International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Efficient Water Marking and Intrusion Detection System Framework for Insider Attack Detection in Cloud Based E-Healthcare Data Management

[1]S. Palani ,[2*]K. Rameshbabu

Submitted:24/04/2023 **Revised:**23/06/2023 **Accepted:**07/07/2023

Abstract: The health care industries in recent years have been facing numerous challenges towards the implementation of electronic health records in preserving the privacy of patients and keeping intact their information. The impact of e-Healthcare system has resulted in inappropriate investigation of patient's health records. This in turn has affected the cost effectiveness and data consumption. Therefore, the liability towards these two factors is greatly affected. Hence, lack of an appropriate system for detection brings about data breaches. The concept of enabling eHealth systems on the cloud environment will provide a practical solution to the existing problem failing which health records are prone to attacks and intrusion. Any such attacks consequently incur misrepresentation of data which may prove to be endangering for the patients. While the recent researches have failed to throw light on identifying a malignant insider, the current research proposes a novel method which is inbuilt of options for identifying a destructive insider.The effective resource allocation module and watermarking methods used in conjunction with Intrusion Detection System (IDS) for Cloud-based Healthcare System are the main topics of this article. Regarding data storage and malware attack detection, the suggested solution is thought to be more effective and functional. This work uses the Improved Cuckoo Search Algorithm (ICSA) and Convolutional Neural Network for task scheduling and resource allocation (CNN). The spatial domain is reliant on the watermark's hindrance on the pictures' least significant bits (LSB) when a later watermarking approach is utilised. Finally, an intrusion assault detection framework is used for identification in order to determine the attackers of information notification in the e-Health care system. In order to defend against data modification attacks and to identify intruders, watermarking is crucial. Thus, this investigational assessment shows the proposed research which delivers efficient storage of health data in a highly secured environment.

Keywords*: Cloud Computing, Resource allocation, Task Scheduling, Improved Cuckoo Search Algorithm (ICSA), Deep Reinforcement Learning (DRL),Watermarking Techniques, Intrusion Detection System (IDS), E-Health Care Services.*

1. Introduction

One of the most fascinating themes of the IT world, in the recent times is Cloud computing. Cloud computing has always proved to be a source of greater dependability and huge flexibility with low cost. This has been the prime reason why it has created an evolution in the field of computing and has attracted great business men and stakeholders [1]. In the recent times, Cloud computing has developed in leaps and bounds in Information Technology. Hence there is a dire need to store huge data of individuals as well as industries, in the cloud there is always an eternal demand for methods that assure safety and security of information. Stalwart companies like Microsoft in software industry have come together to improve the features of Cloud services [2]. The reason behind Cloud computing gaining a lot of importance over a period of time is due to the fact that it is capable of storing a large amount of data in remote services. This has been the sole reason why Clod computing is considered to be sought after technology. It has drawn the attention of a wide spectrum of people from media, business and research at present as it has a colossal advantage of financial and functional features over other methods.

It is understood that Cloud computing has marked the beginning of a new era of computing and that it has set an example of a resource that is readily available on par with the household services [3]. The feature of Cloud computing are diverse as it provides a great asset to the large, medium and small scale enterprises. Cloud computing's inexpensive software, hardware, and setup is one of its main advantages. This proves to be beneficial for running and easing the businesses. Since the advent of Cloud computing there seems to be a huge expansion of

[1]*Assistant Professor, Sri krishna college of Engineering, Coimbatore.*
E-mail: s.palani87@gmail.com
[2*]*Professor, Vellore Institute of Technology, vellore.*
E-mail: krameshbabu@vit.ac.in

facilities like hosting and delivery through internet, which is believed to have greater outlook in providing lucrative services [4]. Cloud computing is a smart way to entrepreneurs such that it saves lot of time in planning and furnishing of the resources. It goes beyond this and also upsurges the resources when there is requirement.

Though the pros of Cloud computing are very many, security seems to be one of the prevalent apprehensions which deters its progress. Apart from security issues, Cloud computing continues to experience hitches like data privacy and data protection in the market [5]. It is very essential on the part of the users to be aware of the danger caused due to data breaches in the cloud environment. Right now, healthcare is facing these types of threats in the field of business. The idea behind this is to provide an integrated storage system where institutions, healthcare workers, and patients may work together [6]. Widely used database infrastructures and software services implemented in the cloud are crucial tools insofar as the health area is concerned. There are both technical and non-technical issues with this particular function [7-8]. Most of the times, the attacks from inside the organization tend to be more malignant than the external one. It is observed that such threats generally emerge from the employees, suppliers or other companies legally connected to a computer [9]. It should also be noted that these insiders have expertise over the internal functioning of the organization and anytime they can acquire control of the rights and privileges in order to initiate an attack. Subsequently, the attack they initiate seemingly appears to take place in a very normal way. In order to protect companies from such inside attacks, they tend to spend more money in the recent times. Sometimes, all efforts may go futile, if there is no prior knowledge about application of protection in the Cloud [10].

However, the insider who carried out the operation was not discovered. Watermarking and encrypting might detect malicious insider changes. Figure 1 shows an illustration of this model.

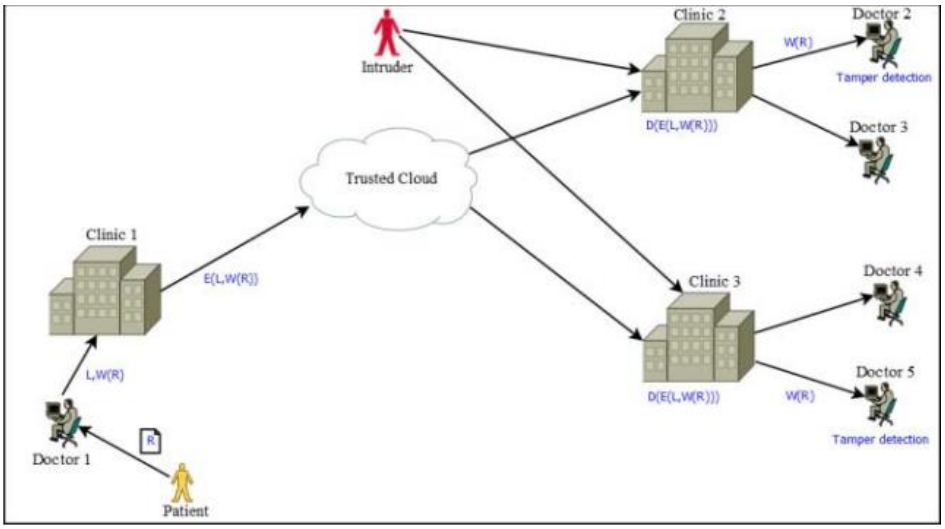

Fig 1: Cloud-based e-Healthcare Insider Attack Detection Framework

Finding insider activity while data is being transferred from the cloud is the main goal of this investigation. During which the data of the healthcare organization is at high risk due to the insider attacks. As soon as these attacks are initiated they launder all the data as an act of vendetta or profit. Hence a situation arises such that there is a dire need to develop a competent method to detect and destroy such attacks. The major apprehension of e-Healthcare systems is to see to that there is no damage of privacy caused due to inadvertent circumstances. At this juncture, what is more important is to invent a system that does not create false alarms [11]. Since, false security alarms may prove to be huge loss with respect to the availability, which in turn may affect the timely accessibility of data during situations of emergency. Unavailability of system can be a huge handicap for an organization owing to great financial and reputational losses.

In order to avoid such crisis, the current study focusses on a methodology for identifying all insider attacks responsible for hosting attacks between organizations and their interface with cloud services. It is also advised to construct a methodology for scheduling tasks and allocating resources depending on Deep Reinforcement Learning (DRL) and the Improved Cuckoo Search Algorithm (ICSA). This project includes a cutting-edge technique for detecting insider attacks on cloud-based healthcare systems. Further, this method also proposes a system which offers thorough knowledge about the existing condition of the system.

Organizing additional research is done as follows: The benefits and drawbacks of attack detection approaches in

cloud environments are highlighted in section 2. For e-healthcare applications, a recommended resource allocation and security paradigm is described in section 3. Discussion of the experimental findings is in section 4. Section 5 concludes the research project by presenting its findings.

2. Literature Review

During the recent times, many a methods have been suggested for developing health monitoring systems. A few techniques are presented in this section.

Gunasekhar et al [12] suggested a method by implementing multi cloud service providers. This method assures security, integrity, access control and confidentiality of sensitive data. The company stores sensitive data in a trustworthy cloud and encrypts it adhering to their security policies and procedures. A separate cloud area is identified to encrypt and store keys which are already encrypted during the encryption process. It is ensured that the organization holds only of encrypted data. Further, it is also verified that even the Administrator of the organization is not aware of the whereabouts of data in the cloud. And if at any point of time the Administrator accesses the data, it is trapped. Therefore, only limited access is given to the users for accessing the data. This also safeguards the cloud from insider attacks.

Yusop et al [13] developeda model to eradicate malignant insider attacks in Cloud computing employing various mitigation approaches and techniques. The Cloud system is more prone to insider attacks and so the chances of invasion are at higher risk. The Cloud service provider can comprehensively reach out from Cloud user to Private and Public Clouds. Hence, the proposed mitigation strategy is considered to provide class-apart security on par with other security tools. It delivers seamlessly an efficient method based on the simulation results.

Duncan et al [14] suggested a model which is in particular employed as attack trees. These trees make it possible to identify insider attack possibilities more quickly. This model, which combines attack-tree and kill-chain-based techniques, is frequently employed in indirect detection techniques. Based on the overlap of the attack tree above a kill chain, the provider determines an attack's potency. As a result, opportunities for indirect detection higher up the tree are accelerated. He makes this suggestion by identifying a specific insider attack. Through a network tap, this attack aims to intercept virtual machines that are in motion within an already-existing cloud cluster. Although the suggested approach is thought to be generally applicable, it applies to all cloud prototypes.

A fuzzy logic based defense mechanism is presented by Iyengar et al [15]. In this model, predefined rules are framed to identify the malware packets. This enables preventive steps to alleviate the DDoS attack. In addition to this model, a descriptive study comprising of various kinds of DDoS attacks and current techniques in are accomplished.

A novel cooperative intrusion detection system (IDS) proposed by Lo et al [16] holds a set-up which could weaken the power of attacks. IDSs in the cloud computing context exchange alarms with one another in order to get this power. To calculate and choose whether to receive the signals transmitted from IDSs or not, a cooperative agent is used. To prevent occurrences of similar types, these cooperating agents are included into every IDS. The installation results show that the suggested system can withstand DoS attacks. The suggested cooperative IDS solution protects against single point of failure attacks without increasing computing effort compared to pure Snort-based IDS.

Dastjerdi et al [17] proposeda model which could deliver invasion detection for cloud applications irrespective of their locations. This model is considered to possess a line of defense with the implementation of mobile technology. This feature of cloud makes it more secure for the chief information officers (CIOs) to widen their set-up by improving on their ability.

Chiba et al. [18 suggested a CH-NIDS (compliance and hybrid NIDS) to monitor networks for intrusions. This technology can only be used on the Cloud, where traffic and service quality are constantly monitored. Snort uses a Back-Propagation Neural network (BPN) to detect known threats based on their signatures and network anomalies. When used in a way that exclusively detects previously unknown threats, BPN classifiers drastically shorten the time it takes to make a discovery. An optimisation algorithm is used to improve the settings. Furthermore, it verifies that the detection rate is at its highest possible accuracy level, with few false positives and few false negatives. Overall, efficiency in terms of money spent is also taken care of. With the help of the signals stored in the central log, IDSs are able to combat DoS and DDoS attacks. Since this strategy largely reduces computing costs, it is regarded to be very cost effective. By using this technique, other IDS will be better able to detect attacks in the Cloud, even if they haven't seen them before.

A flexible Intrusion Detection System (IDS) is presented by Roschke et al [19] where a novel proposal of management architecture of event gatherer component is vividly explained. The familiar IDS and a plug-in idea is employed such that it affirms the expandability and congruity. Research is implemented to perform the flexibility and virtualization-compatibility of this IDS management architecture.

An innovative security model known as the hybrid-network intrusion detection system was presented by Modi et al [20] (H-NIDS). To operate, this model combines a number of classifiers, including Bayesian, Associative, Decision Tree, and Snort. This model's goal is to detect network assaults while managing network traffic.It assesses the competency and efficacy in detection of H-NIDS to confirm its possibility in the Cloud. Further, it also confirms the overall performance and excellence in service. This model possesses greater success rate compared to other models.

Network intrusion detection systems (NIDSs), according to Oktay et al. [21], are made up of different combinations of network designs. Although the majority of providers choose virtualization, a virtualized environment is taken into account. This was chosen to make transitioning from other cloud technologies, like grid and blade designs, easier for application developers. Proxy NIDS architecture, a gateway-based technique, is used in this particular model. The decision to adopt a proxy NIDS model was made because an external entity performs the intrusion detection work. Furthermore, in a virtualized environment, it requires extremely little hardware to show IDSs a productive location. In summary, both customers and providers may easily find their location.

Patel et al [22] asserteda unique combination of self-management and principles of Autonomic Computing (AC) concept. This model is distinctive as it is built on a self-managed method of anomaly instruction prevention system and risk assessment. By taking up Autonomic Computing, breach is controlled and checked. In addition to this, it also enhances the intrusion detection mechanism by shielding the data stored in the system. The major application of this is in the field of digital forensics and investigations. The traditional IDS security frameworks place a lot of emphasis on enhancing network dependability and security using various methods

Okikiola et al. [23] created a solution that utilises watermark extraction and logging detection to detect insider assaults in Cloud-based Healthcare systems. Microsoft Azure, an Opennebula emulator, PHP, and MySQL were used to create the framework. The volume of user action and the number of both approved and illegitimate system intrusions were revealed via an audit trail. The final examination showed that the method had excellent precision, recall, and accuracy, making it a great candidate for implementation.[31],[32].

Chakraborty& Kishor [24] The use of machine learning (ML) classification algorithms as a means of diagnosing cardiac disease is the objective. Cloud-based diagnostics based on IoMT have been suggested as a treatment for cardiovascular disease. In order to swiftly classify patient data using ML, fog layer is deployed.

In Kishor &Chakarbarty [25], Smart Ant Colony Optimisation (SACO) is introduced as a task offloading solution for IoT-sensor applications; it operates as a meta-heuristic planner in a fog setting. The suggested method is evaluated alongside two bio-inspired algorithms—the modified particle swarm optimisation (MPSO) and the Bee life algorithm (BLA)—and the classic Round Robin (RR). In comparison to RR, throttled MPSO, and BLA, SACO reduces latency while offloading workloads for IoT sensor applications.

Kishor &Chakarbarty [26] utilises reinforcement learning-based multimedia data segregation (RLMDS) and the Computing QoS in Medical Information System Using Fuzzy (CQMISF) method to improve service quality over a distributed network. Classifying healthcare data, selecting optimal gateways for data transfer, and improving transmission quality by accounting for throughput, end-to-end delay, and jitter are the three phases of operation for the proposed methods. algorithms were proposed for sorting medical records by risk and providing just the most urgent information to the end user over the most secure channel.In Table 1, we see a comparison of the current methods used to identify insider threats.

.

Table 1: Comparative Analysis of the Existing Approaches

Author	Techniques used	Results	Disadvantages
Gunasekhar et al (2015)	Ensures the security, integrity, access control, and confidentiality of cloud clients' sensitive data by utilising several cloud service providers.	50 % of government websites are vulnerable, with no security safeguards in place.	Even while this system guarantees security for stored data, there are many worries about key management and the possibility of side channel attacks.
Adrian Duncan et al (2019)	Detection Architecture Based on Attack Trees and Kill Chains	It enhances broad knowledge of the attack while also assisting in clearly identifying important nodes.	Extremely complicated and tough to navigate
Iyengar (2022)	Defense system based on fuzzy logic	Detect malicious packets and implement appropriate	The accuracy of these systems is jeopardized since they rely on

		countermeasures to counteract the DDoS attack.	erroneous data and inputs.
Chiba et al (2016)	CH-NIDS identifies network threats cooperatively and hybridly.	IDSs may identify unknown assaults. This lowers processing expenses for detecting intrusions at other IDS and enhances Cloud detection rate.	It is susceptible to noise in data.
Okikiola et al (2020)	Watermarking extraction and logging detection for cloud-based healthcare insider attacks	The strategy's precision, recall, and accuracy were excellent in the researchs assessment, making it an optimal performance.	It achieves greater performance across all metrics.
Chakraborty & Kishor, A. (2022).	Classification algorithms in machine learning (ML) for cardiovascular disease prediction	Gets 97.32 percent exact, 97.58 percent recall, 97.16 percent precision, 97.37 percent F1-measure, 96.6 percent sensitivity, and 97.2 percent mean	No interpretability, overfitting can easily occur,

The models stated above centre around single threaded and are seen as ineffective techniques of dealing with a vibrant and dispersed world. The offered solutions have limitations, particularly when dealing with Clouds, which are regarded as large data repositories. Furthermore, this solution requires the purchase of a third-party service that detects and sends recurring notifications. As a result, the current research focuses on providing effective IDS that is based on a water marking defence mechanism and includes a built-in third-party monitoring service.

3. Proposed Methodology

This section intensely discusses the research methodology involved in this proposal. At the outset, this research work highlights on the method of recreating an inventive medical data storage system which is used for detecting insider attacks. This model is recreated by enabling efficient resource allocation module along with watermarking techniques with Intrusion Detection System (IDS) for Cloud-based Healthcare system.

Initial efforts used a model for scheduling tasks and allocating resources that relied on the Convolutional Neural Network and the Improved Cuckoo Search Algorithm (ICSA). In later years, the LSBs of photos were watermarked using a method called watermarking. In conclusion, an intrusion attack detection model is created for data notification in e-Health care systems with the aim of identifying the attackers. The primary motivation for developing watermarking was to facilitate data manipulation and IDS strategy. It would also aid in spotting a potential invader.

3.1. Resource Allocation andTask Scheduling Framework

For example, given that T_n represents the 'nth' independent task of the user, $T = \{T_1, T_2, T_3, \ldots\ldots\ldots T_n\}$ is a valid data frame. It's important to remember that each data centre server is equipped with a wide variety of capabilities. A data centre is set up to provide the infrastructure required to support several servers. Then, the computing resources are allocated using the resource allocation technique. The collection of virtual machines defined by $VM = \{VM_1, VM_2, VM_3, \ldots\ldots, VM_n\}$ is to be noted here. Task scheduling and resource allocation are modelled in this work.

$$\text{Minimize } T_c \quad (1)$$

$$\text{If } D_{T_i} < D_m \text{ then put into short Q} \quad (2)$$

$$\text{If } D_{T_i} < D_m \text{ then put into log Q} \quad (3)$$

$$\text{Maximize } VM_U \quad (4)$$

The major objective of Eq (4) is to reduce the total completion time (T_c). When a task's goal value (D_{T_i}) is less than the median goal value (D_m), jobs are distributed using short queues. When a task's due date is farther away than the median due date, a lengthy queue is created. To reduce energy usage, virtual machines VM benefit from an expanded action and state space (VM_U).

3.1.1. Task Queuing using Median Deadline

After incoming tasks have been submitted to the global queue (G_Q), they are further divided into either a short queue or a long queue based on their due date. These incoming jobs are split into two queues, each with its own optimal time, and their due dates are represented as $D_T = d_{T_1}, d_{T_2}, d_{T_3}, \ldots\ldots, d_{T_n}$. For task queuing, we determine the median deadline value, or (D_m), of the tasks that are included. There is a wide variety of work in both the long and the short queues. Those tasks with a deadline that is less than the deadline median value will be placed in the short queue (S_Q), while those with a deadline that is more than the deadline median value will be placed in the long queue (L_Q). This information is then prioritised using a Binary In-order Traversal tree. The deadlines for each work are compared to the median timeframe. Since reducing buffering time is of paramount importance, shorter queues are paired with more manageable activities.

This ensures that these responsibilities are given top priority. In addition, tasks with later due dates are lined up in a long queue.

3.2. Task Scheduling Using Binary In-order Traversal Tree

By quantifying task weighting values, Binary In-order Traversal Tree schedules them. For weighted value estimation, the following task criteria are considered. length of the task (l), tardiness (t), makes pan (m), and slack time (s). The construction of a binary search tree for task scheduling results in a flawless schedule with unwavering quality. Examine the user tasks that have been requested first, then take the appropriate action from the tree to execute it.

Definition 1. In a tree (T), each node (A) includes a key (Key (A)). Instead of root node, the set of nodes has a parent P (A). In the elements of tree T, a right child node (Right (A)) or/and a left child node (Left (A)) might existed.

For each node A, all nodes B in the left subtree of A are the same, $Key(B) \leq Key(A)$. For all nodes B in the right subtree of A $Key(B) \geq Key(A)$ Tasks from the short queue $S_Q = \{T_1, T_2, T_3, \ldots\ldots T_n\}$ are moved towards the scheduler for scheduling. Besides, $L = \{L_1, L_2, L_3, \ldots\ldots L_n\}$ signifies that each task comprises the associated task length; $M = \{m_1, m_2, m_3, \ldots\ldots m_n\}$ denotes the Makespan value (i.e. completion time of task). An arrival time between tasks (i.e., delay) indicates Tardiness value, which has particularly referred to the variance within a late job's due data and its accomplishment time that has characterized by $T = \{T_1, T_2, T_3, \ldots\ldots T_n\}$. The slack time is considered to be an amount of time that the task can be delayed (but within the deadline) and the order of tasks in accordance with non-decreasing slack-time; the formulation of this process can be, $S = \{S_1, S_2, S_3, \ldots\ldots S_n\}$. Task duration, makespan, tardiness, and slack time estimate weighted value as follows,

Weight of task, $w = \frac{\sum l_i t_i m_i s_i}{\sqrt{\sum l_i m_i^2 \sum l_i t_i^2 \sum l_i s_i^2}}$ (5)

Here, the following notation: w for the task's weight; li for the ith task's length; mi for the ith task's maketime; ti for the ith task's tardiness; and si for the ith task's slack time. To be sure, the task scheduling algorithm has already placed constraints on the execution order before the scheduling process begins. To get around this problem, a tree has been constructed to display each value's weight. Each job's weight is represented in a Binary In-Order Traversal Tree node. In Binary in-order Trees, the Root node is where the initial weighted value is received. The subsequent work then evaluates how those weighted values stack up against the initial root node. The task's weighted value, if greater than the root value, must therefore be located at the right node of the root. If its value is less than the root's, however, the left-most node of the root will be given more weight. Therefore, tasks are planned using In-order traversal, where the job in the left node is scheduled first (because it is the child of the root node), and the work in the right node is scheduled last. If the weights of the tasks in the short queue are TW = [9,2,7,3,5,4,1,8,6], then a Binary In-order Traversal Tree (see Figure 2) will be used to schedule the tasks.

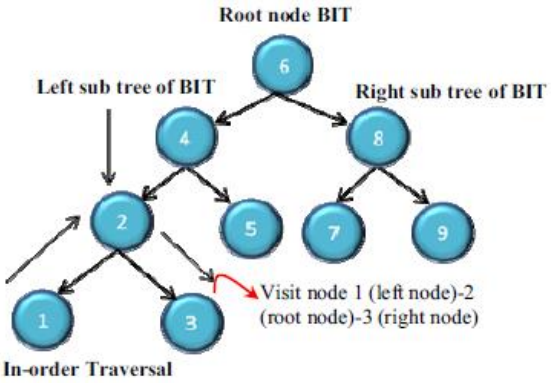

Fig 2: Task Scheduling using Binary In-order Traversal Tree (BIT)

Scheduling steps for the weighted value w=6, which puts the task on the root node, are displayed in Figure 2. It appears to have been positioned on the right of the root node when w=8, since its weighted value is more than that of the root node, and to the left of the root node when w=4. This method is repeated until the weighted values of completed jobs converge on a single node in the binary tree. Next, the tasks were scheduled as TW = 1, 2, 3, 4, 5, 6, 7, 8, 9 using the Left-Node; Root-Node; Right-Node technique. The procedure is shown in pseudocode below.

Algorithm 1. Pseudocode of TaskScheduling

Input: SQ and LQ
Output: Scheduled task
1. Begin
2. $S_Q = \{T_1, T_2, \ldots \ldots T_n\}$S
3. Get L,M,T,S
4. calculate Task's weighted values eqn.(5)
5. $T_W = T_{W1}, T_{W2}, \ldots \ldots T_{Wn}\}$
6. Assign
   $T_{W1} = root\ node$
7. If $(T_{Wi} > T_{Wj})$
   {Place in left node}
   Else
   {Place in right node}
8. Schedule Left node - Root node - Right node
9. End

3.3. Resource Allocation

For enhancing the resource utilization, the resources have classified as action space and state space. In action

space, allocated physical machines have involved, whereas the state space consists of idle physical machines, where the tasks have yet to be allocated. A resource predictor and overload predictor exist in each VM, which could be able to reveal the intensives of resource (Memory, I/O, and CPU). Then, as requested by the scheduled tasks, it explores the necessitated resource on the basis of ICSA. Post-identification of optimal VM, it has been allocated. In each PM existed in the state space, the overloaded predictors have correlated to the VMA manager that observes and handles the process of resource allocation. The migration count of VM can be minimized through the overloaded predictor, as it predicts the overloaded resource with ease. Following that, the overloaded VM has migrated from a PM to some other PM. Subsequently, the idle PMs have been shifted to sleep mode, concerning the preservation of power.

3.3.1. Deep Reinforcement Learning (DRL)

In DRL algorithms, the agents (i.e. RL agents) have utilized for leaning the optimal policy, through which the entire reward has augmented in the long run. In accordance with the values of action/state, the enhancement of optimal policy has done. However, there exists the exploration and exploitation trade-off between ICSA and DRL algorithms by reason of the searching nature. Generating the optimal and long lasting decisions (by learning through different scenarios) has vitally focused by the DRL (such as, a request pattern of user), which is appropriate to train many users alongside high speed convergence. DRL proves its significance in resource management due to its effective implementation of tasks, the constant enhancement of efficiency through learning from a specific user tasks (instead of FIFO order step), eradication of divergence and energy expense, and reduction of runtime.

To resolve the resource allocation problem in Cloud computing, presented the Automatic decision-making methods (e.g. Reinforcement Learning (RL)). There necessitates the identification of actions, reward functions, and states to recognize the loss of each action during the training process, besides to acquire the optimal policy while adopting an ACO. Nevertheless, high dimensions belong to state and action spaces have revealed by the entire system of cloud resource allocation, through which the utilization of conventional RL approaches can be prohibited. The DRL is very popular today because of its ability to solve complex consequential decision-making problems. Because DRL can learn different levels of data abstractions, it can solve many complicated tasks with accuracy and faster.

The resources have classified as action space and state space. In action space, VMs have allocated to the tasks previously. Later on that, the unallocated VMs (i.e. idle resources) existed in state space have been assigned to the scheduled tasks. Estimating the utilization of each VM has carried out through associated CPU intensive, Memory intensive and I/O intensive utilizations.

$$VM_U = U_{VM_i}^{CPU} + U_{VM_i}^{M} + U_{VM_i}^{I/O} \qquad (6)$$

Where,

$VM_U -$ Virtual machine utilization

$U_{VM_i}^{CPU} -$ CPU utilization of i[th] virtual machine

$U_{VM_i}^{M} -$ Memory utilization of i[th] virtual machine

$U_{VM_i}^{I/O} -$ I/O utilization of i[th] virtual machine

Figure 3 shows the action space if the state space has completely assigned VMs.

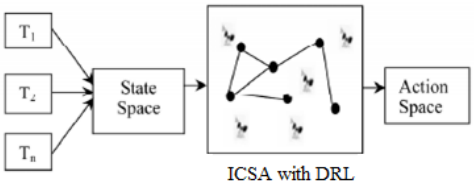

Fig 3: Resource allocation using ICSA with DRL

3.3.2. The Use of Deep Reinforcement Learning for Resource Management

State Space: It has divided the virtual machines into two zones (state and action) based on the resources they use the most. Define the state of the server cluster at the arrival time of the task T_t, S^{T_t}, and the state of the task T, S_T, as the union, i.e., $S_T = S^{T_t} = S_C^{t_T} \cup S_T$. The entire server can be partitioned into K equal subsets, labelled G1. Then, label the current status of G_K servers as g_kt. In addition, describe $U_{mR}^{t_T}$ as the utilisation level of server m at time t, and assign U_{TR} as the utilisation requirement of resource type R of task T. Because of this, U_{TR} as $U_{mR}^{t_T}$ represents the state of the DRL-based cloud service allocation tier S^{T_t}. It has taken the shape of,

$$S^{t_T} = S_C^{t_T}, S_T] = [g_1^{t_T}, \dots\dots\dots g_k^{t_T}, S_T] \qquad (7)$$

$$= [U_{11}^{t_T}, \dots\dots, U_{1|D|}^{t_T}, \dots\dots,, U_{|M||D|}^{t_T}, U_{T_1}, \dots\dots U_{T|D|}, d_T] \qquad (8)$$

In which, the (estimated) task duration has denoted by d_T. The state space contains all possible states of VM with high dimension.

a) Action Space: In DRL, the definition of action space for cloud resource allocation can be allocated VMs. The definition of The action space for a cluster with N servers can be as follows,

$$A = \{a | a \in \{1,2, \dots\dots |N|\}\} \qquad (9)$$

In the state space, the process of resource allocation has performed through ICSA with DRL as depicted by figure 3. The efficient reduction of action space has illustrated by Figure 4, in which the reward refers to the range of the allocated and idle resources. In the reward function, two assumptions have entailed.

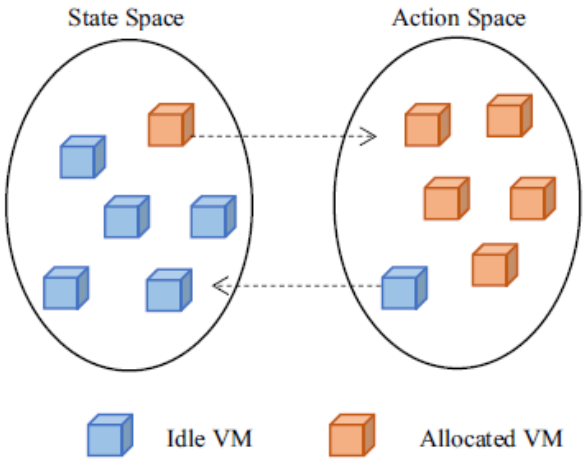

Fig 4: Transition of VM using DRL

- If VM resources have fully/partially allocated to perform their task (according to their necessity), that VM will be located over action space.
- If VM resources have not utilized by the tasks, that VM will be located over state space. During this overall process, the VMs exist in the state space has favored for the task allocation.

Transition: During the resource allocation, the transition refers to the transmission of VM from the state space to action space and vice versa. When a VM in the state apace becomes engaged, it has switched over to the action space; Conversely, if a VM in the action space turns out to be idle, it has transmitted to the state space.

$$VM_{Busy} \rightarrow AS \qquad (10)$$

$$VM_{Idle} \rightarrow SS \qquad (11)$$

In which, the action space has denoted by AS; state space has represented by SS. The transition of VM in cloud has demonstrated by Figure 4.

3.3.3. The Use of Improved Cuckoo Search Algorithm (ICSA) for Resource Allocation

The alleviation of resource complexity has carried out using Deep Reinforcement Learning algorithm. On the basis of ICSA with DRL, the allocation of resources has made to the required tasks in accordance with the placement of scheduled tasks over state space.

In cloud computing, the proposed DRL takes place to attain the goals (high-power consumption and less execution time) by conquering the challenges of overall resource allocation and power management. As specified previously, there are two spaces in proposed DRL model, i.e. state/action space. State space allocates VM resources to servers, whereas action space handles all servers' VMs. Moreover, the proposed ICSA significantly diminishes the search time by improving the scalability, and by reducing dimensions of states and action space, for which continuous-time and DRL technique has adopted. During the process, every single decision has taken through the arrival time of a new VM request.

- **Cuckoo Search Algorithm (ICSA)**

A unique optimization metaheuristic algorithm known as CS imitates the behaviour of cuckoos by searching for other birds' nests to lay its eggs. Host birds will raise cuckoo chicks. The cuckoo's eggs hatch faster in the host bird's nest [27]. Some varieties of cuckoos used to deposit their eggs in the nest of another bird, thus there was no difference between them and the eggs of the host species.

At the time of hatching, the cuckoo chick expels the chick/yet-to-hatch eggs of host bird impulsively in order to obtain the entire food. Moreover, it is able to mimic the call of host chicks. Once the host bird recognize that a cuckoo laid in its egg, the host bird throws the egg away from the nest or simply leaves the nest. Each nest represents a solution in this optimization procedure, and the following principles describe cuckoo reproduction.

- The egg has laid by each cuckoo in an arbitrarily selected nest
- The most successful cuckoo nests are passed down to the subsequent generations.
- The number of available host nests has set (limited), and the probability to identify the cuckoo's egg by the host bird is p_a that ranges 0, 1. Birds can find solely the poorest nests, hence they are out of the population

To keep things simple, a portion p an of the n nests that have been switched with new nests to provide fresh random solutions can be used to approximate the last assumption. However, it appears that CS has experienced the problem of early convergence because it is simple to trap it in local optima. To address these issues and enhance the performance of the CS, a cuckoo search algorithm based on the Elitism strategy (ES) has been created.

- **Cuckoo Search Algorithm based on ES**

Levy flight approach plays a vital role in CS. Because of this short-distance and sporadic long-distance cooperative random search mode, cuckoo nest search paths are likely to jump from one location to another, making CS advantageous for a robust global search. Due to this setting, CS exhibits a reliable random hop while conducting a search. As a result, searchers are looking in the vicinity of each weakly constructed cuckoo nest.

Therefore, the convergence of CS got delayed and possesses the insufficient accuracy of convergence. An Elitism strategy (ES) based improved CS algorithm has proposed to surpass the inadequacies, and for enhancing the CS performance, which sustains highly appropriate candidates and assists in exploration. Practically, a small value (e.g. one) has frequently defined as the degree of ES, number of elites. The ES is most prominent and widely utilized, as it keep on advancing the speed of convergence, aids to balance between exploration and exploitation due to its one-sided focus on exploitation. Consequently, to adapt ES, there necessitates the metric to improve exploration.

- **Mathematical Modeling of Elitism Strategy**

At the end of each iteration, ES automatically eradicates numerous weak nest locations, further generates the equal amount of new nest locations close to best nest location. This is reason for the existence of new nest locations has set across current optimal location, through which the algorithm performance has tried to be elevated.

The following equation expresses the estimation of new bird nest locations:

$$X_i^t = X_*^t + randn\left(1 - \frac{t}{T_{max}}\right)^p \qquad (12)$$

In which, the new location of the nest at time step t has denoted by X^t; the current optimum bird nest location has indicated by X^t; r *and* n obeys normal distribution with mean value is 0 and variance is 1; the maximum iterations has notated by T_{max}; regulatory factor signified as p, if $p = 2$.

The particulars about resource allocation using ICSA with DRL is specified through the pseudocode.

The summary of overall algorithm for resource allocation as trails,

Algorithm 2. Resource Allocation

Input: Scheduled tasks
Output: Resource allocated tasks
1.Begin
2. By using DRL, resources are divided into state space and action space
3. Define state space from (Eq.)
4. Compute reward function
5. Scheduled task entered into state space // improved cuckoo search algorithm for resource allocation
6. Initialize the requirement of tasks (CPU, MEm, I/O)
7. Compute the best solution S_{best}
8. Update fitness function
9. Get optimal resource if maximum iteration reached
10. Put allocated virtual machines into action space
11.Update the action space
12.End

3.4. Insider Attack Detection Framework with IDS based Water Marking Technique

Patients' treatment might be affected through data alteration bring about human casualties in emergency conditions. Signature data base is regarded as the promising solution to intrusion detection because of data the high dimensionality and prominent dynamicity encompassed in system. In practically all current healthcare intrusion detection systems, network flow measurements or patient biometric data are heavily incorporated into the dataset structure process. This research mainly concentrates on network and biometric metrics combination for features improvement. For appropriate besides realistic intrusion analysis is obtained through the amalgamation of network flow and biometrics information by means of collection besides new data analysis associated to healthcare.

The suspicious behavior is identified through IDS pertaining to monitoring, collection, log analysis, and network traffic investigation as well as process user action for suspicious behavior recognition. The capability of prior alarm upon exposure risks instigated through any attack is attained by IDS. Thereby alerting the system administrators for execution of respective response measurements which in turn cut back the system serious damage probability. In IDS, a sensor generates security events, a console monitors, alerts, and controls the sensor, and a central engine logs sensor events in a database and generates alerts from security events received.

3.4.1. Insider Attack Detection Framework

Medical Records security and privacy is assured through encryption approach. Records reading and encryption is achieved through this approach by merging the authorized users. The decryption of encrypted data is accomplished through authorized user who uses a key. The health centre location from where the data is sent does not affect authorized user for data accessing which is been directed to the cloud environment. Medium security is achieved through encryption nevertheless malicious insiders might pretense an attack on the data through modification deprived of being detected. The alterations made by the malicious insider can be detected nevertheless insider who accomplished the action was not distinguished through this watermarking and encryption approach. This description of model is specified in figure 5.

3.4.2. Medical Scenario Description

There are two scenarios in which a cloud environment is deployed in a medical setting. P, a patient, might have a few minor medical conditions but overall feels ill. He or she then decides to visit Clinic 1 for a checkup in anticipation of seeing Doctor 1, a specialist, to address a

medical condition. If P's medical record is R (i.e., medical data), then Doctor 1 was unable to identify P's proper medical problem. Doctor 1 might need to confer with particular specialists (Doctor 2 and Doctor 5) in Clinics 2 and 3. As a result, Clinic1 is forced to submit R of patient P to the reputable Cloud storage, where other Clinics can access it.

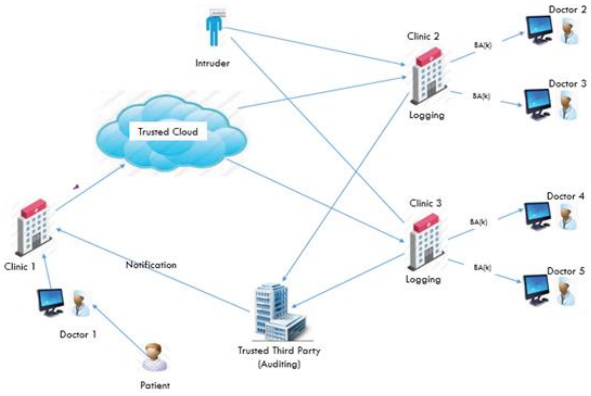

Fig 5: Proposed IDS and Watermarking based Insider Attack Detection Framework

Any user at Clinic2 or Clinic3 can utilise the login module shown in Figure 5 to keep track of when they made changes to patient records.When it comes to pinpointing the identity of the person responsible for the changes, biometric value, BA(k), is invaluable. In an encrypted system, key theft can occur even if the authorised user is unaware of the theft. There are three distinct components in this paradigm: a watermarking module for detecting tampering with data, an event logging module for monitoring the administration of biometrically-verified drugs, and a security module for encrypting data in transit.

3.4.3. Intrusion Detection System (IDS)

IDS are greatly involved in monitoring of Cloud networks for distinguishing malicious activity. The elimination of licensing new software, training new personnel, purchasing hardware, etc are accomplished through IDS , rather than securing user's critical applications and data [28]. IDS framework is considered to be the most reliable scheme since capability of Cloud users obtain opportunities and responsibility for governinqqqqqqqqg IDS. The User Data Collector (UDC), Cloud Service Component (CSC), and Cloud Intrusion Detection Component (CIDC) are the three main parts of an IDS framework. Confidential information sent between UDC and CSC is protected during transit thanks to the use of encrypted communication. The CIDBS structure is outlined in Figure 6, with further details provided in [29] - [30].

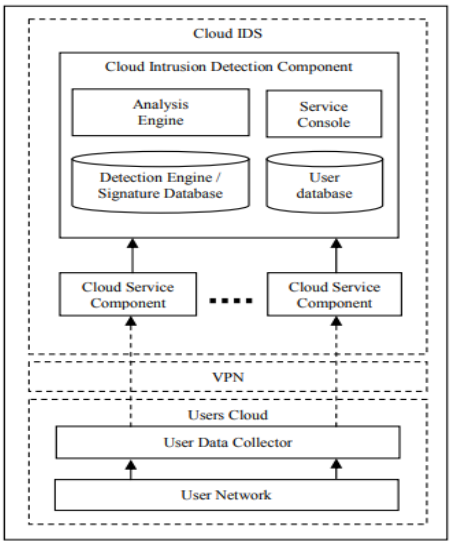

Fig 6: Intrusion Detection Service Framework

- **(UDC**

A secured independent server encompassing necessary information collection corresponds to UDC. On the basis of user requirements for whole network protection efficiently, UDC integration is done inside the user Clouds. The main function of UDC is to standardize as well as filtering packets information collection before being forwarded to Cloud IDS via a secure VPN connection.

- **CSC**

The external intrusion is identified, before decisive whether deletion of these information or forwarding them to appropriate analysis engine inside the Cloud Intrusion Detection Component (CIDC) is attained through CSC. It also helps in analysis and validation of received information. The received information translation into a common format is also acquired through CSC and that is implicit by the CIDC.

- **CIDC**

Intrusion detection relies on CIDC. Analytical Engine, Service Console, Signature Database, and User Database are CIDC's four main modules.

- **AE**

The Pattern matching mechanism is mainly accomplished by AE. The data analysis process obtained from CSC along with data pattern matching stored in the signature database is accomplished. Any sort of IDS is regarded as an independent process in AE e.g. Snort. Service Console possess an eminent role in intrusion activities identification through AE.

- **Service Console (SC)**

SC is regarded as remote user desired service controller. On the basis of user demands through SC, CIDC configuration for fine tuning is accomplished. The Privileged access to CIDC is attained through SC handling of user authentication. Reporting of Malicious activities to the user is done via SC. There exist various formats for IDS alerts generally besides SC signifies every alerts in a common format termed as Intrusion Detection Message Exchange Format (IDMEF).

- **Signature Database (SDB)**

SDB encompasses entire up-to-date attack signatures.

- **User Database (UDB)**

UDB contains the bulk of the various user data, such as login details and user activity. The CBIDS-based example scenario architecture is shown in Figure 7. Customers (Users) are primarily made up of representatives from SMEs. There are various prerequisites that must be met before CBIDS can be utilised. An intermediary server (proxy), virtual private network (VPN), or virtual switch connection must be assigned on the user's end of the network. The two primary parts are the users' cloud and the cloud intrusion detection system. Users' Cloud includes servers, hosts, and other networking components that operate in a virtual environment. The Users Cloud allows for the creation of a large number of servers and hosts.

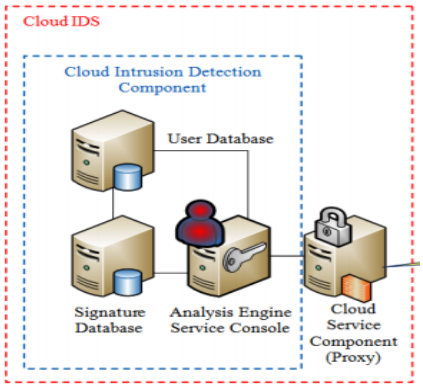

Fig 7: IDS Sample Architecture

Here it is presumed that each single machine acts as a Host VM and connection of each Host-VMto virtual switches is done for aiding SPAN features. The copying of sensitive information for inbound as well as outbound traffic of every Host-VM is done besides forwarded to the UDC (proxy). A secure VPN connection is established for gathering the information as well as forwarding it to the Cloud CSC (proxy) residing in the IDS Cloud by means of UDC. A comprehensive global defensive mechanism is provided by IDS Cloud and thereby detecting complete attacks on the basis of information forwarded by the user in real time. The filtering process is accomplished through CSC after arriving packets substantiation at the CSC before forwarding the essential besides suitable information to the AE. The matching of suitable information with the attack signatures stored in the signature database (SDB) is done through AE. A report is produced by AE once analysis process is completed or event to the UC for the user's attention is done. A web-enabled service (UC) is greatly utilized for configuration as well as updating their service requirements for affording particular access for users into the Cloud IDS

3.4.4. Watermarking Module (WM)

The realm of space On the basis of watermark embedding on images' least important bits, watermarking is selected (LSB). A very perceptible image quality is maintained to guarantee a high embedding capacity. This method's characteristics are that it embeds the region of non-interest and is invisible, blind, and fragile (RONI). The Human Visible System (HVS) is not given watermark visibility. Fragile watermarking is widely used to detect image file tampering or alteration whenever it occurs. Through the blind feature, the original image is obtained in addition to being hidden from a third party. Although not necessary for diagnosis, RONI feature aids in achieving watermark to be imprinted in the region. In addition to being embedded outside of this region, the region of interest (ROI) of RONI is assumed to be an elliptic region in the medical image. These photos are divided into ROI and RONI before the watermark is inserted. Through the addition of a watermark to the image pixel LSB, the segmentation problem is resolved. Similar techniques are used to identify changes and confirm the accuracy of medical data. Watermark embedding in LSB makes extensive use of the image boundaries. Equation 13 can be used to determine the number of bits that can be included in an image when using d bits as the bit depth.

$$B = d \times (2Nw + 2Mw - 4w^2) \quad (13)$$

Where M is the number of rows, N the number of columns, w the border width, B the amount of bits that can be inserted into the image, and d the number of LSB bit planes that are used to embed it. The watermark can be incorporated into the image by increasing its width or depth.

Algorithm 3. Watermark Preparation and Embedding

Input: Original image (Med_Img), Patient information (Med_info)
Output: watermarked med_img
1.Begin
2. upload Med_Img
3. Compute number of bits, B
4. Create 16 bits header from bits
5. Generate hash function using 160 bits of data by applying hash function and encryption key K1
6. Set the two LSB bits of med_img to zero
7. Apply SHA1 to obtained image generating a 160 bits hash
8. Encrypt hash using a key K1 to be shared by the doctors. (128 bits of AES algorithm)
9. Upload med_info
10. Generate payload=binary(med_info)
11. Embed watermark to border region of host file, size of (header +hash + payload)
12. Return water markedmed_img, payload

Algorithm 4. Watermark Extraction

Input: Original image med_img
Output:extraction report
1.Begin
2. Extract the first 16 bits from LSB of watermarked med_img
3. Compute load length = decimal (extracted 16 bits)
4. Extract load from watermarked med_img
extracted watermark=header+hash+payload
extract h1= hash
5. Find the hash of the image after setting two LSB bits to aero
6. Encrypt using AES algorithm with shared key K1, h2
7. if h1=h2

4. Results And Discussion

This segment assesses the performance of the proposed resource allocation and task scheduling based security approach through experiments and couple of simulations. The entire simulation has carried out under cloud environment using Cloud Sim that handles the scientific workflow implementation through its extensions. During the workflow implementation, many task parameters have included to attain the task requests. In the second phase, Virtual Machine and the server parameters, such as baseline energy consumption rate, number of VMs, and mode of server have included. Further involved the table of resource allocation, Policy of Private Message (PM) mode switch.

Table 2: Parameter Settings

Parameter item	Domain
Scientific workflows count	{0,4,8,12,16,20}
Tasks in every workflow	[5,25]
Required VMs for every task	[1,15]
Every task duration (hour)	[0.1,5.0]

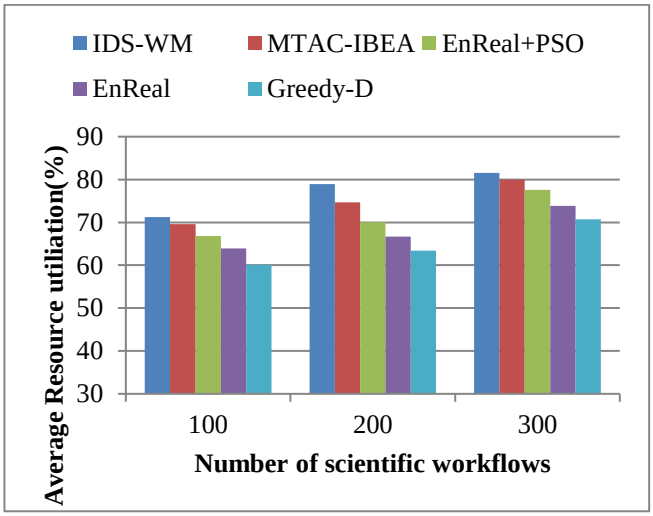

Fig 8: Comparison of Resource Utilization at Different Arrival Instants with Resource Utilization Methods

In figure 8, the chart demonstrates that the proposed Intrusion Detection System based Watermarking Module (IDS-WM) proves its efficiency to obtain 81.57% of average resource utilization for 20 hrs, which is superior to other approaches, like MTAC-IBEA, EnReal-PSO, EnReal and Greedy-D, as they solely delivers 79.982%, 77.566%, 73.88% and 70.75% for the value workflow of 300. Hence, the proposed Deep Reinforcement learning considerably augments the resource utilization, when compared to other approaches which is due to the searching nature of proposed algorithm.

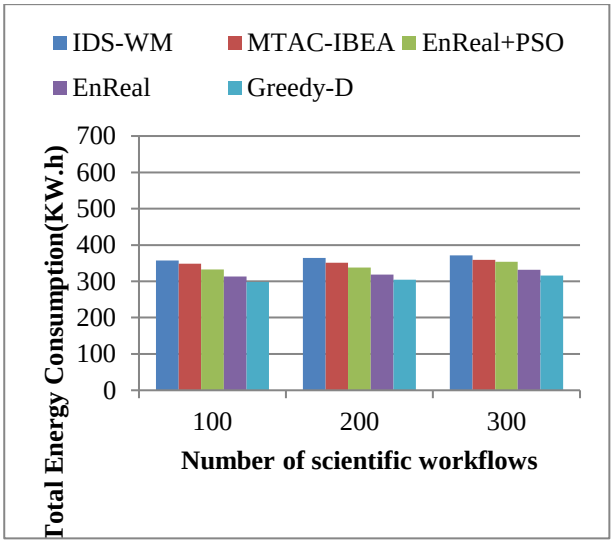

Fig 9: Comparison of Energy Consumption at Different Arrival Instants with Energy Consumption Methods

Figure 9 represents the optimal energy consumption rates obtained by the proposed approach at the time of maximizing arrival instance, which is better than other approaches. The chart depicts the capability of proposed Intrusion Detection System based Watermarking Module (IDS-WM) to attain371.65% of average resource utilization for 20 hrs, which is higher than other techniques, like MTAC-IBEA, EnReal-PSO, EnReal and Greedy-D, as they could deliver359.247%, 353.7%, 331.36667% and 316.15% for the value workflow of 300which is due to the searching nature of proposed algorithm.

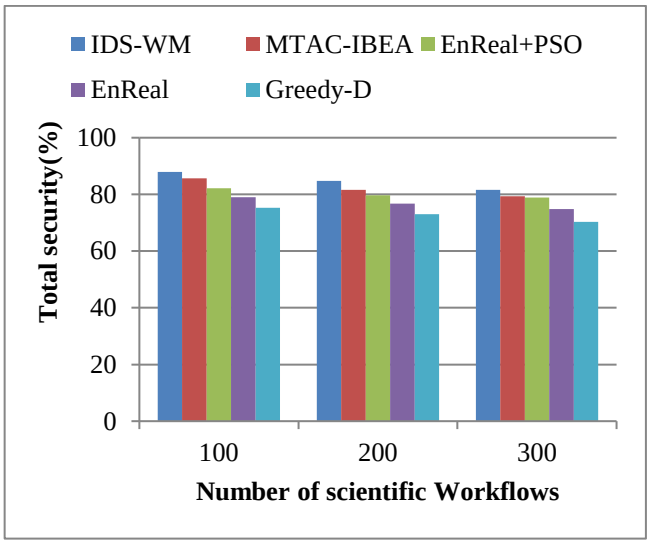

Fig 10: Comparison of Security at Different Arrival Instants with Security Methods

Figure 10 demonstrates the capability of proposed Intrusion Detection System based Watermarking Module (IDS-WM) to deliver optimal security against attackers using IDS. It has observed from the graph that the proposed insider attack detection system accompanying the IDS-based watermarking module can deliver maximum security and higher resource utilization than the other prevailing methodologies. The chart depicts the capability of proposed Intrusion Detection System based Watermarking Module (IDS-WM) to attain 70.27% of total security, which is higher than other techniques, like MTAC-IBEA, EnReal-PSO, EnReal and Greedy-D, as they could deliver 81.57%, 79.32%, 78.86% and 74.86% for the value workflow of 300 which is due to the searching nature of proposed algorithm.

5. Conclusion

Throughout this study, the security models and security requirements has scrutinized using a methodical approach, exclusively for Cloud-based Healthcare application. Besides, to attain the efficient medical data storage and to identify the insider attacks, a novel strategy has proposed, which uses efficient resource allocation module and watermarking methods alongside Intrusion Detection System (IDS). In the initial phase of this research, a task scheduling and resource allocation model has employed on the basis of ICSA and CNN. Further involved watermarking method, which is spatial domain that relies on the embedment of watermark over Least Significant Bits (LSB) of images. In addition, the intrusion attack detection model is introduced to cater for logging and imperfect detection of attackers, concerning information notification in eHealth care system. Here proposed a strategy to identify the attackers with the help of watermarking and logging technique. The data modification attacks have handled by the watermarking method, and detection of intruder has carried out through logging method. An empirical finding depicts the satisfactory level of ease, efficiency, effectiveness, and immense benefit to the medical domain possessed by the proposed approach. For instance, the result demonstrates that the proposed Intrusion Detection System based Watermarking Module (IDS-WM) proves its efficiency to obtain 81.57% of average resource utilization for 20 hrs, which is superior to other approaches, like MTAC-IBEA, EnReal-PSO, EnReal and Greedy-D, as they solely delivers 79.982%, 77.566%, 73.88% and 70.75% for the value workflow of 300. The assessment demonstrates the efficiency of the proposed model to have maximum accuracy and precision level.In future, this study can be further extended through executing the proposed system in some other real time applications, apart from healthcare. The enhancement of access speed can be focused, in future. In addition, this system can be considered for being executed over e-voting system, which necessitates the maximum trust within its operations.

References

[1] D. Ergu, G. Kou, Y. Peng, Y. Shi, Y. Shi, The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment, The Journal of Supercomputing, **64**(3): 835-848, 2013, doi: 10.1007/s11227-011-0625-1.

[2] P. S. Pillai, S. Rao, Resource allocation in cloud computing using the uncertainty principle of game theory, IEEE Systems Journal, **10**(2): 637-648, 2014, doi: 10.1109/JSYST.2014.2314861.

[3] S. Vakilinia, B. Heidarpour, M. Cheriet, Energy efficient resource allocation in cloud computing environments, IEEE Access, **4** (12): 8544-8557, 2016, doi: 10.1109/ACCESS.2016.2633558.

[4] P. Mishra, E. S. Pilli, V. Varadharajan, U. Tupakula, (2017). Intrusion detection techniques in cloud environment: A survey, Journal of Network and Computer Applications, **77**(1): 18-47, 2017, doi: https://doi.org/10.1016/j.jnca.2016.10.015.

[5] D. Zissis, D. Lekkas, Addressing cloud computing security issues, Future Generation computer systems, **28**(3): 583-592, 2012, doi: https://doi.org/10.1016/j.future.2010.12.006.

[6] K. Popović, Ž. Hocenski, Cloud computing security issues and challenges, In The 33rd International Convention Mipro, Opatija, Croatia, pp. 344-349, 2010, doi: https://ieeexplore.ieee.org/abstract/document/553331.

[7] M. Almorsy, J. Grundy, I. Müller, An analysis of the cloud computing security problem, Published in in Proceedings of the APSEC 2010 Cloud Workshop, Sydney, Australia, pp. 1-6, 2016, doi: https://doi.org/10.48550/arXiv.1609.01107.

[8] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud, Journal of Network and Computer Applications, **36**(1): 42-57, 2013, doi: https://doi.org/10.1016/j.jnca.2012.05.003.

[9] R. Arora, A. Parashar, CCI transforming, Secure user data in cloud computing using encryption algorithms, International Journal of Engineering Research and Applications, **3**(4): 1922-1926, 2013, doi: 10.4236/jis.2015.61002.

[10] D. Y. Chang, M. Benantar, J. Y. C. Chang, V. Venkataramappa, U.S. Patent No. 8,769,622. Washington, DC: U.S. Patent and Trademark Office, 2014.

[11] M. Almorsy, J. Grundy, A. S. Ibrahim, Collaboration-based cloud computing security management framework, In 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, pp. 364-371, 2011, doi: 10.1109/CLOUD.2011.9.

[12] T. Gunasekhar, K. T. Rao, V. K. Reddy, B. T. Rao, Mitigation of insider attacks through multi-cloud, International Journal of Electrical & Computer Engineering, **5**(1): 136-141, 2015, doi: https://core.ac.uk/reader/334419680.

[13] Z. M. Yusop, J. Abawajy, Analysis of insiders attack mitigation strategies, Procedia-Social and Behavioral Sciences, **129**(5): 581-591, 2014, doi: https://doi.org/10.1016/j.sbspro.2014.03.716.

[14] A. Duncan, S. Creese, M. Goldsmith, A combined attack-tree and kill-chain approach to designing attack-detection strategies for malicious insiders in cloud computing, In 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, pp. 1-9, 2019, doi: 10.1109/CyberSecPODS.2019.8885401.

[15] N. C. S. Iyengar, A. Banerjee, G. Ganapathy, A fuzzy logic based defense mechanism against distributed denial of service attack in cloud computing environment, International Journal of Communication Networks and Information Security, **6**(3), 233, 2014, doi: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=01be7ab80b1ab81d9ba07d5e1ab96ac95bd885de.

[16] C. C. Lo, C. C. Huang, J. Ku, A cooperative intrusion detection system framework for cloud computing networks, In 2010 39th International Conference on Parallel Processing Workshops, San Diego, CA, USA, pp. 280-284, 2010, doi: 10.1109/ICPPW.2010.46.

[17] A. V. Dastjerdi, K. A. Bakar, S. G. H. Tabatabaei, Distributed intrusion detection in clouds using mobile agents, In 2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences, Sliema, Malta, pp. 175-180, 2009, doi: 10.1109/ADVCOMP.2009.34.

[18] Z. Chiba, N. Abghour, K. Moussaid, M. Rida, A cooperative and hybrid network intrusion detection framework in cloud computing based on snort and optimized back propagation neural network, Procedia Computer Science, **83**(1): 1200-1206, 2016, doi: https://doi.org/10.1016/j.procs.2016.04.249.

[19] S. Roschke, F. Cheng, C. Meinel, An extensible and virtualization-compatible IDS management architecture, In 2009 Fifth International Conference on Information Assurance and Security, Xi'an, China, pp. 130-134, 2009, doi: 10.1109/IAS.2009.151.

[20] C. N. Modi, D. Patel, A novel hybrid-network intrusion detection system (H-NIDS) in cloud computing, In 2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Singapore, pp. 23-30, 2013, doi: 10.1109/CICYBS.2013.6597201.

[21] U. Oktay, O. K. Sahingoz, Proxy network intrusion detection system for cloud computing, In 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), Konya, Turkey, pp. 98-104, 2013, doi: 10.1109/TAEECE.2013.6557203.

[22] A. Patel, Q. Qassim, Z. Shukor, J. Nogueira, J. Júnior, C. Wills, P. Federal, Autonomic agent-based self-managed intrusion detection and prevention system, In Proceedings of the South African Information Security Multi-Conference (SAISMC 2010), Port Elizabeth, South Africa, pp. 223-234, 2011.

[23] F. M. Okikiola, A. M. Mustapha, A. F. Akinsola, M. A. Sokunbi, A new framework for detecting insider attacks in cloud-based e-health care system, International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Ayobo, Nigeria, pp. 1-6, 2020, doi: 10.1109/ICMCECS47690.2020.240889.

[24] C. Chakraborty, A. Kishor, Real-time cloud-based patient-centric monitoring using computational health systems, IEEE Transactions on Computational Social Systems, **9**(6): 1613-1623, 2022, doi: 10.1109/TCSS.2022.3170375.

[25] A. Kishor, C. Chakarbarty, Task offloading in fog computing for using smart ant colony optimization, Wireless Personal Communications, pp. 1-22, 2021, doi: https://link.springer.com/article/10.1007/s11277-021-08714-7.

[26] A. Kishor, C. Chakraborty, W. Jeberson, Reinforcement learning for medical information processing over heterogeneous networks, Multimedia Tools and Applications, **80**(16): 23983-24004, , doi: https://link.springer.com/article/10.1007/s11042-021-10840-0.

[27] A. H. Gandomi, X. S. Yang, A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Engineering with Computers, **29**(1): 17-35, 2013, doi: https://link.springer.com/article/10.1007/s00366-011-0241-y.

[28] M. P. K. Shelke, M. S. Sontakke, A. D. Gawande, Intrusion detection system for cloud computing, International Journal of Scientific & Technology Research, **1**(4): 67-71, 2012, doi:

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d60f1d825f98e342cc90f5cc0498f9d544706e19.

[29] B. Brik, N. Lagraa, H. Cherroun, A. Lakas, Token-based clustered data gathering protocol (TCDGP) in vehicular networks, In 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, pp. 1070-1074, 2013, doi: 10.1109/IWCMC.2013.6583705.

[30] B. Brik, N. Lagraa, M. B. Yagoubi, A. Lakas, An efficient and robust clustered data gathering protocol (CDGP) for vehicular networks, In Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications, New York, NY, USA, pp. 69-74, 69–74, 2012, doi: https://doi.org/10.1145/2386958.2386969.

[31] Hemanand, D., Reddy, G. ., Babu, S. S. ., Balmuri, K. R. ., Chitra, T., & Gopalakrishnan, S. (2022). An Intelligent Intrusion Detection and Classification System using CSGO-LSVM Model for Wireless Sensor Networks (WSNs). International Journal of Intelligent Systems and Applications in Engineering, 10(3), 285–293. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2167

[32] S. Gopalakrishnan and P. M. Kumar, "An Improved velocity Energy-efficient and Link-aware Cluster-Tree based data collection scheme for mobile networks," 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2016, pp. 1-10, doi: 10.1109/ICACCS.2016.7586372.

[33] Russo, L., Kamińska, K., Christensen, M., Martínez, L., & Costa, A. Machine Learning for Real-Time Decision Support in Engineering Operations. Kuwait Journal of Machine Learning, 1(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/117

[34] Ricci, A., Jankowski, M., Pedersen, A., Sánchez, F., & Oliveira, F. Predicting Engineering Student Success using Machine Learning Algorithms. Kuwait Journal of Machine Learning, 1(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/118

[35] Rossi, G., Nowak, K., Nielsen, M., García, A., & Silva, J. Enhancing Collaborative Learning in Engineering Education with Machine Learning. Kuwait Journal of Machine Learning, 1(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/119

[36] Agrawal, S. A., Umbarkar, A. M., Sherie, N. P., Dharme, A. M., & Dhabliya, D. (2021).Statistical study of mechanical properties for corn fiber with reinforced of polypropylene fiber matrix composite. Materials Today: Proceedings, doi:10.1016/j.matpr.2020.12.1072