Journal of Theoretical and Applied Information Technology
15th March 2023. Vol.101. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A STRATEGY FOR SELECTIVE DYNAMIC FIREWALL AND PROACTIVE SECURITY METHODS ON CLOUD

[1]S REKHA GARIKAMUKKALA, [2]DR. V.RAVI SHANKAR

[1] Research scholar,Computer Science and Engineering,Gitam(Deemed to be University)
Hyderabad,India
sunitha.garikamukkala@gmail.com

[2]AssociateProfessor ,Computer Science and Engineering,Gitam(Deemed to be University)
Hyderabad,India
ravisankar.vadali@gitam.edu

ABSTRACT

The most crucial concern of modern cloud computing industry is the security. The security for traditional systems can be provided with tightly coupled components such as firewalls or tremendously strong authentication methods. But in the case of cloud computing, the extra delay caused by security mechanisms that are more complicated is not quite adaptable. The recent research outcomes have demonstrated significant improvements in terms of detection of the attack, however those methods are either formulated as reactive methods to reduce the time complexity or highly time complex if the methods are proactive. However, the security concerns must be addressed. In the recent past, a good number of researchers have aimed to solve this long persistent challenge. These outcomes from the parallel research, are highly time complex and cannot cater to cloud computing service level agreement demands. Thus, the proposed work analyzes the possibilities of reduction of model or framework complexities by introducing highly time efficient reduction method using the proposed correlation and further reduces the complexity of the firewalls using proposed regression method. The proposed framework establishes the proactive security framework with 93% accuracy in characteristics reduction, 25% reduction in analysis time and 99% accuracy in attack preventions.
Keywords:- *Collective Correlation, Dimensionality Reduction Regression, Dynamic Firewall, Rule Generator, Firewall Deployment*

1. INTRODUCTION

Firewalls filter hazardous traffic and ensure network security, according to S. Bagheri et al. [1]. Their performance is vital to the network. Most people utilise rule-based firewalls. Larger firewall rule lists increase lookup latency. Reordering rules based on traffic characteristics may minimise firewall packet matches, improving performance and Optimal firewall rule ordering NP-completeness (ORO). Therefore, a centralised firewall for the whole network is unachievable. Our technique dynamically scales firewalls across administrative domains to enhance rule optimization, filtering, and attack response. In this research, micro firewalls with their unique settings are used to perform firewall activities. Local and regional traffic is managed. Experiments show our solution is scalable for the company's network demands. On top of all that, the central firewall is inefficiently performing rules optimization algorithms in repeated time intervals.

Cloud computing provides flexible IT solutions for dynamically evolving companies. Various cloud deployment types provide different security solutions. Hybrid clouds mix public and private cloud resources and security. This paper models and evaluates a no-firewall option, conventional firewalls, and web-blocking firewalls for hybrid cloud installations. For small hybrid clouds (150 nodes), a basic firewall provided high point-to-point use without compromising response time. A web-filtering firewall is more effective for larger networks, according to H. Kurdi et al [2].

As cloud computing gained popularity, security has become increasingly crucial. New and rising dangers are slowing down the cloud-based company operations. Configuring Cloud firewalls is tricky

when integrating Cloud applications into an existing architecture. Technology enables as many ports as needed. A dangerous firewall approach requires dynamic testing. Majhi et al. [3] proposed a dynamic and reliable firewall for corporate cloud computing.

To avoid the wrath of government watchdogs, businesses must adhere to data privacy regulations. The rules of HIPPA and GDPR are generally followed. As a result, they ensure the safety and privacy of companies that operate in the cloud. To the extent that customer information is compromised, cloud service providers bear no responsibility. Authorities are very stringent in their regulations. There is a great deal at stake in the banking, healthcare, and insurance industries. embarrassment-inducing security lapses There was a widespread leak of data from cloud storage services. It is impossible to prevent a leak that is not intentional. There was a decline in their moral standing, reputation, and business opportunities. Due to cloud security measures, this is not possible. Information stored in the cloud can be accessed by employees and other authorized users [4-7].

The rest of the work is furnished such as the foundational methods for dimensionality reduction and the firewall buildings are furnished in Section – II and III, the parallel recent research outcomes are analyzed in Section – IV, the research problems are furnished in the Section – V, the proposed solutions are furnished in Section – VI and VII, the results and the comparative analysis on the obtained results are presented in the Section – VIII and Section – IX and finally the research conclusion is presented in Section – X.

2. FUNDAMENTALS OF DIMENTIONALITY REDUCTIONS

After setting the context in the previous section of this work, this section elaborates on the foundational strategy for building the reduced set.

Assuming that, the complete dataset, DS[], is the collection of n number of parameters denoted as A_i. Thus, this relation can be formulated as,

$$DS[] =< A_1, A_2, A_3, ...A_n > \quad \text{(Eq.1)}$$

Further, each parameter or attribute is a collection m number of values called the domain as D_i. Thus, this relation can be presented as,

$$A_i[] =< D_1, D_2, D_3, ...D_m > \quad \text{(Eq.2)}$$

Also, in the dataset, the decision parameter or the class variable is denoted as A_C.

Further, in order to reduce the dataset, the fundamental process suggests to calculate the influence scores for each parameter and further builds the reduced dataset, RDS[], with parameters with highest influences. Assuming that, λ is the arbitrary function to calculate the influence scores denoted by X_i for the parameter A_i.

Thus, this can be formulated as,

$$\lambda\{A_i :: A_C\} = X_i \quad \text{(Eq.3)}$$

And,

$$\text{s}\,\lambda\{A_j :: A_C\} = X_j \quad \text{(Eq.4)}$$

Further, if $X_i > X_j$, then the A_i must be part of the reduced dataset. Else, A_j must be part of the reduced dataset as,

$$\begin{aligned} &iff\ X_i > X_j, RDS[] \Leftarrow [A_i] \\ &Else\ RDS[] \Leftarrow [A_j] \end{aligned} \quad \text{(Eq.5)}$$

Thus, the final reduced dataset is built as per the foundational method.

The current work is also designed to carry out the research for dynamic firewall building. Hence in the next section of this work, the firewall design foundational principle is also analyzed.

3. FIREWALL DESIGN – FOUNDATIONAL METHOD

The design of the firewall completely depends on the set of attributes and design of the rule engine. The foundational method suggests that the firewall rule sets, FAS[], must be extracted from the dataset as,

$$FAS[] = \prod_{A_C = Attack} RDS[] \quad \text{(Eq.6)}$$

And, further the final firewall, DF[], can be designed as,

$$DF[] = \{FAS[]\} \quad \text{(Eq.7)}$$

Further, based on the foundational methods, in the next section of this work, the recent and most prominent research outcomes are analyzed.

4. LITERATURE REVIEW

This section of the work is dedicated for analysing the parallel research outcomes.

Cloud computing security has many vulnerabilities. Location-based firewalls with static rules require a long time to set up and aren't good for dynamic, virtualized apps. C. Decusatis et al. [4] used software-defined network overlays with forwarding graphs to provide virtual firewalls. Experimental results and performance data will be supplied using a virtual firewall deployment with an industry-standard virtual overlay network.

"Most African countries lack electricity, stifling economic progress. Managing renewable micro-grids requires a solid infrastructure that supports cloud integration for dynamic service delivery. We designed SGEMS for renewable micro-grids to optimise demand side management (DSM). The concept produces and records energy usage using solar PV. SGEMS includes an OpenStack Cloud application with active load-balancers, Trove/Hadoop Bigdata API, OpenFlow firewall, and dynamic network scalability. These features reduce the computational load of user access control. A fast demonstration of OpenStack/Amazon EC2 for transactional workload". K. C. Okafor and his colleagues [5] found the OpenStack engine useful as SGEMS-distributed energy's API middleware [5].

Network security becomes a significant and potentially game-changing application when combined with network function virtualization (NFV), which can "chain virtualized security network functions (VSNFs) such as IP address translation, spam filtering, and packet filtering for cloud datacenter networks. VSNF traffic flows are driven by mobile customers' security service demands, such as network security level, end-to-end latency, and security resource requirements. When placing VSNFs, it's challenging to avoid exceeding end-to-end latency limits and security service requirements due to the dynamic nature of cloud datacenter networks and routing security service paths that maximise resource consumption". It's also called dynamic embedding in the security service chain (SSC-DMP). This study presents an NFV-enabled architecture for incorporating SSC in cloud data centre networks. To solve the SSC-DMP issue, a small-scale ILP model is created. We provide a particle swarm optimization-based SSC dynamic embedding approach to reduce temporal network complexity. W. Qiao et al. [6], who did extensive simulations, say the recommended approach can outperform 35.2% and 23.1% of current benchmarks.

Cloud-based business networks are high-traffic and dynamic, making perimeter security problematic. Most cloud firewalls are inflexible, error-prone, and ineffectual due to static security rule setting or simple rule matching, resulting in major security risks. In this study, we present an artificially intelligent software-defined networks firewall (AI-SDNF). AI-SDNF may extract and analyse data packet content rather than only matching header fields (e.g., source and destination IP/MAC addresses). We train an intelligent SDN firewall using logistic regression and supervised machine learning to detect whether a packet is benign or dangerous. On OpenDaylight and OpenStack, an AI-SDNF prototype runs. We evaluate prototype performance and overheads using a real dataset. AI-SDNF achieves 96.79% detection accuracy with 0.2ms delay, according to Q. Cheng et al [7].

S. Arunkumar et al. [8] defines firewalls monitor and govern incoming and outgoing network traffic based on firewall rules. Firewalls are constantly improved to protect networks. The firewall is now an intrusion detection and prevention system. It evaluates existing firewall restrictions and their relevance in dynamic networks like political party coalitions. The paper suggests using a generative policy model to develop next-generation firewall rules.

Cloud orchestration manages and scales cloud resources to better meet user needs and service provider objectives. These traits, which foster cloud

platform adaptability, cause cloud service providers challenges. Virtual machine migration might cause inconsistencies and security issues. This article provides a formal framework for virtual machine migrations and security policy updates. Using this architecture, we can preserve the pre-migration global security policy. To do this, we build a cloud calculus that describes cloud computing architecture and firewall security requirements. You may establish virtual machine migration and security requirements. We define calculus semantics using structural congruence and a reduction relation. We use cloud-based terms to verify the new configuration's worldwide security policy. Jarraya and colleagues [9] provide an illustration of our technique's usefulness.

"IoT is widely used in corporate, home, and commercial sectors, and it will certainly continue to enhance our lives [10]. IoT botnets are a more dangerous Internet weapon. Mirai and Reaper have affected several industries and devices. This research protects consumer IoT devices using cloud-based machine learning and a dynamic on-site firewall".

Flexible network service deployment and Cloud-based orchestration are becoming increasingly versatile due to new technologies. In this environment, dynamic cloud configurations are hard to forecast. Service provider must relate SLA performance criteria to virtualized infrastructure resources. A VNF profile may speed up this process. We assessed four network services (virtual router/switch/firewall/cache server) using varied workloads and resource parameters. We compared model-building methodologies using the profiled datasets. The most accurate approach estimates service performance based on incoming workload and allocated resources. Our approach proposed resource allocations based on SLA performance targets and maximum workload. By delivering the softwarized service with the right number of resources, this helps prevent extraneous scaling steps, as explained by S. Van Rossem et al. [11].

SDN is adaptable to application deployment, dynamic network topology changes, and autonomous network management in multi-tenant data centres. SDN controller distribution systems may make deploying security solutions challenging, leading to policy conflicts and inconsistencies. Brew's security policy analysis approach relies on an OpenDaylight SDN controller's conflict detection and resolution module. In a distributed SDN-based cloud system, no two flow rules may clash at any layer, preventing information leaks. We provide techniques for decentralised global prioritising of flow rules, extend standard firewall rule conflict classification to SDN flow rule conflicts, and give ways for unaided resolution by recognising and categorising cross-layer conflicts. If administrators need to resolve disagreements, a unique visualisation tool helps them perceive the issues. S. Pisharody et al [12].

According to research by D. Bringhenti et al. [13], new technologies like Software-Defined Routing and Network Functions Virtualization make it simpler to automate traditionally manual procedures. These concerns lead to a novel technique to automatically allocate and configure virtual firewalls in a user-defined network service graph according to a comparable set of security requirements. The presented framework guarantees high confidence in the solution's correctness using a formal method based on the weighted partial MaxSMT issue. The viability of this strategy was tested using an implementation of the recommended technique built on top of the z3 solver.

SDN and NFV are appropriate for dynamically programmable network control functions and protocols, according to S. Ejaz et al[14]. SDN abstracts network resources using well-defined APIs, resulting in tenant networks with topology independent QoS and SLAs. NFV deploys VNFs using virtualization on commodity hardware. Using cloud based VNFs to build virtual IP services like load distribution, routing, and forwarding may improve network performance. This research used a vSDN controller as a VNF to balance traffic. Secondary vSDN controllers may spread vSDN's growing and uneven load. When a backup vSDN controller is needed, it is built with the same parameters as the original and shares traffic load balancing duties. Transparently installed in the cloud, all network clients are aware of the newly established secondary vSDN controller. This research tested load balancing in a Fat-Tree architecture using a Mininet emulator and two vSDN controllers. Average load was cut in half, mean delay was lowered by four-tenths of a second, and system performance increased in ping response time, bandwidth utilisation, and total throughput.

5. PROBLEM IDENTIFICATION

After an in depth analysis of the foundational methods and the parallel recent research outcomes from various sources, it is very conclusive that the following two problem are persistent in term of research bottlenecks to design effective dynamic firewall.

Firstly, the time complexity of the existing method for dimensionality reduction is high and without reducing the complexity is it nearly

impossible to deploy the dynamic firewall in reactive manner. The time complexity for the existing system can be calculated as, assuming that ϕ is the arbitrary function to calculate the size, n, of the dataset and this can be formulated as,

$$\phi\{DS[]\} = n \quad \text{(Eq.8)}$$

Thus, for calculating the influence scores for each parameter, the time complexity, t_1 can be furnished as,

$$t_1 = n \quad \text{(Eq.9)}$$

Also, for checking the higher influence of any parameter with other parameters, the time complexity t_2 can be calculated as,

$$t_2 = n.(n-1) = n + n^2 \approx n^2 \quad \text{(Eq.10)}$$

Hence, the total time complexity, T, for dimensionality reduction can be formulated as,

$$T = t_1 + t_2 == n + n^2 \approx n^2 \Rightarrow O(n^2) \quad \text{(Eq.11)}$$

Naturally this is significantly high for a dynamic firewall deployment model.

Secondly, during the design of the firewall rules, if the complete dataset has only positive records as,

$$RSD[] \approx \prod_{A_C = Attack} RDS[] \quad \text{(Eq.12)}$$

Then,

$$RSD[] \approx FAS[] \approx DF[] \quad \text{(Eq.13)}$$

Hence, the firewall will result into a very heavy set of rules to be deployed dynamically.

The solutions to these two problems are furnished in the next section of this work.

6. PROPOSED SOLUTIONS – MATHEMATICAL MODEL

After the detailed realization of the existing research problems, this section of the work, elaborates on the proposed solutions using the mathematical modelling method.

Firstly, the proposed collaborative attribute reduction process for dimensionality reduction is furnished here.

Continuing from Eq. 3, the proposed method defines a new method for calculating the influence score for each parameter with the conclusive reflection of the parameter influence not only with the class variable, rather along with the class variable with other parameters also, which are available in the dataset as,

$$\lambda\{A_i\} = \frac{DS[] - A_i}{DS[]}.A_i :: A_C \Rightarrow X_i \quad \text{(Eq.14)}$$

Similarly for all the parameters in the dataset, the influences scores must be calculated and kept in the influence score collection, X[].

$$X[] = < X_1, X_2, ...X_n > \quad \text{(Eq.15)}$$

As the influence scores are calculated with correlation with other attributes in the dataset, thus the higher influence score not only defines the attribute individual influence on the class variable, rather also defines significant influence of that attribute over other attributes. Thus, sorting the collection X[] will naturally provide the set of attributes with higher relative influences as,

$$RDS[] = Sort\{X[] : DS[]\}_{T \to 0}^{A \to 100\%} \quad \text{(Eq.16)}$$

The final selection of the attributes must be calculated based on the optimal or lowest time complexity, T, and optimal or highest accuracy, A, for classification using any standard methods.

Considering that, the new size of the reduced dataset is k as,

$$\phi\{RDS[]\} = k \quad \text{(Eq.17)}$$

Secondly, the dynamic firewall design process using the regression method is furnished here.

As per the initial assumptions, the A_C is the class variable, and the following regression formulation can be established using the regression coefficients, β_i as,

$$FAS[] = \{A_C \rightarrow \beta_0 + \sum_{i=1}^{k} \beta_i . A_i\} \quad \text{(Eq.18)}$$

Further, the regression coefficients can be calculated as,

$$\beta_i = \frac{\{\sum(RDS[] - A_i).(\sum A_i^2[])\} - \{(\sum A_i[]).(RDS[])\}}{k.(\sum A_i^2[]) - (\sum A_i[])^2} \quad \text{(Eq.19)}$$

Thus, the new proposed dynamic firewall, DF[] is just set of rules instead of data samples present in the existing method.

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

7. PROPOSED ALGORITHM AND FRAMEWORK

After the justification and elaboration of the proposed methods, in the previous section of this work, in this section the proposed algorithms and the proposed framework is furnished.

Firstly, the Collective Correlation based Dimensionality Reduction (CC-DR) Algorithm is furnished here.

Algorithm - I: Collective Correlation based Dimensionality Reduction (**CC-DR**) Algorithm

Input:

Network Characteristics KDD Dataset for Cloud as DS[]

Output:

Reduced Network Characteristics Dataset as RDS[]

Process:

Step - 1. Load the dataset as DS[]

Step - 2. For each element in DS[] as DS[i]

a. Calculate the influence factor X[i] for DS[i] using Eq. 14

Step - 3. Finalize the influence factor collection as X[]

Step - 4. For each element in X[] as X[j]

a. If X[j] > X[j+1]

b. Then,

t = X[j], X[j] = X[j+1], X[j+1] = t

Step - 5. For each element in X[] as X[k]

a. Calculate the accuracy as

A(k) = KMeans(X[0..(k-1)]

b. Calculate the time as

T(k) = KMeans(X[0..(k-1)]

c. If A(k) > A(k-1) && T(k) < T(k-1)

d. Then, RDS[r] = {X[k]::DS[]}

Step - 6. Return RDS[]

Statistical methods that reduce the number of dimensions is often used in data visualization. But same methods may be employed in applied machine learning to streamline a dataset for classification or regression in order to better build a prediction model.There might be a vast volume of data in a feature space with many dimensions, therefore the points in the space (rows of data) frequently reflect a tiny and unrepresentative sample.

Secondly, the Regression Based Dynamic Firewall Rule Generator (RB-DF-RG) Algorithm is furnished.

Algorithm – II: Regression Based Dynamic Firewall Rule Generator (**RB-DF-RG**) Algorithm

Input:

Reduced Network Characteristics Dataset as RDS[]
Output:
Firewall Rule Engine as FAS[]
Process:
Step - 1. Read the reduced dataset as RDS[]
Step - 2. For each element in RDS[] as RDS[i]
a. Calculate the B(i) using Eq. 19
Step - 3. Build the regression calculation FAS[] using Eq. 18
Step - 4. Return FAS[]

A firewall is vital to any security architecture because it transfers host-level defenses to your network security device. Next Generation Firewalls focus on malware, application-layer threats, and an integrated intrusion prevention system (IPS) to identify and react to network-wide assaults. Set rules to secure your network and undertake fast assessments to spot invasive or dubious behavior, like malware.

Finally, the Dynamic Firewall Deployment (DFD) Algorithm is furnished here.

Algorithm - III: Dynamic Firewall Deployment (**DFD**) Algorithm
Input:
Network Characteristics as N[]
Firewall Rule Engine as FAS[]
Time instance as t
Output:
Activate Firewall as DF[]
Process:
Step - 1. Monitor the network characteristics as N[]
Step - 2. For each instance of N[] as N[t]
a. If N[t].DS[] similar to FAS[]
b. Then, active DF[t]
Step - 3. Deploy DF[t]

In today's environment, the phrase network monitoring is common across the IT business. All networking components, including routers, switches, firewalls, servers, and virtual machines (VMs), are constantly monitored for faults and performance to ensure their availability. The proactive nature of network monitoring is an essential consideration. Proactively identifying performance issues and bottlenecks helps in the early detection of problems. Efficient proactive monitoring helps avoid network downtime or failures.

Henceforth, based on the proposed algorithms, the proposed framework is furnished in Fig. 1.

Further, the obtained results from the proposed algorithms are furnished in the next section of this work.

8. RESULTS AND DISCUSSIONS

After the detailed discussion on the proposed algorithms in the previous section of this work, in the section of the work, the obtained results are furnished.

Firstly, the description of the dataset is furnished in Table I.

TABLE I. Initial Dataset Description

Parameter Name	***Value***
Number of Instances	3500
Number of Attributes	54
Number of Instances with Attacks	2985
Number of Instances without Attacks	515
Number of Missing Values	3
Number of Outlier Values	5

Henceforth, it is natural to realize that the selected KDD dataset [15] for cloud is the perfect dataset for this analysis. Also, the dataset is highly adapted due to the fact that this dataset is nearly 0% influenced by missing values and outlier.

The further analyses are carried on the live network for nearly 100 times, however for the representation purposes only 10 are listed here.

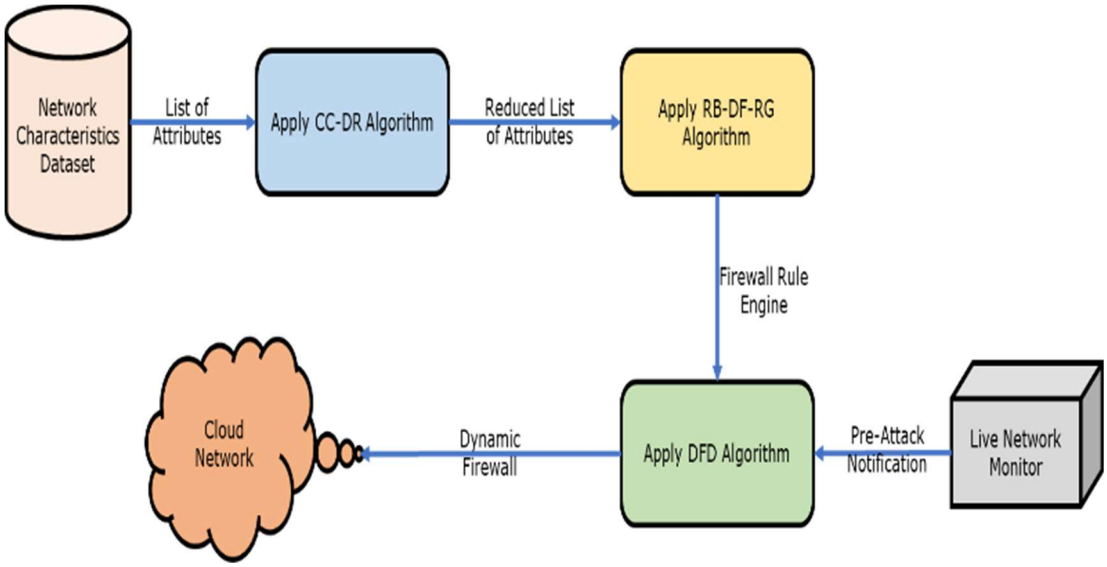

Fig. 1. Proposed Framework

Secondly, the attribute reduction process outcomes are furnished in Table II.

TABLE II. Attribute Reduction Process

Test Sequence ID	*Number of Attributes*	*Accuracy with Reduced Set (%)*	*Time for Clustering with Reduced Set (ns)*
1	48	53.95	10.019
2	45	87.18	19.822
3	42	87.18	16.862
4	35	87.18	13.195
5	31	90.19	12.008
6	30	91.71	15.612
7	25	92.37	13.663
8	16	93.25	14.170
9	15	92.14	18.156
10	14	89.25	19.397

It is clear to observe that, the optimal reduced set is achieved at the 8th iteration with the help of accuracy and time complexity optimization. The results are visualized graphically in Fig. 2.

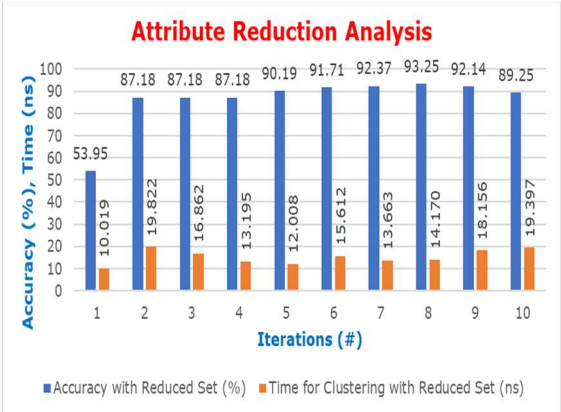

Fig. 2. Attribute Reduction Analysis

Further, the sample rule engine details along with few sample rules are furnished [Table – 3].

Journal of Theoretical and Applied Information Technology
15th March 2023. Vol.101. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE III. Sample Rulesets

Generated Rule Engine (25 Sample)
Rule- 1. "dst_host_diff_srv_rate <= 0.01 AND count <= 27 AND service = ftp_data,3"
Rule- 2. "service = X11 AND dst_host_diff_srv_rate <= 0.08,2"
Rule- 3. "service = private AND protocol_type = udp AND count <= 151 AND count <= 7,4"
Rule- 4. "dst_host_diff_srv_rate > 0 AND dst_host_diff_srv_rate <= 0.01 AND protocol_type = tcp AND count > 7,4"
Rule- 5. "dst_host_diff_srv_rate > 0.08 AND service = ecr_i AND count <= 1,3"
Rule- 6. "dst_host_diff_srv_rate > 0.08 AND service = smtp AND dst_host_diff_srv_rate <= 0.16 AND dst_host_diff_srv_rate > 0.14,4"
Rule- 7. "dst_host_diff_srv_rate > 0.08 AND service = private AND protocol_type = udp AND count > 21,4"
Rule- 8. "dst_host_diff_srv_rate > 0.08 AND count > 20 AND service = private,3"
Rule- 9. "dst_host_diff_srv_rate > 0.08 AND service = private AND protocol_type = tcp,3"
Rule- 10. "count > 49 AND protocol_type = tcp,2"
Rule- 11. "service = other AND dst_host_diff_srv_rate > 0.97,2"
Rule- 12. "dst_host_diff_srv_rate <= 0.01 AND service = http AND count > 2 AND count <= 5,4"
Rule- 13. "count > 19 AND service = finger,2"
Rule- 14. "service = ftp_data AND dst_host_diff_srv_rate > 0.14 AND count <= 2 AND count <= 1,4"
Rule- 15. "service = http AND dst_host_diff_srv_rate > 0.01 AND count <= 12 AND dst_host_diff_srv_rate <= 0.04,4"
Rule- 16. "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 > 0.28,4"
Rule- 17. "service = other AND protocol_type = tcp AND dst_host_diff_srv_rate <= 0.18,3"
Rule- 18. "service = smtp AND dst_host_diff_srv_rate > 0.03 AND dst_host_diff_srv_rate <= 0.05,3"
Rule- 19. "service = ecr_i AND count <= 1,2"
Rule- 20. "service = http AND dst_host_diff_srv_rate > 0.01 AND count > 12,3"
Rule- 21. "service = ftp_data AND dst_host_diff_srv_rate <= 0.24,2"
Rule- 22. "service = other AND protocol_type = udp AND dst_host_diff_srv_rate > 0.07 AND count <= 41,4"
Rule- 23. "service = other AND dst_host_diff_srv_rate > 0.47 AND protocol_type = udp AND dst_host_diff_srv_rate <= 0.73,4"
Rule- 24. "service = http AND count > 1 AND dst_host_diff_srv_rate > 0 AND dst_host_diff_srv_rate <= 0.03,4"
Rule- 25. "service = http AND count <= 46,2"

Further, the accuracy of the cloud-based network attack prevention process using the dynamic firewall is furnished here [Table – 4].

TABLE IV. Attack Prevention Accuracy Analysis

Test Sequence ID	Number of Attacks Generated	Number of Attacks Prevented	Accuracy (%)
1	5685	3411	99.60
2	7354	3677	99.50
3	9051	4525	99.50
4	5655	3393	99.60
5	9931	2979	99.30
6	6053	3632	99.60
7	7955	2386	99.30
8	9716	8744	99.90
9	4947	1979	99.40
10	7979	3192	99.40

It is clear to observe that, the proposed dynamic firewall has prevented nearly 99% attacks in all iterations with a mean accuracy of 99.51%. The results are also visualized graphically [Fig – 3].

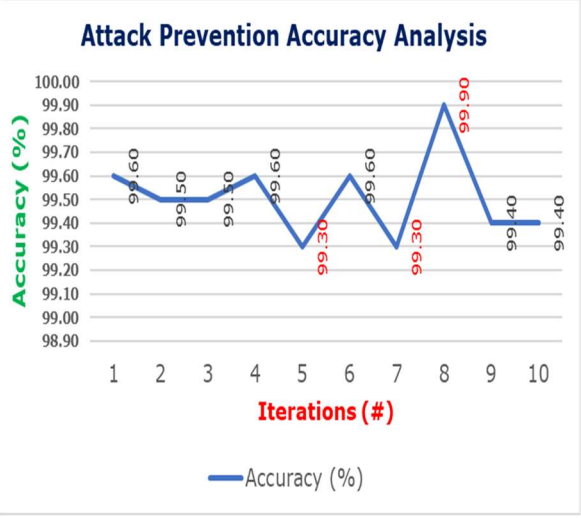

Fig. 3. Attack Prevention Accuracy Analysis

Finally, the time complexity of the detection process is also analyzed [Table – 5].

TABLE V. Time Complexity Analysis

Test Sequence ID	Time to Build Firewall (ns)	Time to Deploy Firewall (ns)	Total Time (ns)
1	2.584	1.968	4.552
2	3.773	3.729	7.502
3	3.079	4.309	7.388
4	2.307	3.210	5.517
5	4.020	2.014	6.034
6	5.529	1.463	6.992
7	5.312	3.177	8.489
8	3.589	4.375	7.964
9	3.155	2.668	5.823
10	5.094	1.475	6.569

It is natural to realize that the time complexity of the entire operational framework is highly similar in all iterations with a mean time complexity of 6.683 ns.

Further, in the next section of this work, the obtained results are compared with the recent benchmarked research outcomes in the next section of this work.

9. COMPARATIVE ANALYSIS

Further, the obtained results are compared with the recent benchmarked outcomes and the findings are furnished here [Table – 6].

TABLE VI. Comparative Analysis with Benchmarked Researches

Author, Year	Proposed Model	Model Complexity	Accuracy (%)
S. Bagheri et al. [1], 2020	Regression	$O(n^2)$	78.54
W. Qiao et al. [6], 2021	Regression	$O(n^2)$	83.29
D. Bringhenti et al. [13], 2020	Classification	$O(n^2)$	81.26
S. Ejaz et al. [14], 2019	Clustering	$O(n^2)$	88.76
Proposed Method	Correlation & Regression	O(n)	99.51

It is natural to realize that the obtained results have outperformed most of the parallel research outcomes and with this realization, in the next section of this work, the final research conclusion is furnished.

On the other hand, the proposed work highly relies on the available server traces, which makes the proposed method highly dependent on the training phase during implementation. Hence, the initial response time from the proposed method is slightly delayed.

The proposed method relies deeply on the regression methods and the benefits of such implementations are building local infrastructure in multiple areas is technically impossible and unstable. Infrastructure upgrades and maintenance require a distinct crew. Regression testing for cloud security can prevent this added overhead. Teams from any location or time zone can access the test environment. Regression testing on the cloud promotes team collaboration because other team members may see others' testing activity.

10. CONCLUSION

In order to find the most optimal framework for proactive and preventive light weight dynamic firewall for cloud computing enabled networks, a unique Collective Correlation-based Dimensionality Reduction technique was developed by the authors of this research as a means of effectively condensing network attribute datasets. As a consequence of these simplifications, it is now possible to construct a lightweight firewall that is both simple to configure and capable of being deployed in real time. In addition, a one-of-a-kind method known as the Regression-Based Dynamic Firewall Rule Generator is described and implemented in the process of designing the rule engine that will be used in the construction of the dynamic firewall. This work also demonstrates the installation of a dynamic firewall to block attacks on cloud-based networks, as well as the installation of regular network listeners or network monitors to detect assault scenarios. Both of these security measures are important in preventing attacks. This proposed work showcases the proactive security framework with 93.25% accuracy in characteristics reduction, 24.93% reduction in analysis time and 99.51% accuracy in attack preventions. Also, in comparison with various parallel research outcomes, this work is proven to be one of the benchmarked works.

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] H. Kurdi, M. Enazi and A. Al Faries, "Evaluating Firewall Models for Hybrid Clouds," 2013 European Modelling Symposium, Manchester, UK, 2013, pp. 514-519.

[3] S. K. Majhi and P. Bera, "Designing an adaptive firewall for enterprise cloud," 2014 International Conference on Parallel, Distributed and Grid Computing, Solan, India, 2014, pp. 202-208.

[4] C. Decusatis and P. Mueller, "Virtual Firewall Performance as a Waypoint on a Software Defined Overlay Network," 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS), Paris, France, 2014, pp. 819-822.

[5] K. C. Okafor, J. A. Okoye and R. M. Onoshakpor, "Towards Smart Green Energy Metering Design for OpenStack/Amazon Elastic Cloud Integration," 2019 IEEE PES/IAS PowerAfrica, Abuja, Nigeria, 2019, pp. 328-333.

[6] 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.

[7] 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.

[8] S. Arunkumar et al., "Next generation firewalls for dynamic coalitions," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 2017, pp. 1-6.

[9] Y. Jarraya, A. Eghtesadi, M. Debbabi, Y. Zhang and M. Pourzandi, "Cloud calculus: Security verification in elastic cloud computing platform," 2012 International Conference on Collaboration Technologies and Systems (CTS), Denver, CO, USA, 2012, pp. 447-454.

[10] 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.

[11] 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.

[12] 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.

[13] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Automated optimal firewall orchestration and configuration in virtualized networks," NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2020, pp. 1-7.

[14] 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.

M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set, 2022.