Turkish Journal of Computer and Mathematics Education *Vol.12 No.3(2021), 4284-4297*

Research Article

A Novel Technique for IDS in Distributed Data Environment Using Merkel Based Security Mechanism for Secure User Allocation

D. Priyadarshini[a]&Dr. K. Sarojini [b]

[a] Ph.D. Research Scholar, Department of Computer Science, L.R.G. Government Arts College for Women, Tiruppur-641604, Tamil Nadu, India
[b]Asst. Professor & Head, Department of Computer Science, L.R.G. Government Arts College for Women, Tiruppur-641604, Tamil Nadu, India.

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Multiple corporations and people frequently launching their data in the cloud environment. With the huge growth of data mining and the cloud storage paradigm without checking protection policies and procedures that can pose a great risk to their sector. The data backup in the cloud storage would not only be problematic for the cloud user but also the Cloud Service Provider (CSP). The unencrypted handling of confidential data is likely to make access simpler for unauthorized individuals and also by the CSP. Normal encryption algorithms need more primitive computing, space and costs for storage. It is also of utmost importance to secure cloud data with limited measurement and storage capacity. Till now, different methods and frameworks to maintain a degree of protection that meets the requirements of modern life have been created. Within those systems, Intrusion Detection Systems (IDS) appear to find suspicious actions or events which are vulnerable to a system's proper activity. Today, because of the intermittent rise in network traffic, the IDS face problems for detecting attacks in broad streams of links. In existing the Two-Stage Ensemble Classifier for IDS (TSE-IDS) had been implemented. For detecting trends on big data, the irrelevant data characteristics appear to decrease both the velocity of attack detection and accuracy. The computing resource available for training and testing of the IDS models is also increased. We have put forward a novel strategy in this research paper to the above issues to improve the balance of the server load effectively with protected user allocation to a server, and thereby minimize resource complexity on the cloud data storage device, by integrating the Authentication based User-Allocation with Merkle based Hashing-Tree (AUA-MHT) technique. Through this, the authentication attack and flood attack are detected and restrict unauthorized users. By this proposed model the cloud server verifies, by resolving such attacks, that only approved users are accessing the cloud info. The proposed framework AUA-MHT performs better than the existing model TSE-IDS for parameters such as User Allocation Rate, Intrusion Detection Rate and Space Complexity

Keywords: Intrusion Detection System, Machine Learning, Cloud Storage, Security, User- Allocation.

1. Introduction

There is an emerging model in computing called cloud computing which provides users with unlimited services. On the one side, the absence of useless knowledge renders cloud storage valuable. Cloud platforms should be seen as an effective storage platform. Both hackers and terrorists will misuse the cloud for their purposes. For e.g., it is conceivable that a harmful user is residing in a Virtual-Machine (VM), essentially intrudes in several VMs in the cloud, and uses the VM to disperse ransomware or initiate a Distributed-Denial ofService (DDoS) attack and so on. There would be a lot of network traffic related to occupant activity in the cloud world both external and internal traffic. The "external" traffic refers to the network traffic between customers accessing cloud resources from the network, and the "internal" traffic refers to network traffic between VMs in the cloud (Moustafa et al. 2019).

Data transmission will begin to grow exponentially and be at risk for disruptive threats. Cyber threats harm cloud vendors and data users, although cloud usage still suffers. Intruder avoidance is an essential aspect of protection control for cloud storage. This is an overview of the role of IDS to verify the proper operation of the internet in cloud computing (Moustafa et al. 2017).

Today, technology has contributed to the emergence of Block Chain (BC) based systems in various industries. The BC provides new possibilities by facilitating smart contracts, value transfers, and conflict settlement. The BC system has different uses in multiple fields in various sectors that go beyond digital currencies and financing (Moustafa et al. 2018).

The solutions of IDS and BC have been deployed in the cloud to classify and secure data (Keshk et al.2019). IDS based on the cloud are mainly categorized into host-based and network-based classification. Since a Host-

based IDS (HIDS) operates on a host machine or VM, it can track and investigate network packets of the web browser, including the audit of memory and method of the web browser. When suspicious behavior with an active route or VM is observed, the source IP is marked as a connection to the entire network to deter the attacker from jumping through another host and acquiring a connection to some other VM. A Network-based IDS (NIDS) is installed at the infrastructure layer of the organization, or progressively the cloud system to track the traffic of all networks and applications inside a subnet (Keshk et al.2018).

The main problem for this research is about cloud systems for managing data security and confidence sharing across CSPs. Cloud services are known as distribution, publically available and autonomous this brings up certain confidence problems when there are several components under the charge of various parties. The tendency among Cloud services is protection and data transparency. That is very challenging to calculate the credibility of the vulnerable groups.

The motivation of this research is security as the major concern for providing cloud services to authorized users because of the unreliability of CSP. Cloud computing enables multiple users to access the data stored in the cloud. There is a possibility for unauthorized access to stored cloud data which leads to an increase in the overload with a lack of security in the cloud environment. From that, the load needs to be balanced by allowing the only authorized persons to access the data for enhancing the security level with less overload, communication delay, cost and execution time.

During data storage service in the cloud, the balancing of the load is a major concern due to the lack of authentication. The issues related to load balancing lead to reduce the security of the data stored on a cloud platform. From that, there is a necessity to verify whether the user is authenticated or not and increase the data security for achieving secured load-balanced data storage with minimized space consumption on the cloud. With this intend, some research works are developed on cloud storage with load-balanced algorithms and authorization techniques.

To address the issues related to intrusion (attack) discovery, authentication and load balancing on the cloud, the proposed Authentication based User-Allocation with Merkle based Hashing-Tree (AUA-MHT) technique is employed. In this proposed technique, two important stages such as Identity-based authentication and Merkle Hash Tree construction are carried out to attain secured cloud data storage with less space complexity.

The contributions of this research are:

- In the identity-based authentication stage, the login process is carried out by the user sending the valid session key to the server and then the user ID and password of a specific user are successfully stored on the cloud. If the user wants to store their data on the cloud, the server checks the ID from the user and the ID stored on the cloud which is entered for the first time for login purposes. From that, the user is authenticated or not (i.e., Authentication attack and Flooding attack) addressed.
- In this way, the user is authenticated to store their data which leads to effectively balance the load by only allowing the authorized users for accessing the data stored on the cloud.
- The cloud server enables services to the only authorized user by detecting the attacks with the help of identity-based authentication.
- Followed by, Merkle Hash Tree is generated for storing the data with minimal storage space in a secured manner.
- Through the Merkle Hash Tree, the cloud users store their data in terms of the hash value.
- Thereby, the data is securely extracted from cloud storage which leads to enhance data security on the cloud.

The rest of the paper is organized as follows: Section 2 details the related works of IDS and load balancing in the Cloud Server, Section 3 briefs about the Existing System and Methodologies of the proposed systems, Section 4 provides the Results and Discussion of the proposed research work and Section 5 concludes the article.

2.Related Work

G Loukaset al. (2017) utilized RNN type Deep-Learning improved by LongShort-TermMemory (LSTM) to vastly enhance network security precision for something like a mobile robot. Researchers showed how those methodologies achieved an accurate result with greater flexibility than approaches focused on conventional artificial intelligence.

Aygunet al. (2017)suggested multiple algorithms utilizing Deep-Learning and AE for denoising to identify attacks such as zero-day with higher precision. This research was using a probabilistic methodology to assess the accuracy threshold models.

Shone et al. (2018) introduce a unique auto-encoder model of non-symmetric dimensionality reduction. This approach was coupled with such a method to build a classification model. The whole test obtained useful conclusions for the KDD Cup-99 dataset and the NSL KDD dataset.

Papamartzivanoset al. (2019)have suggested an innovative scheme of combining the advantages of a fragmented and big data IDS with the MAPE-K platform to create a scalable, self-adaptive and autonomous IDS. They integrated the datasets given by KDD Cup 99 and NSL-KDD and generated a unified dataset for review.

Xie et al. (2020) have addressed instances of cloud sharing using blockchain. The authors have spoken about the protection and privacy concerns in cloud storage with the general principles of blockchain technology in contrast.

3.Methodologies

Due to the Cloud computing features such as scalable, flexible, reliable the minimal expense, many users wished to store their sensitive data in the cloud environment. Here, the CSP is fully responsible to access, administrative and outsource the user data on the cloud via the internet. Figure 1 gives the architecture for the storage of data in the cloud.

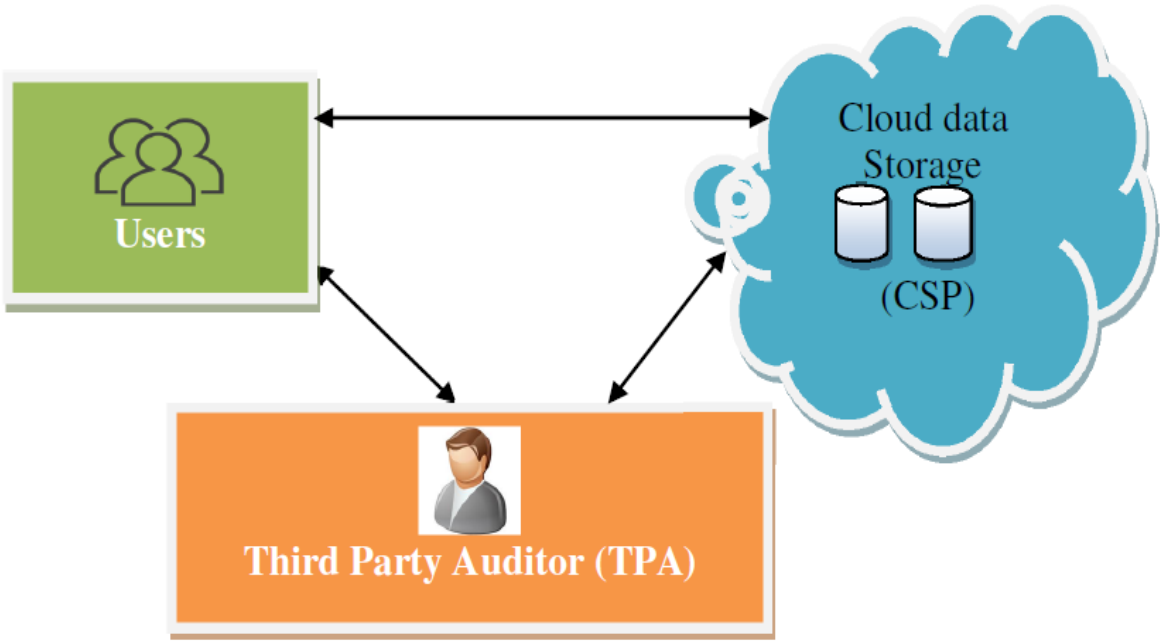

Figure 1: Cloud Architecture for Storing Data

As shown in Figure 1, the system model of cloud data storage contained three entities such as Cloud-User, Third-Party-Auditor (TPA) and Cloud-Service-Provider (CSP). Let us assume, the number of cloud-users who store their sensitive data in the cloud environment. The CSP can store, control and distribute the cloud data in a server. An optional TPA allows trusted users for accessing the data stored on the cloud.

3.1 Existing System

Two Stage based Classifier-Ensemble for Intellectual Anomaly Based IDS (TSE-IDS)

The system consists of a selection of features, classifier model training, and finally, model validating. The key aim of the first-tier is to decide a reasonable set of features for the vulnerability scanning role. The whole methodology requires a mixture of adaptive searching approaches such as Particle Swarm Optimization, Ant Colony Optimization, and Genetic Algorithms (GA). In this way, a two-stage model for classification is planned. The second-tier blends two classifiers, i.e. RotationForest (RF) and Bagging (BG). Thirdly, the developed two stage meta classifier is tested. The training is conducted utilizing 10 fold cross-validation methods.

Disadvantage:

- The rate for detecting the attacks had not improved.
- It failed to minimize the space complexity at the required level.
- The balancing of load on the cloud remained unaddressed in this system.

3.2 Proposed System

Authentication based User-Allocation with Merkle based Hashing-Tree (AUA-MHT)

The proposed AUA-MHT technique is introduced to enhance the load balancing efficiency and minimizes the space complexity on cloud data storage system. Then, the unauthorized users are identified and removed by considering the authentication attack and flooding attack. By addressing these attacks, the cloud server verifies whether the data on cloud-only is accessed by the authorized users. Figure 2 shows the architecture diagram of AUA-MHT.

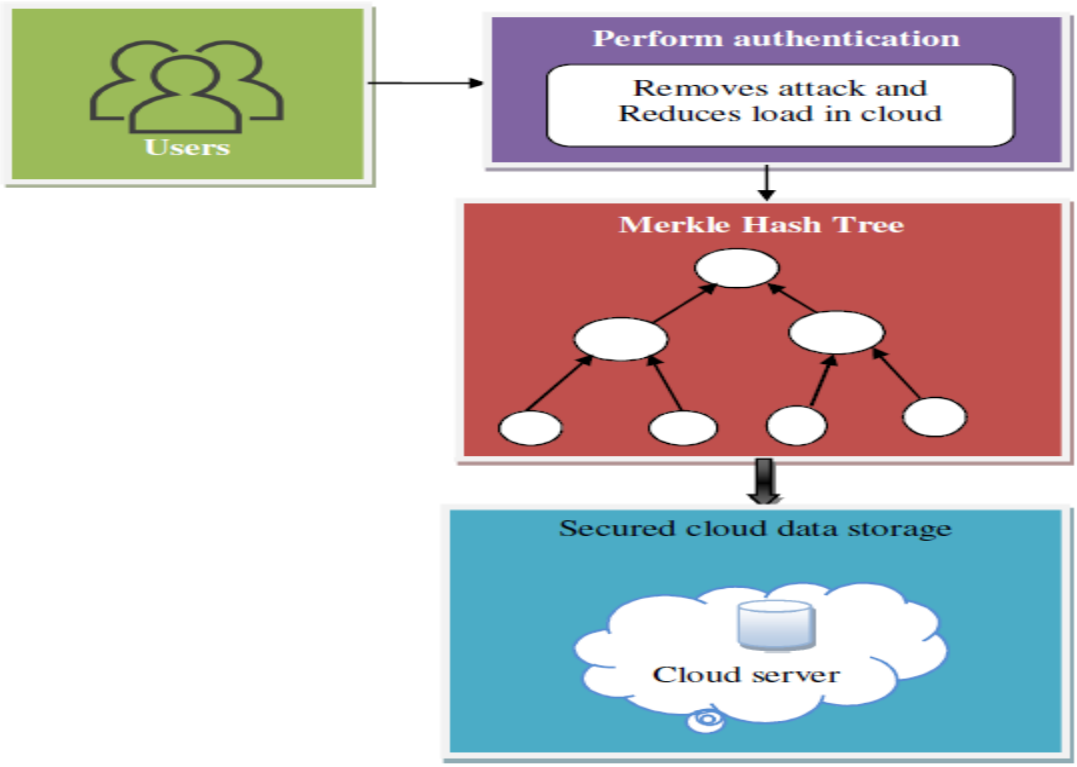

Figure 2: Architecture Diagram of AUA-MHT

As shown in Figure 2, the proposed AUA-MHT technique provides secured cloud data storage with minimized space complexity. Each user attains a unique identity (ID) and stores their user ID on the cloud. When the server receives a request from the user for accessing the data stored on the cloud, the server authenticates that user through the user ID. If the ID from the user is matched with the ID of the user stored in the cloud, then the server enables the services to the user.

Otherwise, the cloud server discards the user request. This addresses the authentication attack in the cloud. In this way, the load is successfully balanced by enabling the services only to the authorized user. Besides, attacks (i.e., unauthorized user) such as authentication attack and flooding attack is effectively reduced for enabling the secured data storage on the cloud. Besides, the authenticated users store their data with minimized space complexity through the Merkle Hash-Tree (MHT) for improving the security level. In the cloud server, the MHT is constructed by providing the hash value of each input data. Hash value ensures the secured storage since the server use this value for storing their data. This ensures the secured cloud data storage in the cloud using the AUA-MHT technique. The process involved in the design of the proposed AUA-MHT technique is clearly explained in the following sections.

3.2.1Load Balancing By Performing Authentication Based User-Allocation

In the implementation of the proposed AUA-MHT technique, the balancing of load on cloud data storage is the first entity. In the proposed, the load balancing is achieved by carrying out the authentication process. For ensuring cloud services, two phases such as registration and authentication phase are required to carry out on the cloud. Before performing the data communication, each user needs to register on Cloud Server (CS) to access the cloud services. Figure 3 shows the process carried out for cloud user registration.

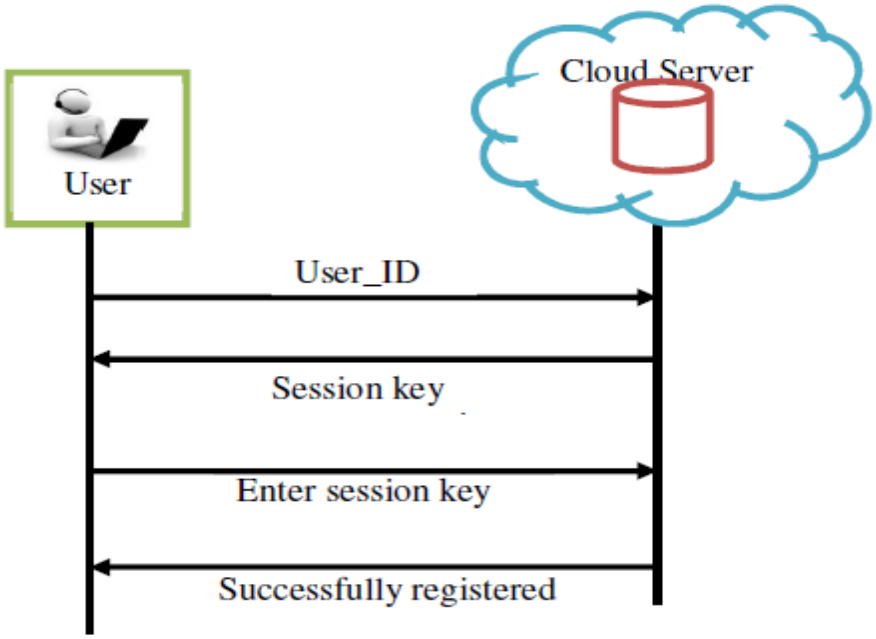

Figure 3: Process of Cloud User Registration

As shown in Figure 3, each user successfully registered on the cloud. At first, the user enters 'User-ID' and password 'PW' as an input to the cloud server in the registration phase. Next, the cloud server creates the Session Key (SK) and sends it back to the registered user ID. Followed by, the user transmits the received session key to the cloud server within an ascertain period. If the user did not enter the session key at in specific time, then the user has to login again. Then, the generation of the valid session key by the cloud server is expressed as given below in Equation 1.

$$CS \rightarrow SK \quad \text{Eq}\rightarrow \quad 1$$

Here, 'CS' and 'SK' denote the Cloud Server and Session Key. The user successfully registered within the cloud server after receiving the valid session key. Then, the user uses the help of 'User-ID' and password 'PW' for login purposes which are expressed as given below in Equation 2.

$$U \rightarrow \{User\_ID \| PW\} \quad \text{Eq}\rightarrow \quad 2$$

The registered users are allowed to attain the different types of services in the cloud environment. Then, the cloud server verifies if the user is authenticated or not when the user stores the data on the cloud. The user verification is done through the Authentication attack and Flooding attack by comparing the cloud user-ID with the ID stored in the cloud server. Then, the authentication process is carried out as described in below Figure 4.

As shown in Figure 4, the cloud user authentication phase is carried out to detect the presence of an authorized and unauthorized user in the cloud environment. When the user wants to store their data on the cloud, then the cloud server verifies the user through the 'User-ID' and password 'PW'. The verification is carried out by comparing the '*User-ID' and password 'PW' from the user to the 'User-ID' and password 'PW' is given by the user at the time of registration which is stored on the cloud server. When the verification is successful, then the user allows storing the data in the cloud. Otherwise, the user is identified as an unauthorized user if the verification is failed.

In cloud computing, there is a possibility for the unauthorized user because of two types of attacks such as authentication attacks and flooding attacks which are considered in the proposed technique. Thus, the proposed technique performs identity-based authentication for detecting whether the user is authenticated or not before providing the services to users. Thereby, the authentication attack is addressed.

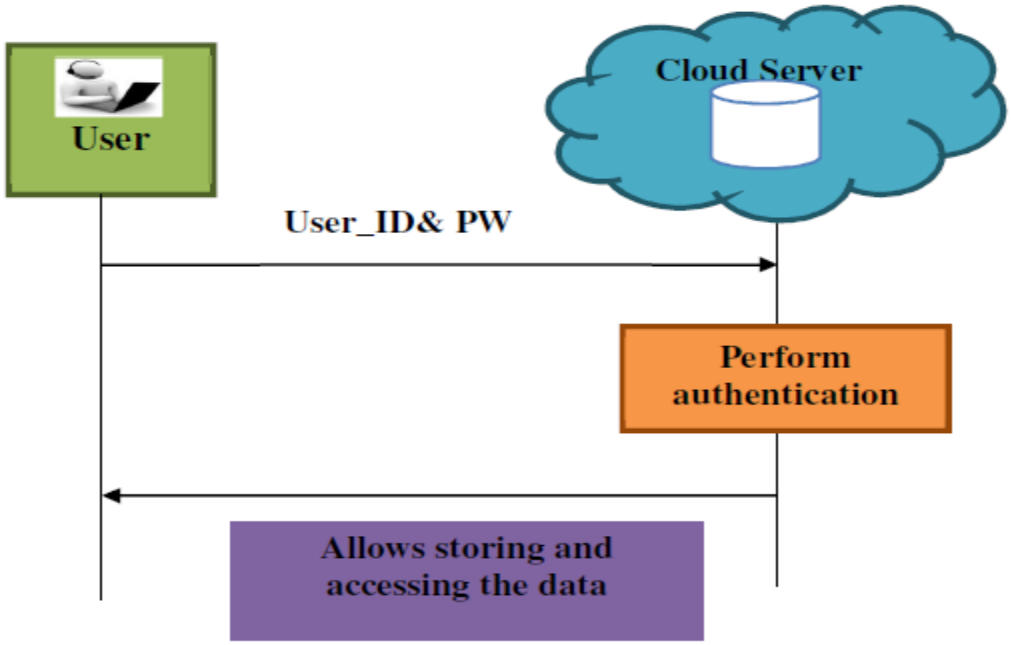

Figure 4: Cloud User Authentication Phase

If the attack is a flooding attack, there is a chance to generate fake data and false requests to the cloud server. After getting the false request from users, the cloud server performs the authentication on the requested users to verify whether the user is authenticated or not.

From that, the proposed technique removes these attacks by performing identity-based authentication. Through authentication, the authenticated users are only allowed to store the data on the cloud server. Then, the unauthorized users (i.e. Authentication attack and Flooding attack) are removed. In this way, the loads get reduced with leads to improve load balancing efficiency. Pseudocode 1 describes the identity-based load balancing in the cloud environment.

Pseudocode 1: Identity Based Load Balancing Algorithm

Input: Number of cloud users $U = U_1, U_2, \ldots . U_n$

Output: Improve the load balancing efficiency and attack detection rate

Step 1: **Begin**

Step 2: **For** each user

Step 3: Send user ID for registration

Step 4: Perform authentication at server side

Step 5: **If** user ID is matched with stored ID then

Step 6: user is said to be an authorized

Step 7: CSP allows the services

Step 8: **Else**

Step 9: user is said to an unauthorized

Step 10: CSP denied the services

Step 11: **End if**

Step 12: **End for**

Step 13: **End**

As shown in Pseudocode 1, the identity-based load balancing algorithm is executed to identify and remove unauthorized users. At first, the registration phase is carried out to login the user into the cloud by using the 'User-ID' and password 'PW'. The identity-based authentication is performed in the cloud server if the user wants to store their data on the cloud.

From that, the user is verified by comparing the cloud user-ID with the ID stored in the cloud server. When two IDs are matched, then the cloud server provides services to the user. Otherwise, the cloud server does not allow the user to access the data stored on the cloud. In this way, the unauthorized users are identified to reduce the loads on the cloud. After performing the identity-based authentication, the authorized users are allowed to access their data with improved load balancing efficiency. Therefore, the proposed technique improves the load-balancing efficiency by removing the attacks while storing the data in the cloud.

3.2.2 Merkle Based Hashing Tree For Secure Cloud Data Storage

The proposed AUA-MHT technique then introduces MHT intending to enhance the security on the cloud data storage with minimized space complexity. The MHT comprises several leaf nodes with the hash value of data blocks in a set of data. By storing the data into a hash value, the security of the data stored on the cloud gets improved. Figure 5 shows the processing diagram of the Merkle hashing technique.

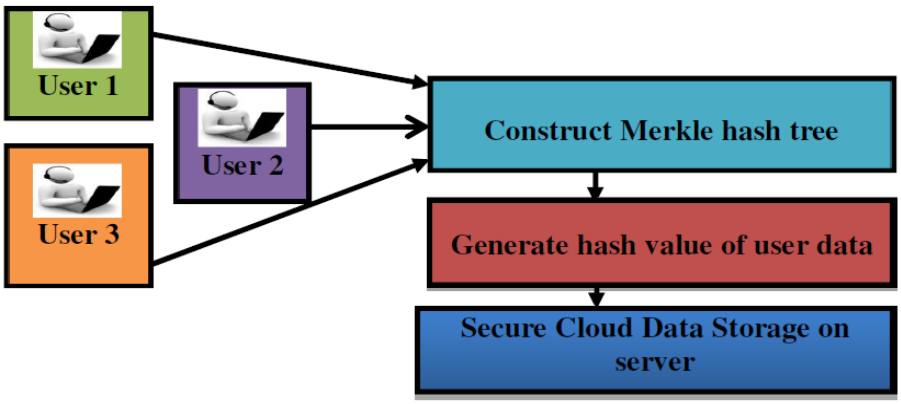

Figure 5: Processing Diagram of MHT Based Secure Cloud Data Storage

As shown in Figure 5, the construction of MHT is helped to securely store the data with less space complexity. At first, the user sends requests to the cloud server for storing the data. Followed by, the MHT is formed at the cloud server by generating the hash value of each data. The generation of hash value leads to attaining the secured storage because the cloud server uses the hash value instead of the data files. Then, the user requested data is successfully extracted and retrieved by the cloud user through the MHT. With the formation of MHT, data integrity and dynamic maintenance are achieved in the cloud environment. Figure 6 shows the MHT structure.

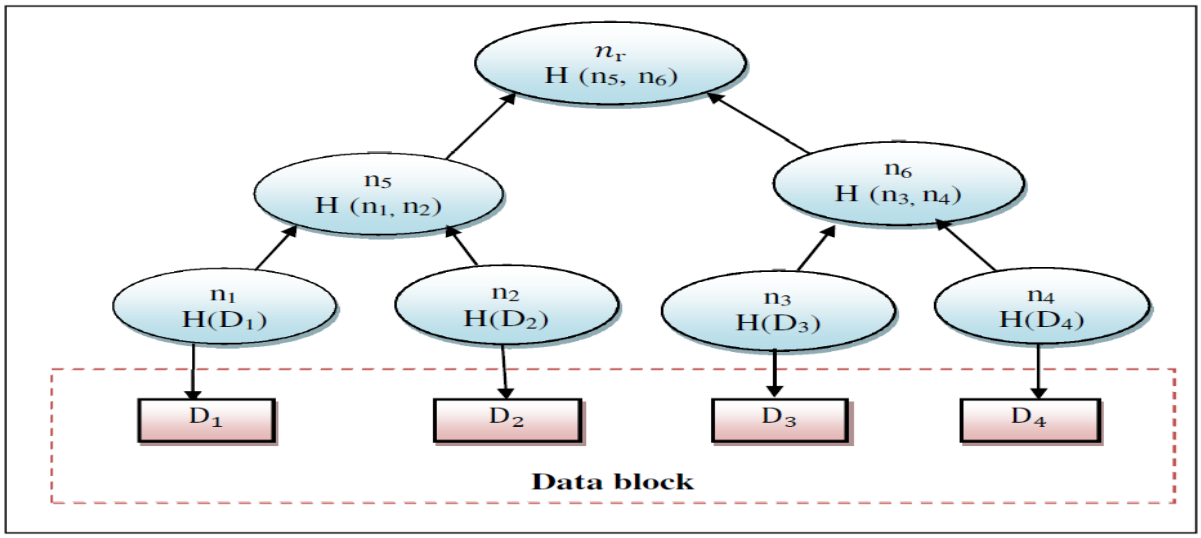

Figure 6: MHT Structure

As shown in Figure 6, the MHT is constructed with a root node and leaf nodes and data block. In the tree construction, the non-leaf nodes (Root node (R), n5, n6) are labeled with the hash value of its children nodes. Then, the leaf nodes (n1, n2, n3, n4) are labeled with the hash value of a data block (D1, D2, D3, D4) (i.e. original file). In the tree, the secured hash function is denoted as 'H'. The MHT structure is contained with the 'n' number of leaf nodes which is mathematically expressed as below in Equation 3.

$$MHT = \{n_i | n_i = H(D_i), \quad 1 \leq i \leq n$$

Eq→ 3

From the above (3.1), the MHT is represented as 'BCD'. Here, the number of nodes in the tree is represented as 'n_i'. Then, 'H(Di)' is signified as a secure hash function of data Dithat contains various data stored in the tree. Thus, the value of the non-leaf node is generated through the concatenation of two leaf nodes is expressed as following Equation 4.

$$n_i = H(n_i^L \| n_i^R)$$

Eq→ 4

From the above Equation 4, the left and right child nodes are denoted as 'n_i^L' and 'n_i^R' respectively. Then, the root node of the Merkle hash tree is signified as 'n_r'. The smallest ordered node-set " $\Omega_i=\{n_1^i, >>n_2^i>>...\}$" is utilized by '$n_i$' for estimating the root node which is known as Auxiliary Authentication Information (AAI). As shown in Figure 6, the leaf nodes n1, n2, n3, n4 holds the hash of data 'H(D1), H(D2), H(D3), H(D4)' respectively. The non-leaf nodes 'n_5'is expressed as follows in Equation 5.

$$n_5 = H(n_1, n_2) = H(H(D_1) \| H(D_2))$$

Eq→ 5

From the above Equation 5, '||'is signified as a concatenation symbol. Similarly, the non-leaf nodes 'n_6' is expressed as in below Equation 6.

$$n_6 = H(n_3, n_4) = H(H(D_3) \| H(D_4))$$

Eq→ 6

Then, 'n_5' and 'n_6' are combined iteratively and rehash the resulting hash value to attain the root node 'n_r' which is expressed as in below Equation 7.

$$n_r = H(n_5, n_6) = H(H(D_1) \| H(D_2) \| H(D_3) \| H(D_4))$$

Eq→ 7

The MHT is generated by performing dynamic operations such as insertion and deletion. When the user wants to insert the data on the cloud server, the user needs to create and sends the signature of data to the cloud server with the help of a secret key. After receiving the requests from the user, the server carries out the update operation. Then, the cloud server stores data '(D)' in the block and the leaf node has the hash value of data 'H(H(D))'. The new leaf node is added to the original tree and the index of the particular root node is increased by '1'. In general, the index of a node is two which holds two child nodes.

During the insert and delete operation, the index is either decreased or increased. If the user wants to insert the data from the tree, the information of all the nodes which lie in the path from this leaf node to the root node is modified. Their hash values are recalculated and then the index of a leaf node is increased by 1.Therefore, the new root path is generated in the tree. Figure 7 shows the data insertion operation.

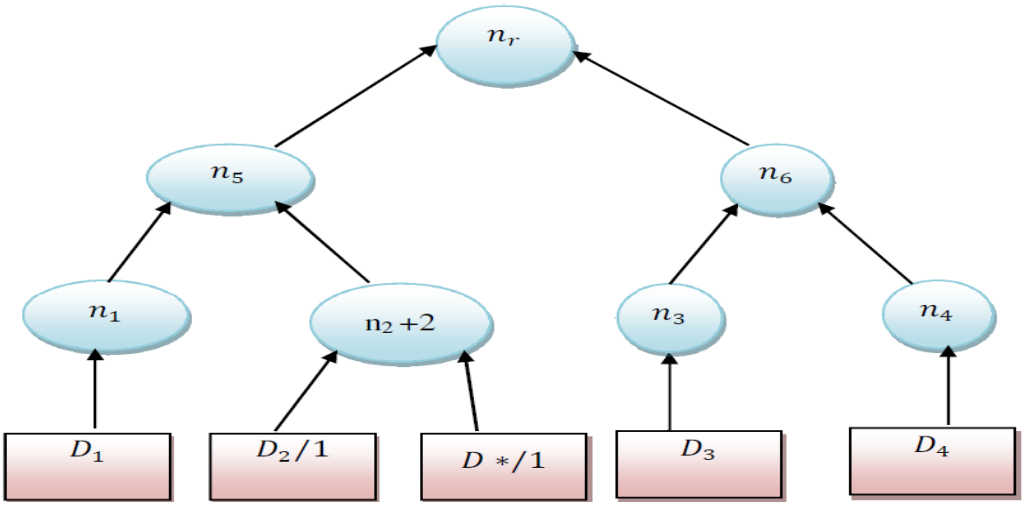

Figure 7: Data Insertion Operation

As shown in Figure 7, the data insertion process is performed to insert the data into a tree. In the above Figure, the new data (D*)is inserted into an MHT. When the user requires to removes the data from the tree, the information of all the nodes which lie in the path from the leaf node to be removed and to the root are modified. Their hash values are recalculated. The index of a leaf node is decreased by -1. Figure 8 shows the data deletion operations.

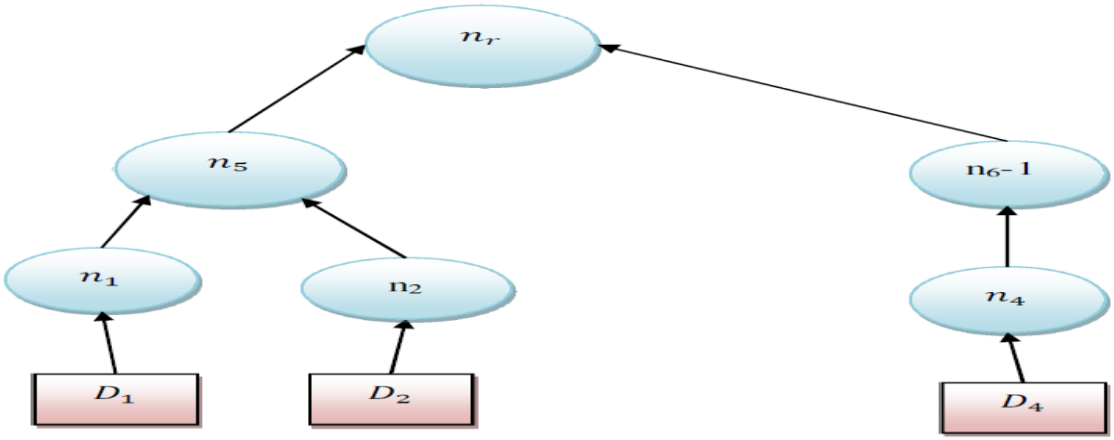

Figure 8: Data Deletion Operation

As shown in Figure 8, the data deletion operation is carried out to remove the unwanted leaf node from the tree. In the above Figure, the data 'D_3' is removed and their leaf node (n_3) is also removed. By the performance of data insertion and deletion operation, the storage space gets reduced on the cloud server. Pseudocode 2 shows the process of MHT for secure data storage is described as follows.

As shown in Pseudocode 2, the construction of MHT helps to enhance the secured data storage with minimized space complexity in the cloud environment. At first, the tree is constructed with the number of nodes and the user input data. For each cloud user data, the hash value is estimated to provide security on the cloud.

Secondly, the insertion and deletion operations are carried out for adding the data into the tree and removing the data from the tree. In the insertion operation, the service provider allocates the new leaf node for the authorized user data. After that, the hash value of data is computed and the index of the particular node gets increased.

Then, the hash value of the root node is recalculated. Similarly, during the deletion operation, the particular leaf node (unwanted leaf node) which holds a hash value of that specific data from the tree is deleted. After that, the hash value of data is computed and the index of the particular node gets decreased. Then, the hash value of the data is recalculated. From that, the proposed AUA-MHT technique enhances the secured data storage with minimum space complexity.

Pseudocode 2:Merkle Based Hashing Tree

// Merkle Hash Tree Algorithm

Input: Nodes $n_i = n_1, n_2, n_3, n_4, \ldots n_n$, user data $D_i = D_1, D_2, \ldots D_n$

Output: Secure cloud data storage with minimum space complexity

Step 1:Begin

Step 2: Construct the Merkle hash tree with number of nodes and data block using Equation(3.1) and (3.2)

Step 3: Calculate the hash value of each data

Data Insertion Operation

Step 4:If user add the data in tree then

Step 5:CSP assign a new leaf node and then add the data

Step 6: Index of particular root node is increased by '1'

Step 7:Recalculate the hash value of root node

Step 8:End if

Data Deletion Operation

Step 9:Ifuser delete the data in tree then

Step 10:CSP removes unwanted leaf node from the tree

Step 11:Index of particular root node is decreased by '1'

Step 12: Recalculate the hash value of root node

Step 13:End if

Step 14: End

4.Results and Discussions

To analyze the performance of the proposed AUA-MHT technique which is compared with the existing methods such as TSE-IDS. The number of cloud users is considered from the range of 1000 to 5000 which is taken as input while conducting the experiments. During the experiment conduction, the proposed AUA-MHT technique is compared with the existing method TSE-IDS. The tables and graphs are generated depends on the attained performance values to assure the effectiveness of the proposed technique.

4.1 Performance Analysis of User Allocation Rate (UAR)

The UAR efficiency is measured as the ratio of the number of the authorized user is identified through the authentication to the total number of cloud users in the cloud environment. The UAR/Load Balancing Efficiency (LBE) is mathematically expressed in Equation 8.

$$LBE = \frac{Numberofauthorizedusers\ are\ identified}{Total\ number\ of\ users\ in\ cloud} * 100 \quad \text{Eq→ 8}$$

From Equation 8 the load across the multiple servers is balanced by the identification of the authorized user to ensure the cloud services. The LBE is measured in terms of percentage (%). The higher value of load balancing efficiency ensures the better performance of the technique.

Table 1: Numerical Comparison Of LBE

CLOUD USERS	TSE-IDS	AUA-MHT
1000	82	95
2000	73	89
3000	66	84
4000	60	79
5000	54	76

The above Table 1 shows the experimental results of the LBE depends on the number of cloud users. The performance of LBE gradually changes in the above two methods with the respect to the number of cloud users in the cloud environment. As shown in Table 1, the proposed technique effectively improves the load-balancing efficiency than the existing method. According to the table value, the graph is plotted in Figure 9.

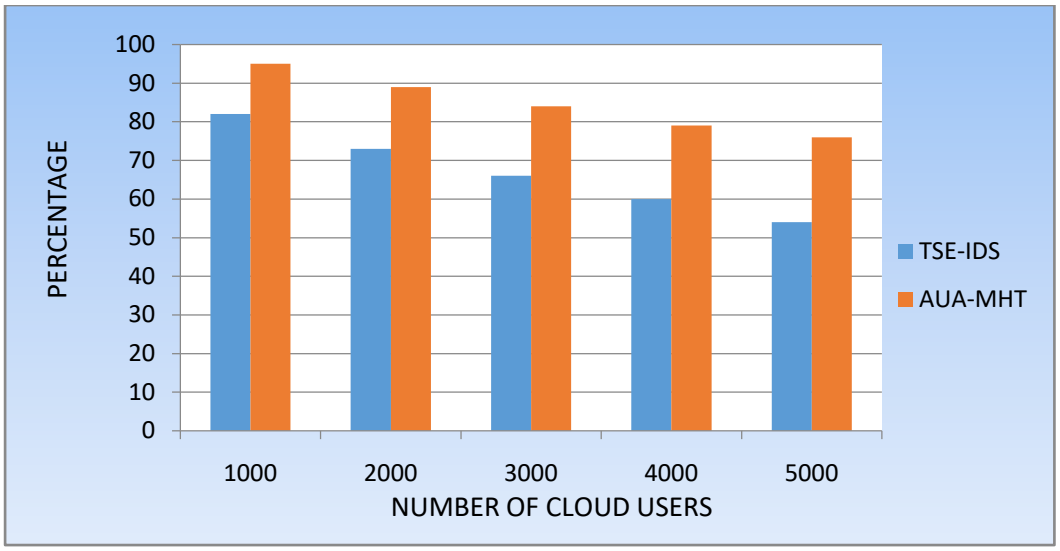

Figure 9: Graphical Comparison Of LBE

4.2 Performance Analysis of Intrusion Detection Rate (IDR)

The IDR is defined as the ratio of the number of intrusions (i.e. unauthorized users) is correctly identified to the total number of users in the cloud environment. The intrusion detection rate is mathematically expressed as given in below Equation 9.

$$IDR = \frac{No.ofusers - No.ofintrusionscorrectlydetected}{No.ofusers} * 100 \quad \text{Eq→ 9}$$

From the above Equation 9, the IDR is measured in terms of percentage (%). The higher value of IDR ensures the better performance of the technique.

Table 2: Numerical Comparison of IDR

CLOUD USERS	TSE-IDS	AUA-MHT
1000	86	97
2000	75	93
3000	69	91
4000	62	87
5000	58	82

The above Table 2 shows the experimental results of the IDR depends on the number of cloud users. The performance of IDR gradually changes in the above two methods with the respect to the number of cloud users in

the cloud environment. As shown in Table 2, the proposed technique effectively improves the intrusion detection rate than the existing method. According to the table value, the graph is plotted in Figure 10.

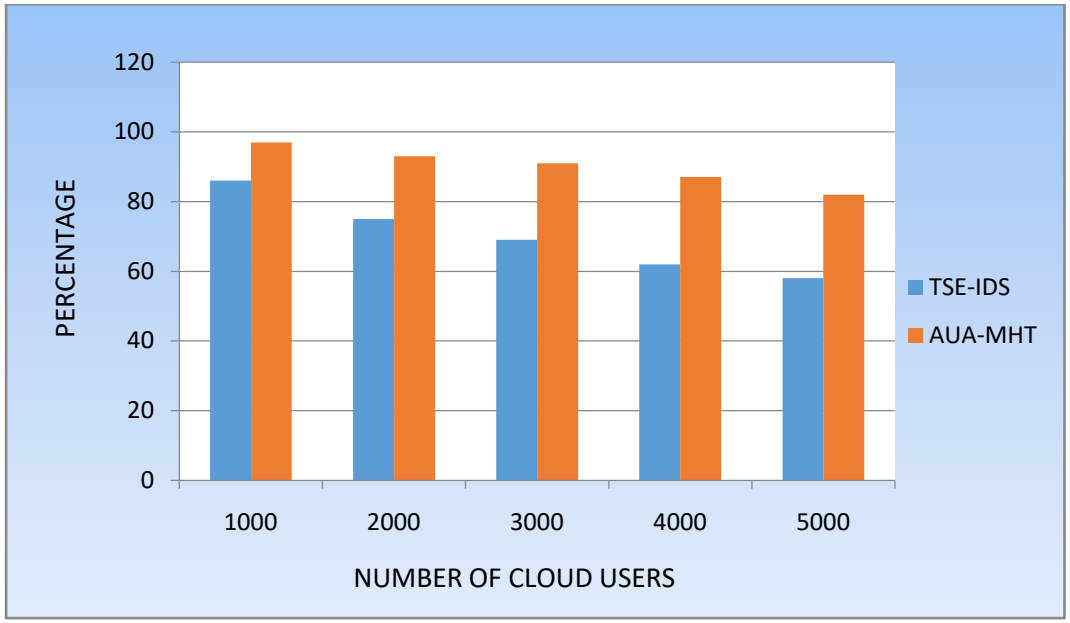

Figure 10: Graphical Comparison of IDR

4.3 Performance Analysis of Space Complexity

The space complexity is evaluated as the amount of memory space is consumed to store the number of user data on the cloud in a secure manner. The space complexity is mathematically expressed as given in below Equation 10.

$$Spacecomplexity = Number\ of\ user\ data\ 'D_i' * memory\ (storingthesingledata)$$

Eq→ 10

From the above Equation 10, 'Di' is represented as the user data. The space complexity is measured in terms of megabytes (MB). The lower value of space complexity ensures better performance of the technique.

Below Table 3 shows the experimental results of the space complexity based on the different number of user data. The number of user data is considered from the range of 100 to 500 which is taken as input while conducting the experiments.

Table 3: Numerical Comparison of Space Complexity

NUMBER OF USER DATA	TSE-IDS	AUA-MHT
100	436	320
200	789	490
300	1143	670
400	1390	890
500	1697	1135

The above Table 3 shows the experimental results of the Space complexity depends on the number of user data. The performance of Space complexity gradually changes in the above two methods with the respect to the number of user data in the cloud environment. As shown in Table 3, the proposed technique effectively minimizes the memory space than the existing method. According to the table value, the graph is plotted in Figure 11.

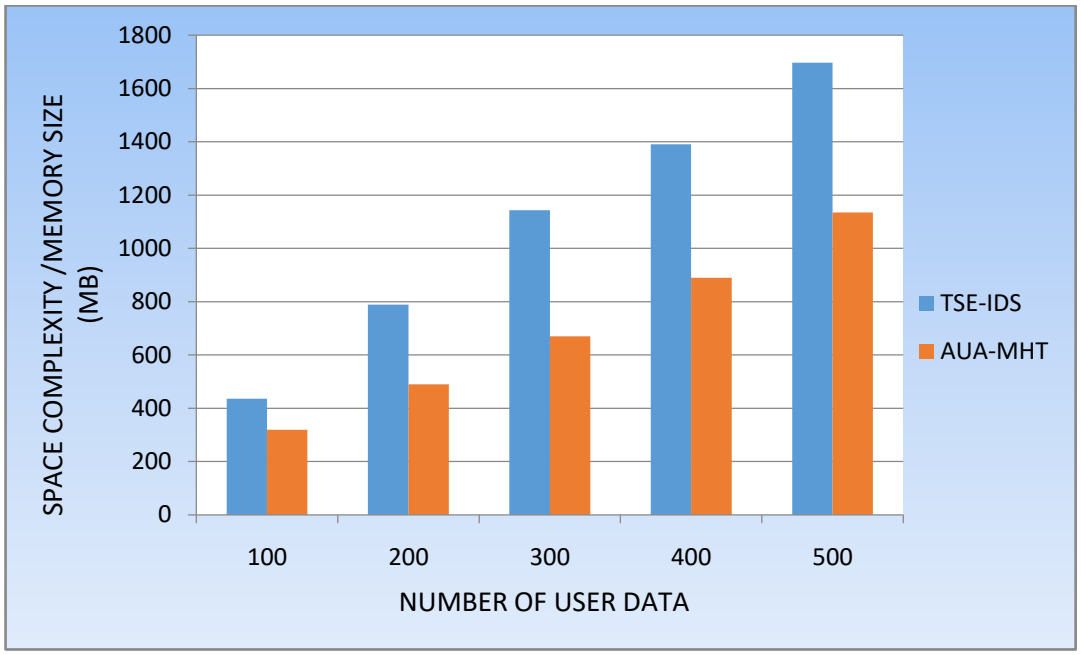

Figure 11: Graphical Comparison of Space Complexity

5.Conclusion

The elevated degree of security vulnerabilities has had a substantial effect on consumers and programmers of cloud services. The design of cloud infrastructure demands new techniques to defend it from attacks and protection risks. Also, conventional network protection technologies and tools may benefit from improvements and modifications. Therefore, IDS as the most basic aspect of networking must be tailored to the context. Distributed IDS is used in cloud environments because of the complex existence and heavy traffic level of the cloud. Even so, such allocation raises the workload of the IDS framework and needs a process to manage the workload of IDS computing, to avoid overload circumstances in certain servers when others are underflowed. The resources can be handled more efficiently because of this. An overload on network congestion can lead to negative effects on the functioning of websites. Using multiple servers that don't usually get enough computing capability, throughput and storage, it may be done in a cloud system. This research paper proposed a novel strategy that was implemented to the above issues to improve the balance of the server load effectively with protected user allocation to a server, and thereby minimize resource complexity on the cloud data storage device, by integrating the Authentication based User-Allocation with Merkle based Hashing-Tree (AUA-MHT) technique. Through this, the authentication attack and flood attack are detected and restrict unauthorized users. Also, the existing condition of server services accessible in different locations is regarded by judging by processors, connectivity and storage. For such a method, including given the context of connections in the same session, load balancing won't negatively impact the alert identification of the IDS. The simulation findings suggest that AUA-MHT is more capable of testing load balancer and IDS technology than the TSE-IDS. In the future, this research will be extended to some advanced ensemble approaches in Machine Learning

References

N. Moustafa, B. Turnbull and K.-K.-R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things", IEEE Internet Things J., vol. 6, no. 3, pp. 4815-4830, Jun. 2019.

N. Moustafa, G. Creech, E. Sitnikova and M. Keshk, "Collaborative anomaly detection framework for handling big data of cloud computing", Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), pp. 1-6, Nov. 2017.

N. Moustafa, G. Creech and J. Slay, "Anomaly detection system using beta mixture models and outlier detection", Progress in Computing Analytics and Networking, pp. 125-135, 2018.

M. Keshk, E. Sitnikova, N. Moustafa, J. Hu and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems", IEEE Trans. Sustain. Comput., 2019.

M. Keshk, N. Moustafa, E. Sitnikova and B. Turnbull, "Privacy-preserving big data analytics for cyber-physical systems", Wireless Netw., vol. 24, pp. 1-9, Dec. 2018.

G Loukas, T Vuong, R Heartfield, et al. "Cloud-based cyber-physical intrusion detection for vehicles using Deep Learning, "IEEEAccess, 2017, 6(1):3491-3508.

R C Aygun, A G Yavuz. Network Anomaly Detection with Stochastically Improved Autoencoder Based Models[C]//2017IEEE 4th International Conference on Cyber Security and CloudComputing(CSCloud). IEEE Computer Society, 2017.

N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection, "IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, 2018, doi: 10.1109/tetci.2017.2772792.

Papamartzivanos D, Marmol F G, Kambourakis. G. Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems [J].IEEEAccess, 2019:1-1.

S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai and M. Imran, "Blockchain for cloud exchange: A survey", Comput. Electr. Eng., vol. 81, Jan. 2020