Multi Constraint Rule Centric Trust Analysis Model for Efficient Intrusion Detection for Improved QoS

Godsaint,Onoja Joshua*
Assistant Professor,
Department of Information Technology,
Federal University of Technology Akure,
Dr. Sabia Abubakar
Professor & Senior Innovator (IHub),
Department of Electrical & Electronics Engineering,
Federal University of Technology,Minna-Niger State,
Akpan, Abundance M
Department of Computer Science
Caritas University, Amorji-Nike, Enugu State.

Abstract-Intrusion attacks are the worst entity in any service orient environment which targets not just the services but also the Quality of service of the environment. We focused on detecting intrusion attack on SOA with the support of user defined rule sets as well as behavior of users in accessing services. Towards this issue, there are many techniques recommended around service features like payload, latency, and so on. However they struggle to detect the attacks in accurate manner. This article defines a novel Multi Constraint Rule Centric Trust Analysis Model (MCRTAM). The model allows the controller in defining set of rules and maintains the history of different access made by different users earlier. Initially, the method generates set of ensemble rules and preprocesses the traces to generate behavior rule sets. This has been performed for unique services and the service request and service packets are fetched. Using the features of service data, the method performs Base Rule Trust Analysis (BRTS), Service Model Trust Analysis (SMTA) and Service Access Trust Analysis (SATA). The BRTS process estimates Base Rule Trusts Measure (BRTM), SMTA process estimates Service Constraint Trust Measure (SCTM) and SATA process estimates Service Centric Access Trust (SCAT). Using all these values, Multi Constraint Trust Measure (MCTM) is measured. With the MCTM value, presence of intrusion attack is detected. The MCRTAM model hikes the intrusion detection accuracy and achieves higher QoS performance.

Index Terms-SOA, Intrusion Attack, Intrusion Detection, Rule Set, MCRTAM, BRTS, SMTA, SATA, MCTM.

I. INTRODUCTION:

The growing information technology supports the organizations to perform their most processes through web based solutions. Around this, there are number of services being designed to support the employees of organization to perform their routine work. In any service orient architecture (SOA); there will be number of services provided for the access of employees. However, any service orient environment has the threat of intrusion attack which is generated by some malicious users to intrude into the system in the sense to steal the information as well as to damage the performance of entire system as well as QoS of the environment.

Presences of intrusion attack are identified based on several features and methods. The frequency based approaches counts the frequency of service access and based on that the intrusion attack are detected. Similarly, there are methods which consider the protocols of the service to detect the threat. On the other side, the behavior of the users in accessing the service is used in detecting the threat. The payload based approaches consider the payload value being submitted to the account, where the hop count based approaches are used in detecting intrusion attack. However, the methods suffer in achieving higher accuracy in detecting the threat.

The rule based approaches are used in several occasions which uses set of rules in accessing the services. But in reality, the static rules are not enough as the adversaries are capable of tracking such rule behaviors and would modify their strategy in generating the threats. So, the rules must be highly dynamic and by monitoring the behavior of the system, the administrator or the controller must be capable of defining set of rules or adding set of rules to the rule set to make the system more rigid on finding attacks. Also, finding the threat just with few features are not efficient and it is necessary to consider most features in the ensemble. Similarly, measuring trust value for any user should be performed in multiple ways. This article focused on defining such approach to handle this issue.

In this point, an efficient multi constraint rule centric trust analysis model (MCRTAM) is presented. The model evaluate the user trust in three directions, one is by applying Base Rule Trust Analysis (BRTA) which analyze the trust of any user according to the constraints mentioned in the base rule set. Similarly, the Service Model Trust Analysis (SMTA) is performed to measure the trust of user in following the prototype of service. Finally, the model applies Service Access Trust Analysis (SATA) which analyzes the behavior trust of the user in accessing the service. The detailed approach is discussed in the next section.

II. Literature Survey

Problem of intrusion attacks are handled with several techniques and this section analyzes some of the methods in detail.

A deep blockchain framework (DBF) [1], enable intrusion detection in cloud IoT networks. The method identifies malicious events according to the learning ensemble to ensure data security.

The application of blockchain in intrusion detection is discussed in [2], and the article analyzes the challenges and solutions around the problem. Various attacks are considered and each has been analyzed for their possibility with the blockchain. A virtual machine introspection (VMI) orient VMGuard [3], mitigate known attacks in cloud environment. The model monitors various events happening in the environment and alerts the malicious behaviors over sensitive data.

Towards mitigating brute force and DDoS attacks, an efficient intrusion detection system is presented in [4]. A blockchain based distributed intrusion detection scheme (DIDS) [5], enforce blockchain as communication model in cloud environment. The random forest algorithm based intrusion detection model [6], uses the data obtained from tcpdump and applies data mining technologies to collect the network traffic. Finally, random forest algorithm is used to perform classification.

Honeypot networkbased intrusion detection scheme [7], designed to improve the security in cloud environment. The honeypot network is used in sharing the files in the environment. A cloudbased intrusion detection system is sketched in [8], towards security development in cloud environment. A deep blockchain framework is presented in [9], towards security development in IoT networks. The model uses blockchain with smart contracts in communication and intrusion detection is enforced with bidirectional long shortterm memory (BiLSTM).

The Network Intrusion Detection System as a Service (NIDSaaS) [10], support the OpenStack cloud.

A Swarm Intelligence Based Feature Selection model is sketched in [11], towards performing intrusion detection in cloud environment. The method uses whale optimization algorithm in classification. Different intrusion detection systems are analyzed and evaluated for their performance in [12,13]. The methods are analyzed for their performance with various data sets. Cuckoo search with ANN based IDS [14], uses cuckoo search in features selection and ANN towards intrusion detection.

A security infrastructure for vehicular information in SDN (DEMO) [15], secure the vehicular communication with uses the cyber defense centers in performing intrusion detection.

Virtual machine interface based IDS [16], consider the system level calls and process calls as the key in detecting intrusion attack. A smart villa intrusion alert system is briefed in [17], to support cloud IoT. The method senses the entry, location detection of various objects to produce alarm. Deep reinforcement learning based IDS [18], enforce fine grained classification of complex attacks. A deep learning based kernalized support vector machine is presented in [19], to support the detection of multi stage jamming attack in radio networks. The method use multi-stage machine learning-based intrusion detection (ML-IDS) to detect the jamming attacks. The brute force based intrusion detection scheme [20], uses various classes of attacks and monitors the network for various activities to perform intrusion detection. Towards detecting stepping store intrusion (SSI) is presented in [21], which uses efficient algorithm to detect outliers in the environment using k means clustering.

A dual branch network is presented in [22], which combines AoI segmentation and pedestrian object detection to perform intrusion detection according to the relative position. Generative adversarial network based deep learning intrusion detection model [23], performs feature selection from the network traffic and intrusion detection is performed with GAN. A Protocol Based Deep Intrusion Detection (PB-DID) is presented in [24], which learn the IoT traffic and perform classification accordingly. A principal component analysis based drift technique is presented in [25], which consider the variance of features to perform intrusion detection.

The approaches explored in this section are not optimal in achieving effective security performance in SOA and introduce poor performance in detecting intrusion attacks.

III. Problem Statement

The problem of intrusion detection in service orient environment is briefly explored. The intrusion detection problem has been approached by several techniques where each considers different features in detecting the intrusion attack. Data mining and machine learning approaches like random forest, deep enforcement learning and cuckoo search, are used in detecting such threats. Also, block chain based techniques are enforced for the detection of intrusion attack. However, the performance of the methods is depending on the feature being selected. Around this, the existing approach does not produce higher accuracy in detecting intrusion attack. All this issues encourages the author in designing a novel strategy in detecting malicious threats by choosing optimal features towards QoS development.

IV. Methodology

The proposed Multi Constraint Rule Centric Trust Analysis Model (MCRTAM) model fetch the rule set and access history of various users towards different services. First, the set of rules are generated as ensemble rules and preprocess the traces to generate behavior rule sets. With the rules, the method performs various analyses like Base Rule Trust Analysis (BRTS), Service Model Trust Analysis (SMTA) and Service Access Trust Analysis (SATA) according to the service request received. The result of various analyses is used in measuring Multi Constraint Trust Measure (MCTM) value. With the MCTM value, presence of intrusion attack is detected.

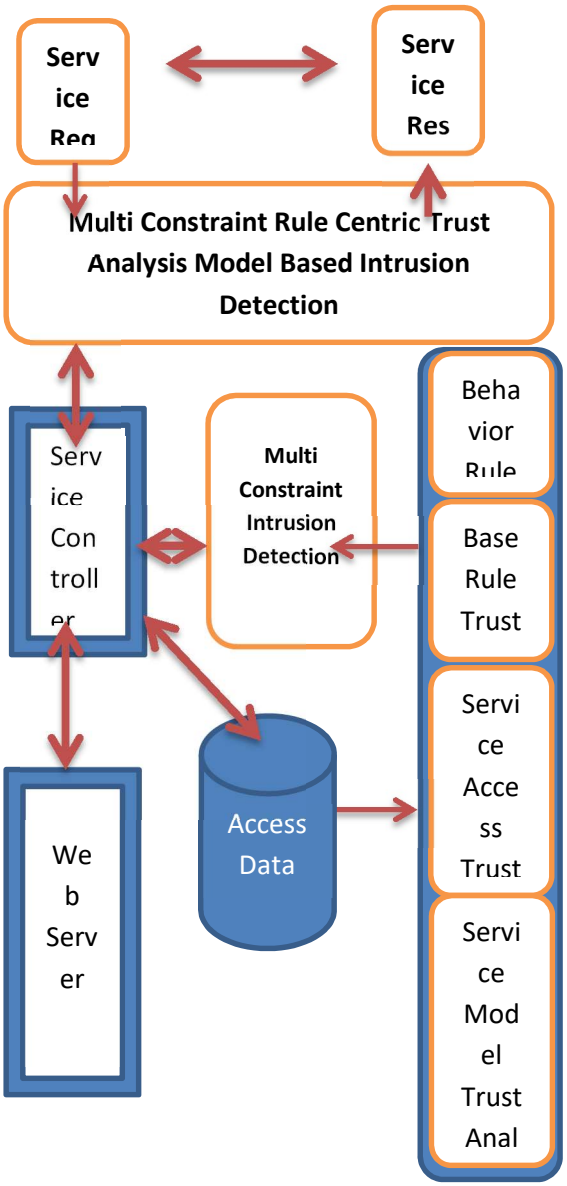

Figure 1: Architecture of MCRTAM Model

The functional architecture of MCRTAM model is presented in Figure 1, and the components of the model are discussed in detail in this section.

A. Behavior Rule Generation:

The access traces available in the model has been used to generate the behavior rule sets. It has been performed by preprocessing the traces available and finds the set of features in the trace. According to the features identified the method traverse through each trace and verifies the containment of each feature to be present. If the trace Ti does not contain any of the feature then it has been removed from the set. Further, the method computes Max Min values for each of the feature. Now, according to the min max values of different features, the method generates ensemble rule. This will be done for each service and by splitting the traces according to the service, different rules has been generated and added to the behavior rule set.

Pseudo Code of Rule Generation:
Given: Service Access Trace SeaT, Service Set Ses
Obtain: Rule Set Ers
Start
Read SeaT, SeS.
For each service S

$$\text{ServiceTracesST} = \sum_{i=1}^{size(SeaT)} SeaT(i).Service == S$$

$$\text{Find feature lists Fl} = \sum_{i=1}^{Size(ST)} (Features(ST(i)) \ni Fl) \cup Fl$$

For each trace Ti
If Ti$\ni \forall Features(Fl)$) then
ST = ST $\cap$ Ti
End
End
For each feature F

$$\text{Compute Max value Mv} = Max(ST(i).F)_{i=1}^{size(ST)} \quad \text{Compute Min value Mnv} = Min(ST(i).F)_{i=1}^{size(ST)}$$

End
Generate Ensemble behavior rule Ebr = {F1.min,F1.max,,Fn.min,Fn.max}
Add to rule set Ers = $\sum(Rules \in Ers) \cup Ebr$
End
Stop

The rule generation algorithm fetch the traces and finds distinct traces of various services. With the service traces, the method finds the features and normalized the data. Wiith the normalized values, the method computes maximum and minimum values. Using these values, the method generates ensemble behavior rules to support intrusion detection.

B. Base Rule Trust Analysis:

The model maintains set of base rules which are generated by the administrator of the entire system. Any rule in this set has been defined with set of properties and values. For any service, the model maintains set of rules which has been used in verifying the trust of the service request. From the service request, the method extract different features, and according to the features being extracted, the method computes Base Rule Trust Measure (BRTS). It has been measured by counting number features followed the rule, and number of them are not followed. Using these values, the method computes the value of BRTS. Estimated value of BRTS has been used towards intrusion detection.

Pseudo Code of BRTA:
Given: Base Rule Set BRS, Service Claimed SC
Obtain: BRTS
Start
Read BRS and SC.
Find the rule for service SC as SR.

$$\text{SR} = BRS(i).Service == SC_{i=1}^{Size(BRS)}$$

Compute Rule support RS = $\frac{\sum_{i=1}^{Size(SR)} SC(i).value <> S\ (i).Min,Max}{size(SR)}$

Compute Rule Deviation Measure RDM = $\frac{\sum_{i=1}^{Size(SR)} Dist(SC(i).value,SR(i).Min,Max) > SR(i).Min,Max/3}{size(SR)}$

Compute BRTS = $\frac{RS}{RDM}$

Stop

The base rule trust analysis function estimates the rule support measure and rule deviation measure. Based on these values, the BRTS is estimated to support intrusion detection.

C. Service Model Trust Analysis:

The trust of user in accessing the service can be measured according to the service model. Any service has its own model which is decided by the protocol and it is distinct for the service. The services has own number of input, type and output. Such protocol must be followed and based on that the trust of user in accessing the service according to the model can be measured. The method computes the value of Service Constraint Trust Measure (SCTM) towards intrusion detection. It has been measured by computing protocol support, value support, and data support values. With all these values, SCTM is estimated to perform intrusion detection.

Pseudo Code of SMTA:

Given: Service Claimed SC, Service Data Sdat

Obtain: SCTM

Start

Read Sdat. and SC.

Compute Protocol support Psup = $\frac{\sum_{i=1}^{Size(SC.Mdata)} Sc.Mdata(i) \in Sdat}{size(SC.Mdata)}$

Compute Value Support Vsup = $\frac{\sum_{i=1}^{Size(SC.Mdata)} Sc.Mdata(i) \in Sdat \&\& Sc.Mdata(i).Type == Sdat(i).Type}{size(SC.Mdata)}$

Compute SCTM = Psup×Vsup

Stop

The service model trust analysis algorithm measure protocol support and value support for the user request. According to the value of value support and protocol support, the method computes the value of SCTM and performs intrusion detection.

D. Service Access Trust Analysis:

The service access trust analysis algorithm measure the trust of user according to the behavior of user in accessing the service. It has been verified based on two constraints like the frequency of access and the fitness access. The frequency service access support is measured based on the value of frequency accessed in earlier time and current time with the max min values available in the rule. Similarly, the service access fitness (SAF) is measured according to the number of complete access and number of hanging access. According to SFS and SAF, the SCAT (Service Constraint Access Trust) is measured to perform intrusion detection.

Pseudo Code of Service Access Trust Analysis:

Given: Behavior Rule Set BRS, User U, Service S, Service Access Trace SaT.

Obtain : SCAT.

Start

Read BRS and U.

Find user trace towards service S as USaT = $\sum_{i=1}^{size(SaT)} SaT(i).Servicer == S \&\& SaT(i).User == U$

Compute Service Frequency Support SFS = if($\frac{\sum_{i=1}^{Size(UsaT)} UsaT(i).Time == Ti}{size(UsaT)} < BRS.S.FrequencyMin, FrequencyMax) >)? 1: 0$

Compute Service Access Fitness SAF = $\frac{\sum_{i=1}^{Size(USaT)} USaT(i).State == Complete}{size(USaT)} \times \frac{1}{\sum_{i=1}^{Size(USaT)} USaT(i).State == Hanging}{size(USaT)}$

Compute SCAT = SFS×SAF

Stop

The service access trust analysis algorithm computes different trust measures like service access fitness and service frequency support values to compute the value of SCAT. According to the value of SCAT, the method would perform intrusion detection.

E. Multi Constraint Intrusion Detection:

The proposed multi constraint intrusion detection algorithm performs several analysis towards detecting the intrusion attack. Initially, the access trace has been used to generate behavior rule sets and with the rule generated and the existing predefined rules, the method performs three analyses like Base Rule Trust Analysis, Service Model Trust Analysis and Service Access Trust Analysis. Using the result of all the three analyses, the method computes the value of Multi Constraint Trust Measure (MCTM). Based on the value of MCTM, the method performs intrusion detection.

Multi Constraint Intrusion Detection:

Pseudo Code of Multi Constraint Intrusion Detection:

Given: Access Trace set ATS, Rule Set Rs

Obtain: Boolean

Start

Read ATS, RS.

Behavior Rule set BRS = Perform Behavior Rule Generation (ATS)

BRTS = perform Base Rule Trust Analysis (BRS, ATS)

SCTM = perform Service Model Trust Analysis(BRS, ATS)

SCAT = perform Service Centric Access Trust Analysis (BRS, ATS)

Compute MCTM = $\frac{BRTS}{SCTM} \times SCAT$

If MCTM>Th then

Return true

Else

False

End

Stop

The above discussed algorithm measures MCTM and based on that the method classify the request as true or false which denotes trusted or malicious.

V. Results and Discussion:

The proposed model has been implemented using advanced java and has been tested for its performance with various

constraints like number of services, number of users and so on. In all the test cases, the methods are evaluated for their performance in different factors and mapped with others.

TABLE 1: EXPERIMENTAL SETUP

Parameter	Value
Tool Used	Advanced Java
Data Set	Amazon
Platform	Microsoft Azure
Total Records	5 lakhs
No of Services	50
No of Users	500

The constraints picked in analyzing the performance of methods is shown in Table 1. The methods are evaluated for their performance on the following factors:

TABLE 2: INTRUSION DETECTION PERFORMANCE

Analysis on Intrusion Detection Performance			
No of Records	3 Lakhs	5 Lakhs	10 Lakhs
BiLSTM	69	74	79
VMGuard	73	77	82
ANN	79	85	87
MCRTAM	87	93	98

The accuracy in detecting intrusion attack is evaluated with different number of records available. In each case, the proposed MCRTAM model achieves noticeable performance growth than others.

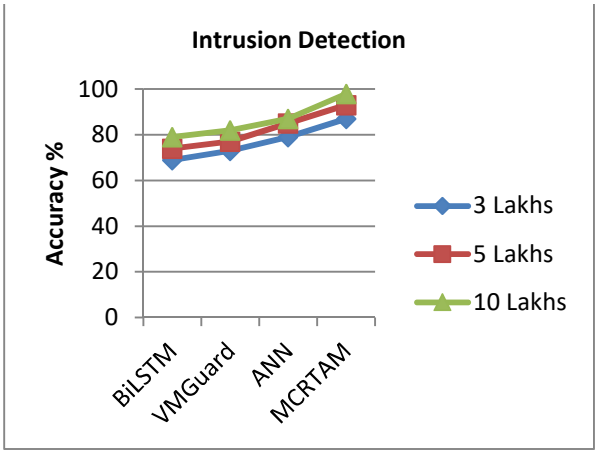

Figure 2: Analysis on intrusion detection accuracy

The performance of various methods at detecting intrusion attack is measured with the presence of various numbers of records in the data set. The MCRTAM scheme produces noticeable hike in detecting intrusion attack compare to other techniques.

TABLE 3: FALSE RATIO

False Ratio on Intrusion Detection			
No of Records	3Lakhs	5 Lakhs	10 Lakhs
BiLSTM	31	26	21
VMGuard	27	23	18
ANN	21	15	13
MCRTAM	13	7	2

The ratio of false classification in detecting intrusion attack is evaluated and shown in Table 3, where the MCRTAM scheme achieved less false ratio than others.

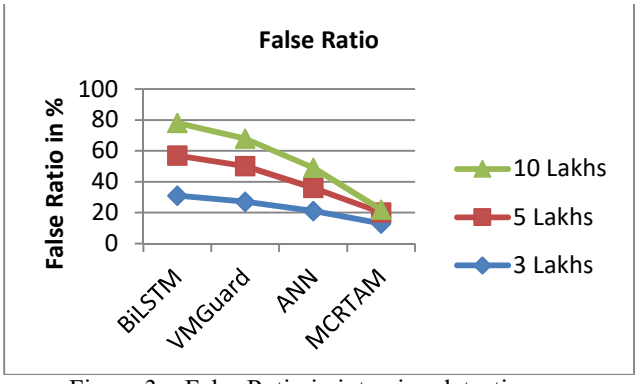

Figure 3: False Ratio in intrusion detection

The performance of various methods in producing false alarm is measured and compared in Figure 3. The MCRTAM model introduces less false detection compare to other techniques.

TABLE 4: TIME COMPLEXITY

Time Complexity on Intrusion Detection in Millie seconds.			
No of Records	3 Lakhs	5 Lakhs	10 Lakhs
BiLSTM	31	26	21
VMGuard	27	23	18
ANN	21	15	13
MCRTAM	13	7	2

The time complexity in detecting intrusion attack is evaluated and given in Table 4, where the MCRTAM model produced less time complexity compare to others.

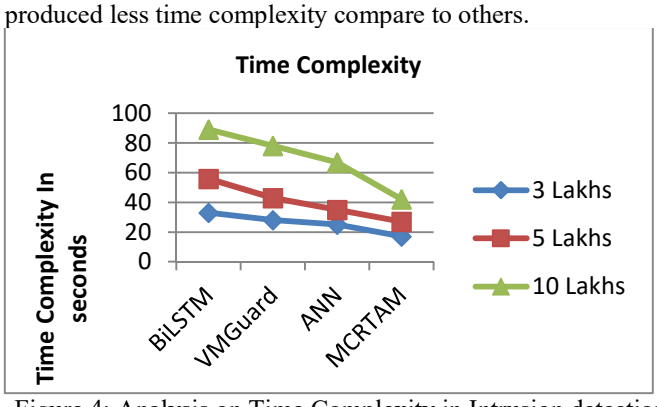

Figure 4: Analysis on Time Complexity in Intrusion detection

The efficacy of time complexity in intrusion detection is measured for different methods with different number of records are measured and presented in Figure 4. The proposed MCRTAM model introduces less time complexity than other approaches.

VI. Conclusion and Scope of Future work:

This article presented a novel multi constraint rule centric trust analysis model towards efficient intrusion detection against service orient environment like cloud. The model use both historic rule sets and behavior rule sets generated from the access trace maintained. Further, the trust analysis is performed in three ways like base rule trust analysis, service model trust analysis and service access trust analysis. Using the result of all the stages, the method computes the value of MCTM to support the detection of intrusion attack. The proposed MCRTAM model improves the performance of intrusion attack up to 97%.

Further, the problem of intrusion detection can be handled and improved for its accuracy by adapting social trust of any user in measuring the trust in accessing any service towards intrusion detection. By measuring the social trust of any user, the legitimacy of any user in global range can be analyzed which would be used in detecting intrusion attack with higher efficiency.

References:

[1] O. Alkadi, N. Moustafa, B. Turnbull and K. -K. R. Choo, "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks," IEEE (ITJ), Volume. 8, Number. 12, pp. 9463-9472, 2021.

[2] O. Alkadi, N. Moustafa and B. Turnbull, "A Review of Intrusion Detection and Blockchain Applications in the Cloud: Approaches, Challenges and Solutions," IEEE, Volume. 8, pp. 104893-104917, 2020.

[3] P. Mishra, V. Varadharajan, E. S. Pilli and U. Tupakula, "VMGuard: A VMI-Based Security Architecture for Intrusion Detection in Cloud Environment," IEEE (TCC), Volume. 8, Number. 3, pp. 957-971, 2020.

[4] M. Nadeem, A. Arshad, S. Riaz, S. S. Band and A. Mosavi, "Intercept the Cloud Network From Brute Force and DDoS Attacks via Intrusion Detection and Prevention System," IEEE, Volume. 9, pp. 152300-152309, 2021.

[5] M. Kumar and A. K. Singh, "Distributed Intrusion Detection System using Blockchain and Cloud Computing Infrastructure," IEEE (ICOEI)(48184), 2020, pp. 248-252, doi: 10.1109/ICOEI48184.2020.9142954.

[6] L. Chen, M. Xian, J. Liu and H. Wang, "Intrusion Detection System in Cloud Computing Environment," IEEE (CCNS), 2020, pp. 131-135, doi: 10.1109/CCNS50731.2020.00037.

[7] P. S. Negi, A. Garg and R. Lal, "Intrusion Detection and Prevention using Honeypot Network for Cloud Security," IEEE (Confluence), 2020, pp. 129-132, doi:10.1109/Confluence47617.2020.9057961.

[8] J. Snehi, M. Snehi, A. Bhandari, V. Baggan and R. Ahuja, "Introspecting Intrusion Detection Systems in Dealing with Security Concerns in Cloud Environment," IEEE (SMART), 2021, pp. 345-349, doi: 10.1109/SMART52563.2021.9676258.

[9] O. Alkadi, N. Moustafa, B. Turnbull and K. -K. R. Choo, "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks," IEEE (ITJ), Volume. 8, Number. 12, pp. 9463-9472, 2021.

[10] Xu, R. Zhang, M. Xie and L. Yang, "Network Intrusion Detection System as a Service in OpenStack Cloud," IEEE (ICNC), 2020, pp. 450-455, doi: 10.1109/ICNC47757.2020.9049480.

[11] V. Ravindranath, S. Ramasamy, R. Somula, K. S. Sahoo and A. H. Gandomi, "Swarm Intelligence Based Feature Selection for Intrusion and Detection System in Cloud Infrastructure," IEEE (CEC), 2020, pp. 1-6, doi: 10.1109/CEC48606.2020.9185887.

[12] H. Gajjar and Z. Malek, "A Survey of Intrusion Detection System (IDS) using Openstack Private Cloud," IEEE (WorldS4), 2020, pp. 162-168, doi: 10.1109/WorldS450073.2020.9210313.

[13] P. Cocoros, M. Sobocinski, K. Steiger and J. Coffman, "Evaluating Techniques for Practical Cloud-based Network Intrusion Detection," IEEE, 2020, pp. 62-67, doi: 10.1109/SmartCloud49737.2020.00020.

[14] Gupta and M. Kalra, "Intrusion Detection and Prevention system using Cuckoo search algorithm with ANN in Cloud Computing," IEEE (PDGC), 2020, pp. 66-72, doi: 10.1109/PDGC50313.2020.9315771.

[15] P. Meyer et al., "Demo: A Security Infrastructure for Vehicular Information Using SDN, Intrusion Detection, and a Defense Center in the Cloud," IEEE (VNC), 2020, pp. 1-2, doi: 10.1109/VNC51378.2020.9318351.

[16] P. Mishra, V. Varadharajan, E. S. Pilli and U. Tupakula, "VMGuard: A VMI-Based Security Architecture for Intrusion Detection in Cloud Environment," IEEE (TCC), Volume 8, Number. 3, pp. 957-971,2020.

[17] J. A. Ajala, G. Saini and Pooja, "CLOUD-IOT Based Smart Villa Intrusion Alert System," IEEE (ICRITO), 2020, pp. 1326-1329, doi: 10.1109/ICRITO48877.2020.9198022.

[18] K. Sethi, R. Kumar, N. Prajapati and P. Bera, "Deep Reinforcement Learning based Intrusion Detection System for Cloud Infrastructure," IEEE (COMSNETS), 2020, pp. 1-6, doi: 10.1109/COMSNETS48256.2020.9027452.

[19] M. Hachimi, G. Kaddoum, G. Gagnon and P. Illy, "Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks," IEEE (ISNCC), 2020, pp. 1-5, doi: 10.1109/ISNCC49221.2020.9297290.

[20] M. Nadeem, A. Arshad, S. Riaz, S. S. Band and A. Mosavi, "Intercept the Cloud Network From Brute Force and DDoS Attacks via Intrusion Detection and Prevention System," IEEE, Volume. 9, pp. 152300-152309, 2021.

[21] L. Wang, J. Yang, M. Workman and P. Wan, "Effective algorithms to detect stepping-stone intrusion by removing outliers of packet RTTs," in Tsinghua Science and Technology, vol. 27, no. 2, pp. 432-442, April 2022, doi: 10.26599/TST.2021.9010041.

[22] Z. Shi, S. He, J. Sun, T. Chen, J. Chen and H. Dong, "An Efficient Multi-task Network for Pedestrian Intrusion Detection," in IEEE Transactions on Intelligent Vehicles, doi: 10.1109/TIV.2022.3166911.

[23] L. Nie et al., "Intrusion Detection for Secure Social Internet of Things Based on Collaborative Edge Computing: A Generative Adversarial Network-Based Approach," in IEEE Transactions on Computational Social Systems, vol. 9, no. 1, pp. 134-145, Feb. 2022, doi: 10.1109/TCSS.2021.3063538.

[24] M. Zeeshan et al., "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," in IEEE Access, vol. 10, pp. 2269-2283, 2022, doi: 10.1109/ACCESS.2021.3137201.

[25] O. A. Wahab, "Intrusion Detection in the IoT under Data and Concept Drifts: Online Deep Learning Approach," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2022.3167005