Berlin, Germany, 2019; pp. 161–176.
6. Jaafar, G.A.; Abdullah, S.M.; Ismail, S. Review of recent detection methods for HTTP DDoS attack. *J. Comput. Netw. Commun.* **2019**, *2019*, 1283472. [[CrossRef](#)]

7. Nawir, M.; Amir, A.; Yaakob, N.; Lynn, O.B. Effective and efficient network anomaly detection system using machine learning algorithm. *Bull. Electr. Eng. Inform.* **2019**, *8*, 46–51. [[CrossRef](#)]
8. Ghaffari, F.; Gharaee, H.; Arabsorkhi, A. Cloud security issues based on people, process and technology model: A survey. In Proceedings of the 2019 5th International Conference on Web Research (ICWR), Tehran, Iran, 24–25 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 196–202. [[CrossRef](#)]
9. Kemp, C.; Calvert, C.; Khoshgoftaar, T. Utilizing netflow data to detect slow read attacks. In Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 6–9 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 108–116.
10. Hatem, M.A.; Shaker, V.; Jabbarpour, M.R.; Jung, J.; Zarrabi, H. HIDCC: A hybrid intrusion detection approach in cloud computing. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4171. [[CrossRef](#)]
11. Aborujilah, A.; Musa, S. Cloud-based DDoS HTTP attack detection using a covariance matrix approach. *J. Comput. Netw. Commun.* **2017**, *2017*, 7674594. [[CrossRef](#)]
12. Alzahrani, S.; Hong, L. A survey of cloud computing detection techniques against DDoS attacks. *J. Inf. Secur.* **2017**, *9*, 45–69. [[CrossRef](#)]
13. Hong, K.; Kim, Y.; Choi, H.; Park, J. SDN-assisted slow HTTP DDoS attack defense method. *IEEE Commun. Lett.* **2017**, *22*, 688–691. [[CrossRef](#)]
14. Kaur, P.; Kumar, M.; Bhandari, A. A review of detection approaches for distributed denial of service attacks. *Syst. Sci. Control Eng.* **2017**, *5*, 301–320. [[CrossRef](#)]
15. Ramírez-Gallego, S.; Krawczyk, B.; García, S.; Woźniak, M.; Herrera, F. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing* **2017**, *239*, 39–57. [[CrossRef](#)]
16. Sahi, A.; Lai, D.; Li, Y.; Diyk, M. An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. *IEEE Access* **2017**, *5*, 6036–6048. [[CrossRef](#)]
17. Gomes, H.M.; Bifet, A.; Read, J.; Barddal, J.P.; Enembreck, F.; Pfharinger, B.; Holmes, G.; Abdessalem, T. Adaptive random forests for evolving data stream classification. *Mach. Learn.* **2017**, *106*, 1469–1495. [[CrossRef](#)]
18. Liu, B.; Chen, J.; Hu, Y. Mode division-based anomaly detection against integrity and availability attacks in industrial cyber-physical systems. *Comput. Ind.* **2022**, *137*, 103609. [[CrossRef](#)]
19. Falliere, N.; O'Murchu, L.; Chien, E. *W32. Stuxnet Dossier (Version 1.4)*; Symantec: Tempe, AZ, USA, 2011.
20. Zeller, M. Myth or reality—Does the Aurora vulnerability pose a risk to my generator? In Proceedings of the 2011 64th Annual Conference for Protective Relay Engineers, College Station, TX, USA, 11–14 April 2011; pp. 130–136.
21. Khalid, A.; Kirisci, P.; Khan, Z.H.; Ghrairi, Z.; Thoben, K.-D.; Pannek, J. Security framework for industrial collaborative robotic cyber-physical systems. *Comput. Ind.* **2018**, *97*, 132–145. [[CrossRef](#)]
22. Choi, J.; Choi, C.; Ko, B.; Kim, P. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. *Soft Comput.* **2014**, *18*, 1697–1703. [[CrossRef](#)]
23. Basheer Riskhan, R.M. Virtual Machine Performance Approaches in the Online Education System. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2016 Vol I, IMECS 2016, Hong Kong, China, 16–18 March 2016.
24. Modi, C.; Patel, D.; Borisaniya, B.; Patel, A.; Rajarajan, M. A survey on security issues and solutions at different layers of cloud computing. *J. Supercomput.* **2013**, *63*, 561–592. [[CrossRef](#)]
25. Bakshi, A.; Dujodwala, Y.B. Securing cloud from DDoS attacks using intrusion detection system in virtual machine. In Proceedings of the 2010 Second International Conference on Communication Software and Networks, Singapore, 26–28 February 2010; pp. 260–264.
26. Munz, G.; Carle, G. Distributed network analysis using TOPAS and wireshark. In Proceedings of the NOMS Workshops 2008-IEEE Network Operations and Management Symposium Workshops, Salvador, Brazil, 7–11 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 161–164.
27. Douligieris, C.; Mitrokotsa, A. DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Comput. Netw.* **2004**, *44*, 643–666. [[CrossRef](#)]
28. Riskhan, B.; Zhou, K.; Muhammad, R. Energy Management of the System: An Empirical Investigation of Virtualization Approaches in Static and Dynamic Modes. *Inf. Technol. J.* **2016**, *16*, 1–10. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.