International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Dynamic Firewall Rule Building Engine for Hybrid Cloud

Shesagiri Taminana[1*] D.Lalitha Bhaskari[2]
[1]Research Scholar, Computer Science &Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam,India, tsgiri32@gmail.com
*Corresponding Author

[2]Professor, Dept. of Computer Science&Systems Engineering, Andhra University College of Engineering(A), Andhra University,Visakhapatnam,India, lalithabhaskari@yahoo.co.in

Abstract - Growth in the cloud computing resource management aspects have also increases the risks associated with the services hosted on the cloud.The foundational challenges are firstly to accommodate the ease of access without violating the security aspects and the time constraints to provide service responses on time, to provide the security to the active services. A good number of researches are observed to achieve the best firewall security to the hosted services. Nonetheless, the existing methods are often criticised for higher complexity and the non-performing aspects for the newer attack types. Henceforth, this work aims to solve the existing research bottlenecks. The proposed work firstly aims to solve the higher complexity of the deployed strategy by reducing the attribute sets with a sense of accuracy, time complexity and information loss. Further, this work proposes a dynamic firewall rule engine design strategy for detection of the attacks using the thresholding method. The proposed algorithms are testing on the benchmarked KDD dataset and as outcome a nearly 99.7% accuracy is observed and the time complexity is reduced to nearly 40%. Hence, this proposed work demonstrates a state-of-the-art technology for firewall design for hybrid cloud and shall be considered as a new benchmark in this domain of the research.

Keywords—Adaptive, Attributes Reduction, Progressive Accuracy, Thresholding, Dynamic Rule Formation

I. Introduction

As stated by S. Bagheri et al. [1], Firewalls are effective in filtering harmful traffic and providing a high degree of network security. As a result, their performance is crucial to the whole network. In the world of conventional firewalls, rule-based firewalls are the most extensively used kind of security measure. The lookup latency of a firewall grows noticeably in proportion to the size of its rule list. “Reordering rules based on traffic characteristics may reduce the number of packets matches and hence improve firewall performance” [1]. Ordering the best firewall rules is an NP-Complete issue. It's impossible to set up a whole network's firewall from a single location. Outsourcing the firewall operations to micro firewalls that are dispersed across the network and each have their own set of settings is the approach used in this study's solution. As a result, traffic is handled at the neighbourhood level and dispersed. Based on the results of our tests, They are certain that our solution can grow with the organization's network needs. Central firewall rules optimization methods are executed sequentially, which is a waste of time.

Internal services and hosts may be protected by establishing “distributed firewalls with access control lists in software-defined” networks as the network environment becomes more dynamic and complicated. The ever-changing nature of cyberattacks necessitates regular modifications to firewall rules. Because of this, the rule placement in most current integer linear programming-based solutions must be recalculated. High computing overhead is a result of this. Dynamic firewall rule changes and rule placement provide a hurdle. They offer a resource constraint splitting approach to calculate just the rules that are relevant to the new policies while retaining the rest. This strategy relies on breaking the issue down into smaller, more manageable components before tackling the whole thing as a whole. According to the study of Y.-W. Chang et al., this strategy exhibits much “less calculation time while retaining the optimal rule placement for network performance”. [2].

This dynamic key system is conceived and deployed using fingerprint features of the wireless channel. According to T. Shangte and colleagues, a new “location” security factor has been introduced to IoT. [3]. “The user's typical communication with the Internet of Things device is

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

used to derive channel characteristics. Quantification, negotiation, and privacy amplification are all steps in the process of generating a dynamic key that is tied to the user's current location in real time. The wireless identity key, unlike the conventional identification key, has no distribution, dynamic change, and compatibility with the current key security system". Users and IoT devices are separated by a "firewall" that changes dynamically. It is impossible for an unlawful user to assault a legitimate user in an unknown area if their locations are distinct. The "fixed-point control" UAV and its control connection have been created.

Firewall Design Foundation

After the initial context setting for the research, in this section, the traditional method for firewall design is discussed.

Assuming that, the connections, C[], made to the cloud services are collection of various properties and each property can be denoted as CX. Thus, for n number of connections the following relation can be formulated,

$$C[] = \{C_1, C_2, C_3, ...C_n\} \quad \text{(\#Eq.1)}$$

Further, to build any sophisticated firewall, the primary step is to understand the domain for all characteristics. The better strategy to identify the characteristics of the domain for any such attributes, is to identify the mean analysis for the domain. Assuming that, the mean for the attribute domain CX[]is M|CX|, which can be formulated as,

$$M \mid C_X \mid = \frac{\sum_{i=1}^{n} C_i}{\phi\{\} C_X} \quad \text{(\#Eq.2)}$$

Where, $\phi\{\}$ is the function to count the number of elements in one domain of the fields.

Further, assuming that, the TH[] is the collection for each attribute threshold, thus, this can be formulated as,

$$TH[] = \int_{i=1}^{n} M \mid C_i \mid \quad \text{(\#Eq.3)}$$

Again, each and every domain of the attribute, CX[] in the connection characteristics set, for m number of observations in the domain, can be formulated as,

$$C_X[] = \{C_{X1}, C_{X2}, C_{X3}, C_{X4}, ...C_{Xm}\} \quad \text{(\#Eq.4)}$$

Finally, during the analysis of the dataset for designing the firewall, the rulesets, Flag[], is formulated as,

$$C_X[X] :: M \mid C_X[] \mid \Rightarrow Flag[] \quad \text{(\#Eq.5)}$$

And,

$$Iff\{Flag[X] :\rightarrow TRUE\} > 70\% * \phi\{\} C_X$$
$$Then, Flag[X] \rightarrow Attack,$$
$$Else, C_X \rightarrow Normal \quad \text{(\#Eq.6)}$$

This is the foundational process for building and designing the firewall.

In the light of the foundational method discussed in this section of this work, in the next section, the parallel and recent research outcomes are discussed with realization for research gaps.

II. Literature Review

After understanding the context of the research and further realizing the base line method of firewall design, in this section, the parallel recent research outcomes are analyzed.

Because of the firewalls' ability to dynamically redistribute filtering rules within the same network, they can adapt to fluctuations in the volume of traffic they handle. For networks with many cascading firewalls, this study proposes "a unique formal model that allows the transfer of rules from a firewall to its downstream neighbours when changes in the input traffic profile indicate it. Finally, a method of preserving the integrity of the network while transferring rules across cascaded firewalls and improving packet processing rates for a given traffic profile is proposed" [3]. By use of simulation, the correctness of the suggested approach has been formalised and confirmed. This method was also put through its paces using L. Durante et al. [4]'s laboratory experimental test-bed.

Since the introduction of this new networking paradigm, as shown by the work of N. Sophakan et al. [5], OpenFlow has become the primary standard protocol.TLS, which has been proved to be susceptible. So-called "Secured-OF" was the goal of this investigation. A stateful firewall was utilised to save data for subsequent investigation. There were two types of DDoS attacks that were studied using a Dynamic Bayesian Network (DBN). An attack's type is determined by analysing packet states, and that information is then added to the flow table entry. Several performance criteria were used to assess the suggested Ryu controller Secured-OF model. The suggested Secured-OF system was tested on an emulation network for analysis. 99.5 percent of attacks were detected by the suggested Secured-OF technique. As a result, compared to an SDN without security implementation, it was able to increase OpenFlow controller security considerably with no performance damage.

Smart gadgets (sometimes referred to as Internet-of-Thing's devices) that connect to the internet are often lacking in basic security measures. As a result, the safety and privacy of online users are jeopardised. Horizontal port scans are a critical step in identifying and exploiting susceptible devices. Horizontal port scanning in home networks is the topic of this investigation.With a minimum of overhead, FleXight provides the controller with the data it needs via per-flow sampling and dynamic sample rate modifications. According to S. Shirali-Shahreza et al. [6], this paper also presents a comprehensive usability examination of our system.

To combat the emerging coronavirus epidemic, doctors and hospitals are relying on medical gadgets that are more exposed to security risks. Software Defined Networks (SDN) may be used to provide a comprehensive security framework for a healthcare institution, as shown in this study by G. Rezaei et al. [7].

The dynamic nature of cloud-based corporate networks makes it difficult to secure the perimeter. Cloud-based corporate networks' firewalls now "rely heavily on static security rule setup or basic rule matching, making them inflexible, error-prone, and underwhelming in terms of security" [8].

There is no doubt that the Internet of Things (IoT) will continue to permeate and improve our daily lives, whether at the workplace, at home, or in the commercial sector. IoT botnets have recently emerged as a potent threat to the Internet. Many businesses and gadgets throughout the globe have been impacted by malware like Mirai and Reaper. According to J. Myers et al. [9], an integrated cloud-based machine learning architecture with an on-site dynamic firewall protects consumer IoT devices.

Mobile customers' demands for security services such as "network security, end-to-end latency, and security" resources heavily influence the traffic flows handled by VSNFs. For example, W. Qiao and co-workers [10] argue that "identifying how to embed VSNFs in cloud datacenter networks and route service pathways that maximise resource efficiency is a tough task, especially without breaching latency limitations and "security service requirements". Thatbe referred to as the dynamic embedding of the security service chain (SSC-DMP)". NFV-enabled framework is presented in this study to accomplish dynamic SSC embedding in cloud data centres. SSC-DMP can be solved precisely in a small-scale network architecture using an integer linear programming (ILP) paradigm. The authors of this paper next present an effective particle swarm optimization-based SSC dynamic embedding solution to minimise the temporal complexity of large-scale network topology applications. "In terms of resource usage and end-to-end time, extensive simulations demonstrate that the proposed approach might beat the existing benchmarks by at least 35.2% and 23.1%, respectively".

It is possible to run Windows on a portable device A malicious attack on executable files is possible. To get beyond normal security detection and protection measures, there are several methods and strategies. For example, "a writable file in a user's temporary folder with a filename that looks like a valid process" may be bypassed by Windows Defender Firewall without the user's involvement. For malware classification, R. Chanajitt et al. [11] use static attributes and dynamic behaviour analysis. The Falcon Sandbox malware website is used for dynamic analysis.

Allows for an unprecedented level of flexibility in the deployment of network services. There is an ever-expanding array of orchestration options available in cloud systems due to the proliferation of new developing technologies. Dynamic cloud setups are difficult to reconcile with predictable performance in this setting. An acceptable resource allocation in the virtualized infrastructure must be linked to the service provider's performance specifications as specified in the Service Level Agreement (SLA). VNF profiles are suggested by S. Van Rossem et al. [12] as a means of easing the burden of this procedure. "A virtual router, switch, firewall, and cache server"are used to demonstrate this point. A model is then built from the profiled datasets using a variety of

techniques. This study selects the most accurate technique for future training. It is possible for our technology to provide the service provider "recommendations on resource allocations based on SLA-specified performance goals and maximum workload". This reduces the number of scaling steps required to provide the softwarized service while still meeting the SLA criteria.

Brew, an OpenDaylight SDN controller-based security policy analysis framework, is presented by S. Pisharody et al. [13] "to ensure that no two flow rules in a distributed SDN-based cloud environment have conflicts at any layer; this ensures consistent conflict-free security policy implementation and prevents information leakage" [14]. SDN flow rule conflicts are treated in the same way as conventional firewall rule conflicts, with the addition of ways for resolving these conflicts on their own using a decentralised approach that recognises and categorises conflict arising from cross-layer issues. As an alternative, if administrator involvement is required to settle disputes, a unique visualisation method is created to aid administrators in seeing the conflicts visually. A proof-of-concept prototype is used to show that our framework is valid, feasible, and scalable.

As more criteria are added to the list, the communication latency increases as a direct result of this linear search. ORO seeks to determine the sequence of rules that reduces classification delay induced by packet classification while respecting classification policy to reduce this latency. So far, heuristics for ORO have been devised, and the problem is known to be NP-complete. SGM is now the most advanced ORO heuristic algorithm. Research by T. Harada and coworkers[14] provides a fresh approach to ORO. However, instead of looking for the highest weight rule and recursively moving it higher,

The systems that underpin company operations are becoming more complicated and sophisticated as well. Increased cybersecurity dangers need more protection for company systems because of this predicament. For the "protection of internal systems from external threats, firewalls, intrusion detection systems (IDS), and web application firewalls" have all been widely used. As a result, these security perimeter devices were unable to identify new and previously undiscovered threats. Threat hunting is proposed in this study, which uses logs from security devices and network forensic analysis to find undiscovered threats and verify the findings using memory forensics. According to A. Oktadika et al. [15], new regulations resulting from this revelation will make border security "more dynamic and effective"for dealing with hitherto identified defence concerns.

To fulfil the varying traffic needs of their users, network operators deploy a broad variety of VNFs. VNFs are efficiently selected, and traffic is steered by the interference effect using a heuristic technique proposed in this study. V. Reddy et al. [16] shown that the suggested technique enhances the SFC acceptance rate by 29 percent compared to previous approaches.

As far as Ad hoc networks are concerned, there have already been a number of successful implementations. In ad hoc networks, nothing is done to provide security and avoid attacks (like MANETs). Ad hoc networks lack centralised defence mechanisms like firewalls, intrusion detection systems, or proxy servers, as research has shown. With its self-maintenance and wireless connectivity, ad hoc networks are very handy. Ad hoc networks may be set up almost anywhere, at any time, thanks to their flexible architecture. Therefore, the ad hoc components must have a safe method in place so that they may have the same level of security as their pre-built counterparts. Ad hoc networks are shown to be well-protected by this work's secure and adaptable implementation. H. Vegda et al. [17] exhibited a full suggested system based on "NS 2.35 software" and "Ubuntu (Linux) OS".

According to M. Korkmaz et al. [18], a significant tendency has emerged in recent years to migrate almost all real-world processes online due to the increasing use of mobile devices. Even if this eases our everyday routines, the anonymity of the Internet renders it vulnerable to security breaches. Most assaults may be thwarted with the help of antivirus software and firewalls. As a result, savvy attackers target computer users' vulnerabilities by using phoney websites to attempt to steal their personal information. In the market for phishing detection there are a variety of approaches, including "blacklists, rule-based detection and anomaly-based detection". "Anomaly detection" using machine learning is becoming more popular in the literature because of its dynamic nature, particularly for spotting "zero-day" assaults. Phishing detection system was suggested in this study employing eight algorithms to evaluate URLs and three datasets comparing the findings with previous research. The results of the experiments show that the provided models perform very well in terms of their success rate.

5G and beyond will need to fulfil strict latency, reliability, and heterogeneous device support criteria, as indicated by S. T. Arzo et al [19]. The current wireless

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

network design, on the other hand, is constrained by these limitations. For greater network adaptability, cloud radio access networks and network function virtualization are being touted. It separates cloud-deployed network operations like firewall and packet gateway from their underlying hardware. The location of virtual network functions hence necessitates an end-to-end specification of this architecture. Virtual function placement is the subject of most extant research, with distinct aims and service needs being addressed independently. Finally, this study proposes service differentiation in order to minimise CPU use and energy consumption, while still using the strategies described above. Using service differentiation to better allocate computer resources and reduce energy consumption may have major benefits, according to the findings.

Cloud-based VNFs may be used to execute services like load balancing; routing; forwarding; firewalls; and so on. This results in an improved network performance. A virtual SDN (vSDN) controller (VNF) was used to perform traffic load balancing in this study. Secondary vSDN controllers may be installed to distribute the increasing and unequal load of vSDN. A secondary vSDN is needed, therefore a duplicate vSDN is made with the exact same settings as the original vSDN. This copy vSDN works precisely as the original vSDN did, and it shares traffic load balancing responsibilities with the original vSDN. Every client in the network is made aware of the presence of a second vSDN controller that has been installed in the cloud, ensuring complete transparency. In a Mininet emulation, two vSDN controllers were used to test the load balancing in a Fat-Tree architecture. As proved by S. Ejaz et al. [20], there was a 50% improvement in average load, a 41% reduction in average latency, and significant increases in terms of ping response, bandwidth usage, and throughput.

Today, "intrusion detection and prevention systems (IDS / IPS) constitute an essential component of network security", as shown by V. Belenko et al. [21]. Unauthorized access to the network may be detected and actions can be taken, such as alerting security personnel, reconfiguring the firewall to stop additional assaults, and safeguarding against malware and other intrusions. Many techniques to thwarting hostile behaviour on conventional computer networks exist; however, for the fast-evolving dynamic adhoc networks, the need for a universal protective measure is essential. Polymorphic viral code and operation sequences are examined, compared, and a bioinformatics-based approach for identifying polymorphic chains of operations is implemented in this study. This study also presents a comparative analysis. This strategy was evaluated with several operation sequences and k-means, and the best parameters for it were found.

Further, in the next section of this work, the identified persistent research problem is formulated.

III. Problem Formulation

After the detailed discussion on the parallel and recent research outcomes, in this section of the research the persistent research problems are discussed.

Firstly, the existing methods significantly fails to identify the newer types of attacks. Continuing from the Eq. 4, assuming that the length of the connection is m at the time instance t1 and further at time t2, the length of the dataset is m+k. Primarily, assuming that the growth rate of the data is (t2 – t1).unit of data is k. Thus, this can be formulated as,

$$C_X[](t_1) = \{C_{X1}, C_{X2}, C_{X3}, C_{X4}, ...C_{Xm}\} \quad \text{(\#Eq.7)}$$

And,

$$C_X[](t_2) = \{C_{X1}, C_{X2}, C_{X3}, C_{X4}, ...C_{Xm}.....C_{Xk}\} \quad \text{(\#Eq.8)}$$

Thus, while calculating the thresholds at the time instance t1+(t2-x), the available dataset length is m+k, however, the length of the data is not calculated appropriately. As result, the threshold is not calculated accurately.

This results into non-detection of the attacks happening between the time instance of t1+x and t2.

Secondly, the existing methods demonstrates significantly higher time complexity.

As per the previous assumptions made in this work, assuming that the time complexity is T, thus the time complexity can be formulated as,

$$T(n) = n * m \quad \text{(\#Eq.9)}$$

If n≈m, then this can be rewritten as,

$$T(n) = n * n \quad \text{(\#Eq.10)}$$

Or,

$$T(n) = O(n^2) \quad \text{(\#Eq.11)}$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

As the process is significantlytime dependent and the complexity is at the order of n2, thus it is natural to realize that for a longer duration, the overall time complexity can be significantly astronomical.

Henceforth, in the next section of this work, the proposed solution methods are established using mathematical models.

IV. Proposed Solution

After realizing the persisting gaps in the current research, in this section of the work, the proposed solutions are formulated.

The proposed solution works in two phases. In the first phase, the proposed strategy reduces the dimension of the connection characteristics in order to reduce the model complexity and further reduces the time complexity.

The proposed method analyses the characteristics set using traditional clustering method to identify the best balance between the accuracy and time complexity while identifying the set of reduced attributes. Continuing from Eq. 2, the following formulation can be established for generating the length of the domain as,

$$\phi\{\}C[] \rightarrow n \qquad \text{(\#Eq.12)}$$

Assuming that, during the reduction process for the clustering any traditional algorithm can be deployed as denoted with CS and during the clustering process, λ and β are the accuracy and time complexity respectively. Thus, the relation can be written as,

$$CS\{C[]:n\} \rightarrow (\lambda, \beta) \qquad \text{(\#Eq.13)}$$

Re-iterating the same, for n and n-1 number of attributes, the following equation can be written,

$$CS\{C[]:n\} \rightarrow (\lambda_1, \beta_1) \qquad \text{(\#Eq.14)}$$

And,

$$CS\{C[]:n-1\} \rightarrow (\lambda_2, \beta_2) \qquad \text{(\#Eq.15)}$$

Further,

$$iff\ (\lambda_2, \beta_2) > (\lambda_1, \beta_1) \Rightarrow C[1..(n-1)] \qquad \text{(\#Eq.16)}$$

Hence, the reduced set of attributes can be considered.

Secondly, the proposed method demonstrates the rule formation for building the dynamic firewall.

Assuming that the reduced dataset is C'[] from the previous reduction process and assuming two attributes as X and Y, the rule formation can be identified as,

$$\begin{aligned} &C'[X] :: C'[Y] \rightarrow [\phi\{C'[X] > M \mid C'_X[]|\} \\ &= \theta_X \ and\ \phi\{C'[Y] > M \mid C'_Y[]|\} = \theta_Y] \\ &\Rightarrow \theta_X . C'[X] + \theta_Y . C'[Y] \end{aligned} \qquad \text{(\#Eq.17)}$$

Henceforth, the final rule engine can be formulated as,

$$\Re = \{\theta_X . C'[X] :: \theta_Y . C'[Y]\} \qquad \text{(\#Eq.18)}$$

Henceforth, based on the proposed mathematical models, in the next section of this work, the proposed algorithms are furnished.

VI Proposed Dynamic Firewall Design Method

Firewall design techniques for cloud computing are critical in protecting cloud-based assets from cyber threats. As more organizations adopt cloud computing, it is imperative to develop and implement robust firewall strategies that ensure security and compliance. In this we propose the firewall design techniques for cloud computing.

Phase - 1. Identification of Network segmentation: In this phase the Network segmentation is a firewall design technique is applied that involves dividing a network into smaller sub-networks, making it easier to manage and secure. In cloud computing, network segmentation is crucial in separating sensitive and non-sensitive data. By segmenting the network, organizations can reduce the attack surface and limit lateral movement of cyber attackers within the network.

Phase - 2. Design the Virtual firewalls: During this phase, the firewall is implemented on a virtual machine in the cloud. Virtual firewalls offer the same security features as traditional firewalls, including intrusion detection and prevention, packet filtering, and network monitoring. However, virtual firewalls are more scalable and cost-effective since they are deployed on a virtual machine.

Phase - 3. CASB Assessment: A Cloud Access Security Broker (CASB) is a security tool that is used

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

to secure data that is stored in cloud services such as Amazon Web Services, Microsoft Azure, and Google Cloud. CASBs help organizations monitor cloud traffic, identify and remediate data breaches, and enforce security policies. CASBs are especially useful in preventing unauthorized access to cloud-based data.

Phase - 4. Validation: Zero Trust Architecture is a security model that assumes no trust in any user or device that accesses a network. This firewall design technique uses a multi-layered approach to security, including identity verification, access control, and data encryption. Zero Trust Architecture is particularly useful in securing cloud-based applications and services since it assumes that all traffic is untrusted.

Phase - 5. Deployment: Software-Defined Perimeter (SDP) is a firewall design technique that enables organizations to create secure perimeters around applications

and data. SDP uses a dynamic, software-defined approach to network security that enables organizations to limit access to sensitive data to only authorized users and devices. SDP is particularly useful in securing cloud-based applications and services since it allows organizations to create secure perimeters around cloud-based applications.

Finally, firewall design techniques for cloud computing are critical in securing cloud-based assets from cyber threats. Network segmentation, virtual firewalls, CASBs, Zero Trust Architecture, cloud-native firewalls, and Software-Defined Perimeters are some of the firewall design techniques that organizations can use to secure cloud-based applications and services. By implementing robust firewall strategies, organizations can ensure the security and compliance of their cloud-based assets.

V. Proposed Algorithm

Based on the problem formulation and proposed solutions elaborated in the previous sections of this work, in this section the proposed algorithms are furnished.

Firstly, the Adaptive Attributes Reduction using Progressive Accuracy and Time Complexity (AAR-PATC) Algorithm is furnished.

Algorithm - I: Adaptive Attributes Reduction using Progressive Accuracy and Time Complexity (AAR-PATC) Algorithm
Input: Characteristics set as C[]
Output: Reduced Characteristics set as CR[]
Process: Step - 1. Load the Characteristics set as C[] Step - 2. For each attribute in C[] as C[i] a. Apply K-Means algorithm with k=10 b. Calculate the accuracy as A[i] using Eq. 13 and 14 c. Calculate the time complexity T[i] using Eq. 13 and 14 Step - 3. For each element in A[] and T[] a. If A[j]<A[j+1]*5% and T[j]>T[j+1]*10% b. Then, CR[] = C[0..n-1] c. Repeat till A[j] highest and T[j] is lowest. Step - 4. Return CR[]

Covariance (and sometimes correlation) matrices are constructed, and eigenvectors are computed on these matrices. In practise an important part of the data can now be reconstructed using the eigenvectors (principal components) that correspond to the largest eigenvalues. In low-dimensional systems, the vast majority of the system's energy can be attributed to the system's first few eigenvectors, which can be used to infer the system's large-scale physical behaviour. There is still a need to prove this one by one, as not all systems display this behaviour. The original space has been reduced to the space spanned by a few eigenvectors (with data loss, but hopefully retaining the most significant variance).

Secondly, the Threshold Based Dynamic Rule Formation (TBDRF) Algorithm is furnished.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Algorithm - II: Threshold Based Dynamic Rule Formation (TBDRF) Algorithm
Input: Reduced Characteristics set as CR[]
Output: Firewall Dynamic Rule Engine as R[]
Process: Step - 1. Load the reduced characteristics set as CR[] Step - 2. For each attribute in CR[] as CR[i] a. Calculate the mean for CR[i] as MCR[i] using Eq. 3 b. If CR(t)[i][j] > MCR[i] c. Then Mark Flag as F[i][j] = True Step - 3. For each attribute in F[][] as F[k][:] a. If Count(F[k][:] = True) > CR[i]*70% b. Then form the rule engine R[] using Eq. 17 c. Repeat the process using Eq. 18 Step - 4. Return the rule engine R[]

In its original sense, a firewall was a structure built to keep a fire contained within a row of nearby buildings. Metal sheets that separate the engine compartment of a vehicle or aircraft from its passenger area are referred to as similar structures in later references. It was first used in the late 1980s, when the Internet was still in its infancy as a global communications network. In the late 1980s, routers were the forerunners of today's firewalls for network security. Routers could apply filtering to packets crossing networks because they had already separated them.

Further, based on the proposed algorithms, in the next section of this work, the obtained results are analyzed and discussed.

VI. Results and Discussions

The obtained results are highly satisfactory and are discussed here in this section of the work.

Firstly, the dataset is described here [Table – 1]. The work is performed on the KDD dataset [22].

TABLE I. Dataset Description

Parameter	Value (#)
Number of Attributes	40
Number of Observations	311027
True Attacks Connections	250436
True Normal Connections	60591

The dataset characteristics are visualized graphically here [Fig – 1].

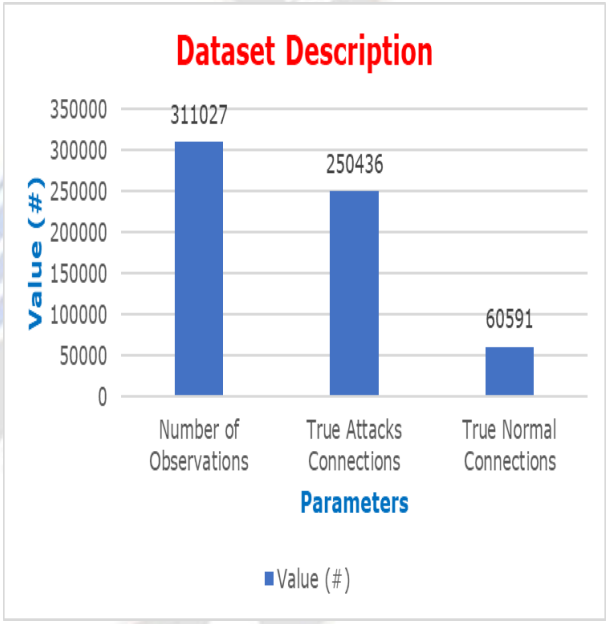

Fig. 1. Dataset Analysis

Secondly, the dimensionality reduction process is explained [Table – 2].

TABLE II. Reduction Process Analysis

Iteration	Accuracy (%)	Time Complexity (ns)	Attribute Set
1	55.95475715	1.205	[39, 40]
2	87.18119317	1.305	[2, 40]
3	90.19738301	1.318	[2, 3, 40]

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

4	91.70547793	1.364	[2, 3, 35, 40]
5	92.37081393	1.748	[2, 3, 35, 23, 40]
6	93.08050566	1.805	[2, 3, 35, 22, 23, 40]
7	93.25792859	1.851	[2, 3, 35, 22, 23, 40, 9]
8	93.25792859	2.496	[2, 3, 35, 22, 6, 23, 40, 9]
9	93.25792859	2.787	[2, 3, 35, 22, 6, 23, 7, 40, 9]
10	93.25792859	3.032	[2, 3, 35, 22, 6, 23, 7, 8, 40, 9]
11	93.25792859	3.258	[0, 2, 3, 35, 22, 6, 23, 7, 8, 9, 10]
12	93.25792859	3.462	[0, 2, 3, 35, 22, 6, 23, 7, 8, 9, 10, 13]
13	93.25792859	4.114	[2, 3, 35, 22, 6, 23, 7, 8, 40, 9, 10, 13]
14	93.25792859	4.186	[2, 3, 35, 6, 7, 8, 40, 9, 10, 13, 14, 22, 23]
15	93.25792859	4.257	[2, 3, 35, 6, 7, 8, 40, 9, 10, 13, 14, 16, 22, 23]
16	93.25792859	4.258	[2, 3, 35, 6, 7, 8, 40, 9, 10, 13, 14, 16, 17, 22, 23]
17	93.25792859	4.383	[2, 3, 35, 6, 7, 8, 40, 9, 10, 13, 14, 16, 17, 18, 22, 23]

It is natural to observe that, at the iteration 7, maximum accuracy can be achieved without missing the information available in the dataset and further, the time complexity increases with static accuracy. Hence, the reduced dataset is accepted at iteration – 7.

The outcome is again visualized graphically here [Fig – 2].

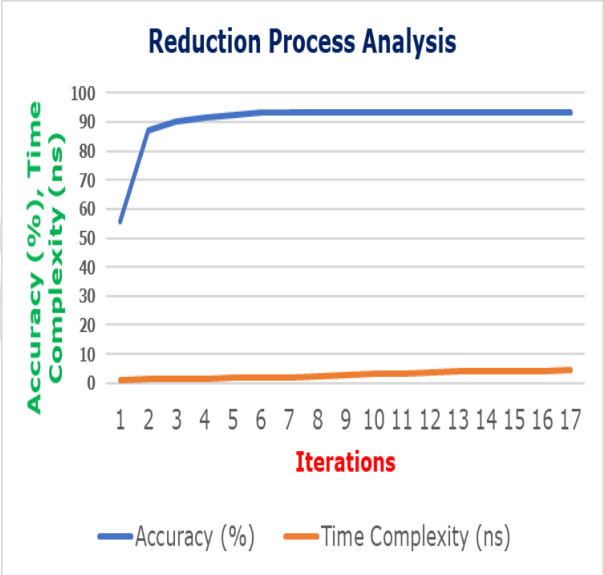

Fig. 2. Reduction Process Analysis

Thirdly, the reduced dataset description is furnished [Table – 3].

TABLE III. Final Dataset Description

SNO	Attribute Sequence	Attribute Name	Attribute Type
1	2	protocol_type	Content
2	3	service	Content
3	9	urgent	Content
4	22	is_guest_login	Traffic
5	23	count	Traffic
6	35	dst_host_diff_srv_rate	Host
7	40	dst_host_rerror_rate	Host
8	42	Class	Identifier

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 September 2023

The results are visualized graphically here [Fig – 3].

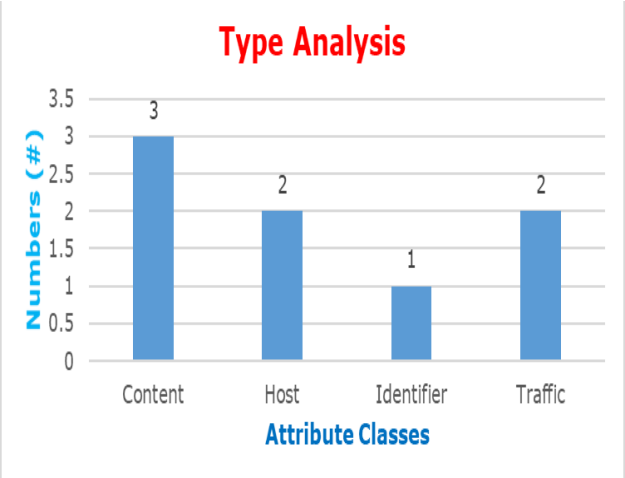

Fig. 3. Type Analysis

Fourthly, the rule engine sample rules are finished here [Table – 4].

TABLE IV. Sample Rule Set

Rule Engine
Rule- 1. "dst_host_diff_srv_rate> 0 AND service = telnet"
Rule- 2. "dst_host_diff_srv_rate> 0 AND service = pop_3"
Rule- 3. "dst_host_diff_srv_rate> 0.04 AND count > 93 AND protocol_type = tcp AND dst_host_diff_srv_rate<= 0.5"
Rule- 4. "dst_host_diff_srv_rate<= 0 AND service = ecr_i AND count > 4"
Rule- 5. "service = other AND count > 160"
Rule- 6. "service = eco_i AND count <= 1"
Rule- 7. "service = ecr_i AND count > 56"
Rule- 8. "service = other AND dst_host_diff_srv_rate<= 0.03"
Rule- 9. "count > 1 AND service = smtp AND dst_host_diff_srv_rate> 0.03 AND count > 2"
Rule- 10. "dst_host_diff_srv_rate> 0.97 AND service = private"
Rule- 11. "service = smtp AND dst_host_diff_srv_rate> 0.12 AND dst_host_diff_srv_rate<= 0.15"
Rule- 12. "dst_host_diff_srv_rate<= 0.01 AND count <= 27 AND service = ftp_data"
Rule- 13. "service = X11 AND dst_host_diff_srv_rate<= 0.08"
Rule- 14. "service = private AND protocol_type = udp AND count <= 151 AND count <= 7"
Rule- 15. "dst_host_diff_srv_rate> 0 AND dst_host_diff_srv_rate<= 0.01 AND protocol_type = tcp AND count > 7"
Rule- 16. "dst_host_diff_srv_rate> 0.08 AND service = ecr_i AND count <= 1"
Rule- 17. "dst_host_diff_srv_rate> 0.08 AND service = smtp AND dst_host_diff_srv_rate<= 0.16 AND dst_host_diff_srv_rate> 0.14"
Rule- 18. "dst_host_diff_srv_rate> 0.08 AND service = private AND protocol_type = udp AND count > 21"
Rule- 19. "dst_host_diff_srv_rate> 0.08 AND count > 20 AND service = private"
Rule- 20. "dst_host_diff_srv_rate> 0.08 AND service = private AND protocol_type = tcp"
Rule- 21. "count > 49 AND protocol_type = tcp"
Rule- 22. "service = other AND dst_host_diff_srv_rate> 0.97"
Rule- 23. "dst_host_diff_srv_rate<= 0.01 AND service = http AND count > 2 AND count <= 5"
Rule- 24. "count > 19 AND service = finger"
Rule- 25. "service = ftp_data AND dst_host_diff_srv_rate> 0.14 AND count <= 2 AND count <= 1"
Rule- 26. "service = http AND dst_host_diff_srv_rate> 0.01 AND count <= 12 AND dst_host_diff_srv_rate<= 0.04"
Rule- 27. "service = ftp_data AND dst_host_diff_srv_rate> 0.14 AND dst_host_diff_srv_rate<= 0.78 AND dst_host_diff_srv_rate> 0.28"
Rule- 28. "service = other AND protocol_type = tcp AND dst_host_diff_srv_rate<= 0.18"
Rule- 29. "service = smtp AND dst_host_diff_srv_rate> 0.03 AND

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

dst_host_diff_srv_rate<= 0.05"

Rule- 30. "service = ecr_i AND count <= 1"

Rule- 31. "service = http AND dst_host_diff_srv_rate> 0.01 AND count > 12"

Rule- 32. "service = ftp_data AND dst_host_diff_srv_rate<= 0.24"

Rule- 33. "service = other AND protocol_type = udp AND dst_host_diff_srv_rate> 0.07 AND count <= 41"

Rule- 34. "service = other AND dst_host_diff_srv_rate> 0.47 AND protocol_type = udp AND dst_host_diff_srv_rate<= 0.73"

Rule- 35. "service = http AND count > 1 AND dst_host_diff_srv_rate> 0 AND dst_host_diff_srv_rate<= 0.03"

Rule- 36. "service = http AND count <= 46"

The complete set of rules are dynamically applied on the network for detection of the attacks and normal connections.

Finally, the accuracy of attacks detections using the proposed dynamic firewall is furnished. The firewall is tested for nearly 100 iterations, however only 25 are listed [Table –5].

TABLE V. Performance Analysis

SNO	Attribute Sequence	Attribute Name
1	99.759	2.898
2	99.725	2.399
3	99.765	2.380
4	99.572	1.325
5	99.693	2.485
6	99.653	2.758
7	99.714	3.053
8	99.830	2.768
9	99.595	2.203
10	99.808	1.215
11	99.812	2.522
12	99.728	1.176
13	99.592	2.186
14	99.589	1.519
15	99.576	1.436
16	99.687	1.654
17	99.811	2.550
18	99.726	2.217
19	99.882	3.315
20	99.802	3.146
21	99.635	2.520
22	99.782	3.045
23	99.844	2.346
24	99.713	1.837
25	99.678	1.437

It is natural to observe that, the accuracy of the proposed dynamic firewall is 99.71% and time complexity is 2.256 ns mean.

Further, the results are visualized graphically [Fig – 4].

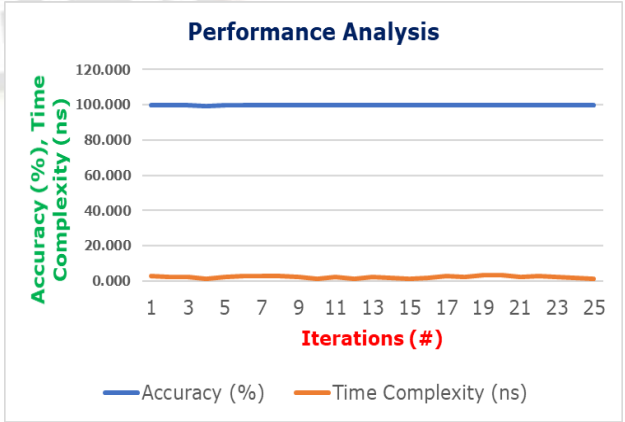

Fig. 4. Performance Analysis

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

VII. Comparative Analysis

In order to achieve the Furthermore, in the next section of this work, the obtained outcomes are compared with the parallel recent research outcomes.understanding of the improvement over the parallel research outcomes, this work results are compared with the recent researches [Table – 6].

TABLE VI. Comparative Analysis

Author, Year	Method	Accuracy (%)	Time Complexity (ns)	Model Complexity
W. Qiao et al.[10], 2021	Clustering	99.5	3.818	O(n3)
S. Shirali-Shahreza et al. [6], 2018	Clustering	95	3.769	O(n3)
Q. Cheng et al. [8], 2018	Classification	96.79	3.345	O(n3)
Proposed Method, 2022	Dynamic Threshold Based	99.71	2.256	O(n2)

Henceforth, it is natural to realize that, the proposed algorithms have outperformed the benchmarked parallel research outcomes.

The comparative analysis is visualized here [Fig – 5].

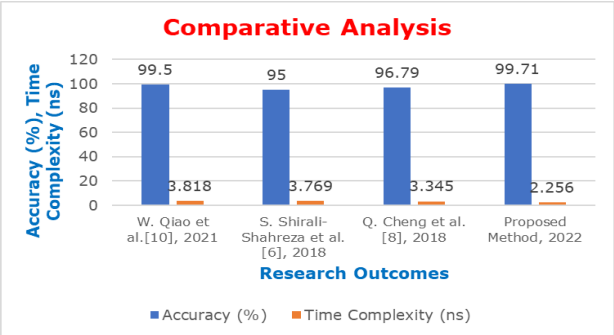

Fig. 5. Comparative Analysis

After realizing the improvements over the existing methods, in the next section of the work, the final research conclusion is furnished.

VIII. Research Conclusion

The expansion of cloud computing resource management features has resulted in an increase in the hazards connected with the services that are hosted in the cloud. Among the most difficult basic difficulties are: first, accommodating the ease of access without jeopardising security features; second, meeting time restrictions in order to offer service replies on time; and third, providing security to active services. A significant number of studies have been conducted in order to get the best firewall security for hosted services. Nonetheless, the present approaches are frequently criticised for their higher complexity as well as their lack of performance when applied to emerging attack types. As a result, the goal of this effort is to eliminate the bottlenecks that now exist in research. As a starting point, the suggested work intends to reduce the complexity of the deployed strategy by lowering the attribute sets in the sense of accuracy, time complexity, and information loss while maintaining high levels of accuracy. This study also provides a dynamic firewall rule engine design strategy for detection of attacks utilising the thresholding method, which is further developed in this work. The proposed algorithms are being tested on the benchmarked KDD dataset, and the results show that the accuracy is approximately 99.7 percent and the time complexity is decreased by over 40%. This proposed work, as a result, provides state-of-the-art technology for firewall design for hybrid cloud environments, and it will be recognised as a new benchmark in this field of research.

References

[1] S. Bagheri and A. Shameli-Sendi, “Dynamic Firewall Decomposition and Composition in the Cloud,” in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3526-3539, 2020.

[2] Y. -W. Chang and T. -N. Lin, “An Efficient Dynamic Rule Placement for Distributed Firewall in SDN,” GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6.

[3] T. Shangte, Y. Shufang, L. Shijian and Z. Xihui, “Research on the Dynamic Firewall of the Internet of Things,” 2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), Yilan County, Taiwan, 2021, pp. 78-81.

[4] L. Durante, L. Seno and A. Valenzano, "A Formal Model and Technique to Redistribute the Packet Filtering Load in Multiple Firewall Networks," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2637-2651, 2021.

[5] "N. Sophakan and C. Sathitwiriyawong, "A Secured OpenFlow-Based Software Defined Networking Using Dynamic Bayesian Network," 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 2019, pp. 1517-1522."

[6] "S. Shirali-Shahreza and Y. Ganjali, "Protecting Home User Devices with an SDN-Based Firewall," in IEEE Transactions on Consumer Electronics, vol. 64, no. 1, pp. 92-100, Feb. 2018".

[7] "G. Rezaei and M. R. Hashemi, "An SDN-based Firewall for Networks with Varying Security Requirements," 2021 26th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, 2021, pp. 1-7".

[8] "Q. Cheng, C. Wu, H. Zhou, Y. Zhang, R. Wang and W. Ruan, "Guarding the Perimeter of Cloud-Based Enterprise Networks: An Intelligent SDN Firewall," 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 2018, pp. 897-902."

[9] "J. Myers et al., "Dynamic Internet of Things Malware Detection Using Machine Learning (Work-in-Progress)," 2018 International Conference on Software Security and Assurance (ICSSA), Seoul, Korea (South), 2018, pp. 67-72".

[10] W. Qiao et al., "A Novel Method for Resource Efficient Security Service Chain Embedding Oriented to Cloud Datacenter Networks," in IEEE Access, vol. 9, pp. 77307-77324, 2021.

[11] R. Chanajitt, B. Pfahringer and H. M. Gomes, "Combining Static and Dynamic Analysis to Improve Machine Learning-based Malware Classification," 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), Porto, Portugal, 2021, pp. 1-10.

[12] S. Van Rossem, W. Tavernier, D. Colle, M. Pickavet and P. Demeester, "Profile-Based Resource Allocation for Virtualized Network Functions," in IEEE Transactions on Network and Service Management, vol. 16, no. 4, pp. 1374-1388, Dec. 2019.

[13] S. Pisharody, J. Natarajan, A. Chowdhary, A. Alshalan and D. Huang, "Brew: A Security Policy Analysis Framework for Distributed SDN-Based Cloud Environments," in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 6, pp. 1011-1025, 1 Nov.-Dec. 2019.

[14] T. Harada, K. Tanaka, R. Ogasawara and K. Mikawa, "A Rule Reordering Method via Pairing Dependent Rules," 2020 IEEE Conference on Communications and Network Security (CNS), Avignon, France, 2020, pp. 1-9.

[15] A. Oktadika, C. Lim and K. Erlangga, "Hunting Cyber Threats in the Enterprise Using Network Defense Log," 2021 9th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2021, pp. 528-533.

[16] V. Reddy, G. Garg, B. R. Tamma and F. A. Antony, "Interference Aware Network Function Selection Algorithm for Next Generation Networks," 2019 IEEE Conference on Network Softwarization (NetSoft), Paris, France, 2019, pp. 54-59.

[17] H. Vegda and N. Modi, "Secure and Efficient Approach to Prevent Ad Hoc Network Attacks Using Intrusion Detection System," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 129-133.

[18] M. Korkmaz, O. K. Sahingoz and B. Diri, "Detection of Phishing Websites by Using Machine Learning-Based URL Analysis," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7.

[19] S. T. Arzo, R. Bassoli, F. Granelli and F. H. P. Fitzek, "Study of Virtual Network Function Placement in 5G Cloud Radio Access Network," in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2242-2259, Dec. 2020.

[20] S. Ejaz, Z. Iqbal, P. Azmat Shah, B. H. Bukhari, A. Ali and F. Aadil, "Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function," in IEEE Access, vol. 7, pp. 46646-46658, 2019.

[21] V. Belenko, V. Krundyshev and M. Kalinin, "Intrusion detection for Internet of Things applying metagenome fast analysis," 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), London, UK, 2019, pp. 129-135.

[22] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA).