Journal of Theoretical and Applied Information Technology
31st May 2023. Vol.101. No 10

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

# EFFICIENT IOT-BASED CLOUD COMPUTING FRAMEWORK FOR SECURE DATA STORAGE USING MACHINE LEARNING ALGORITHM

**RUPALI S. PATIL[1*], AMINA KOTWAL[2], SWATI S. PATIL[3]**

[1*]Assistant Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India

[2]Assistant Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India

[3]Assistant Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India

Email: [1*]rupali.patil@bvcoenm.edu.in, [2]amina.qazi@bvcoenm.edu.in, [3]swati.patil@bvcoenm.edu.in

**ABSTRACT**

Cloud computing is a widely used technology that has changed the way people and organizations store and access information. This technology is versatile, and extensive amounts of data can be stored in the cloud. However, with the development of cloud computing, it is also faced with many difficulties, cloud computing security has become the leading cause of impeding its development. Cloud computing security has become a hot topic in industry and academic research. As a consequence, the security of data stored in the cloud serves as a key concern for cloud consumers due to ongoing hacking incidents in the cloud. This work used encryption with access management because authenticities, anonymity, and security over accessibility are mandatory. Accordingly, the article proposed a machine learning-based method for secure data storage in the cloud. Initially, the data is compressed using the Huffman algorithm, which minimizes text data size and storage, resource use, or transmission power. Accordingly, the compressed data are encrypted using a novel cryptographic technique. This method encrypts the data before uploading it onto the cloud. Subsequently, the malicious intention in the cloud platform is identified by proposing a Weighted Chimp Algorithm optimized Gaussian Kernel Radial Basis Function Neural Network. This malicious code can be spread through infrastructures in the cloud platforms and pose a great threat to users and enterprises. The proposed method accurately detects malicious code in the cloud. The proposed work is implemented using Python software. The proposed method is compared with the other existing methods like Fully Homomorphic Encryption (FHE), Ciphertext Policy-Attribute based Encryption (CP-ABE), and Quasi Modified Levy Flight Distribution Reversed Sheamir Algorithm (QMLFD-RSA). Accordingly, the proposed method outperforms these existing methods. The result revealed that the deduplication rate, throughput, cipher text and encryption time of the proposed method produce higher performance than the existing methods, ie) the deduplication rate for the proposed method is 94% and the outcome of the work proved that the proposed work produces better security than the other existing research respectively. This hybrid technique provides the user to get an advantage from retrieved information in a protected manner.

**Keywords:** *Cloud Computing, Security, Data Storage, Huffman Algorithm, Data Compression, Malicious Behaviour, Gaussian Kernel Radial Basis Function Neural Network, and Weighted Chimp Algorithm.*

## 1. INTRODUCTION

The Internet of Things (IoT) has evolved as a technology with a significant current-day application, and its use has resulted in a sustained increase in network traffic levels over time. In the coming years, many more devices are anticipated to become connected. The IoT paradigm places a strong emphasis on data since it can be used for a variety of applications, including manufacturing, transportation networks, smart cities, industries, and healthcare. In addition to threats to the confidentiality, integrity, and privacy of data, IoT must be protected from assaults that prevent it from providing the necessary services [1] [2]. Accessing data from any location with internet access is a key benefit of using the cloud. When considering the volume of data being moved between data centres and IoT nodes, guaranteeing security is another endeavour. A vulnerable network can result in unintended harm. Strong algorithm encryption is one potential method for preventing unauthorised

access to data. Common encryption techniques cannot scale for IoT devices since they have limited resources, primarily electricity, as was previously discussed [3] [4].

The use of cloud computing in bioinformatics is growing dramatically as a result of its many benefits, including low cost, scalability, high performance, infinite storage, and many others. However, there are several drawbacks to cloud computing, including issues with security, privacy, and transferability. Access control is one of the most important issues in the cloud computing environment among all of these issues. For the advantage of researchers, some of the well-known cloud-based bioinformatics apps are also introduced [5]. A flexible and dynamic access control paradigm for securing smart devices, data, and resources in the cloud-enabled IoT architecture is urgently needed to provide fine-grained access control and overcome the constraints of existing access control models [6]. The framework for IoT-based cloud computing for patient health monitoring has been proposed in [7] for efficient patient monitoring. The newest linked sensors and IoT gadgets keep a watch on the patient's eye movement, body temperature, and pulse rate. The impacts of each measure are recorded and the collected data are utilised to assess the patient's health in a cloud database. Cloud data storage must be protected to assure privacy and a stringent encryption policy must be enforced to guarantee safe data storage so that data leaking can be stopped and consistency is maintained [8].

Another field of study that has been effectively used to address numerous networking issues, including routing, traffic engineering, resource allocation, and security, is machine learning (ML). There has been a recent increase in the use of ML-based techniques to enhance various IoT applications. Despite the substantial research on big data analytics and machine learning, there aren't many studies that specifically address the development of ML-based methodologies for big data analysis in the IoT healthcare industry [9]. Using Ethereum blockchain technology, [10] provides distributed, safe, and authorised access to such sensitive data. For a healthcare application that stores and analyses electronic health records, an integrated low-powered IoT blockchain platform is created. In addition to the medical and paramedical personnel, may all securely obtain health information with the help of this architecture, which is built on the Ethereum blockchain.

In [11], a hierarchical method based on Software Defined Networking (SDN) is suggested for expediting data management and balancing the load not only across IoT devices of a single domain but also between multiple network clusters. The suggested design avoids having a single point of failure in the controller and enables the application of various load-balancing and management algorithms in hierarchical and multistep scenarios. End users of cloud services face a data security challenge since they have no control over their data once it has been sent to the cloud. When data in the cloud is encrypted using a user's key, forensic investigation teams cannot access it [12]. IoT-based Hierarchical Health Monitoring Model is suggested in [13] to obtain an appropriate evaluation of sports person health monitoring wearable while reducing energy usage. The introduction of the optimal energy-efficient resource assignment algorithm optimises the complexity of limited resources and energy utilization. With the aid of a deep learning approach and fuzzy rules with temporal information, a novel e-healthcare system is introduced in [14] for tracking the level of deceased diseases. This technology collects medical information from a variety of nearby individuals who are using e-healthcare aids. As a result of frequent hacking occurrences in the cloud, people are increasingly concerned about the security of data kept there. The company and the user are seriously threatened by several assaults that both cloud service providers and users must deal with [15].

To ensure data integrity and availability in the cloud and IoT storage system, users need to verify the integrity of remote data. However, the existing remote data integrity verification schemes are mostly based on the RSA and BLS signature mechanisms. The RSA-based scheme has too much computational overhead. The BLS signature-based scheme needs to adopt a specific hash function, and the batch signature efficiency in the big data environment is low. For secure data storage, the article proposed a symmetric jumbling-salting encryption algorithm that forms a highly secured form of encrypted password which makes it difficult to decrypt. Furthermore, it is necessary to set up a strong intrusion detection system (IDS) to recognise and thwart any attacks in the cloud at an early stage. Weighted Chimp Algorithm optimized Gaussian Kernel Radial Basis Function Neural Network is proposed to identify malicious behaviour in the cloud, this provides the high secure

data storage. The rest of the work is organized as follows, section 2 portrays the literature survey of the study, section 3 reveals the problem definition and motivation of the research, and Section 4 illustrates the proposed research methodology. Section 5 elucidates the experimentation and result in the discussion section, and section 6 demonstrates the conclusion of the research.

## 2. LITERATURE SURVEY

All manufacturing divisions, including the most recent and quickly developing technology of additive manufacturing (AM) or 3D printing, have the opportunity to modify their production processes thanks to cloud-based technologies for remote data collection, intelligent machine interconnectivity, and sensor monitoring. In the context of the AM industry, 4.0, Haghnegahdar et al [16] analyse the review of the cloud-based model and idea of cloud computing, cloud manufacturing, and IoT and their relationships and influences. This study also introduces CM apps and their connection with AM, as well as suggests a combined AM cloud platform. The smart IoT cloud frequently improves the standard spontaneous structure in addition to the traditional discovering institutions by treating mobile devices as corporate centres, such as by recognising institutions. This has numerous advantages right away, but many critical issues must be resolved before the unwavering consistency of the cloud environment and the network's decision-support system can be improved. The monitoring of load, resistance, and other security hazards in the cloud state presents similar challenges. Instead, the RSA encryption algorithm [17] will be used to transport the acquired data to the mobile cloud for storage.

As more businesses become aware of the IoT's revolutionary potential, they have started to discover several obstacles they must remove to utilize it effectively. Many businesses and organisations employ ML to tap into the untapped potential of the IoT. Mishra and Tyagi [18] assess the various machine-learning techniques that address the difficulties provided by managing IoT data. To overcome these issues, Almurisi and Tadisetty [19] investigate a practical solution based on cloud computing and virtualization approaches. The virtualization approach enables resources to be virtualized and shared between many applications, while cloud computing offers effective computer resources and enormous storage space. Public key encryption with an equality test based on DLP with double decomposition difficulties over near-ring was presented by Deverajan et al [20]. Additionally, the suggested system is very safe, guards against the Type-I rival are chosen-ciphertext attack and is indistinguishable from the type-II rival's random oracle model. The suggested method is extremely secure, and the security analysis controls are significantly more robust than those used currently.

Ali et al [21] have suggested a novel hybrid deep neural network-based group theory-based binary spring search (BSS) algorithm to address IoT network problems. The suggested method successfully finds the infiltration in the IoT network. Blockchain is employed as a database, but these solutions have mostly focused on data storage. For using the blockchain as a distributed database using homomorphic encryption to guarantee secure keyword searches and database access. Cloud-based data storage has recently gained increasing attention from academia and business due to its efficient and economical administration. Service providers must employ secure data storage and sharing procedures since services are given via an open network, protecting user privacy and the confidentiality of data. Two dual access control systems, one for each desired area, were designed by Babu et al [22]. The experimental and security assessments of the systems are also presented. However, there are several reasons why the cloud cannot be trusted, such as when it compromises user data or exposes private information. As a result, attribute-based encryption techniques are frequently used in cloud storage to achieve data secrecy and granular access control. To generate and manage partial keys, Lu et al [23] presented a revolutionary consortium blockchain-based searchable attribute encryption system that does away with the conventional key generation centre.

However, two main issues with IoT-enabled healthcare infrastructure are the security of patient data and the protection of user privacy. To safeguard healthcare data in IoT-enabled healthcare infrastructure, Das and Namasudra [24] presented a unique encryption approach employing elliptic curve cryptography, Advanced Encryption Standard (AES), and Serpent. By integrating both symmetric and asymmetric-based encryption approaches, the proposed hybrid encryption technique enhances security measures for healthcare data. Additionally, the proposed approach uses an elliptic curve-based digital signature to guarantee data integrity. Formal

security analyses as well as performance comparisons are offered to demonstrate the effectiveness of the suggested strategy. Due to the flaw in the authentication procedure, the safe authentication of these individuals is a significant challenge. As a result, [25] built an efficient hybrid and adaptive cryptographic-based security authentication framework to carry out an authentication procedure in IoT. The authentication procedure is carried out by the suggested method using cryptographic operations such as the exclusive-or operation, a hashing function, and hybrid encryption. The deployment of IoT devices depends on their security and ease of provisioning onto the cloud infrastructure. Mamvong et al [26] implemented a low-cost client-side encryption algorithm based on the AES to carry out data encryption the IoT devices due to the lack of information on the ease of secure provisioning of IoT devices. The inherent challenge these devices face in encrypting data before transmission to the cloud is due to their constrained nature.

## 3. RESEARCH PROBLEM DEFINITION AND MOTIVATION

Today's Machine Learning (ML) is a blend with the Internet of Things (IoT) based cloud applications which play a significant role in our everyday life. As indicated by Gartner's recent study, there are around 25 billion devices and a gadget interfacing with IoT including wearables and automated vehicles to smart homes and smart cities applications. All such connected (smart) devices generate immense data that needs to be examined and analyzed, to ensure that they continually learn from the available data sets and better themselves without any manual interference. This is where the prerequisite for machine learning comes into being. Numerous ML algorithms and techniques are introduced in a short time to easily evaluate big data measurements, increasing the IoT's productivity. The various concerns surrounding this area, specifically, the challenges of integrity, security, confidentiality, and authentication have been addressed. Due to the lack of trust in cloud-based storage services, data confidentiality must be protected or well-preserved before outsourcing sensitive data to the cloud.

With the tremendous increase in the amount of data, there is a higher requirement to process this huge amount of data (generated through billion of ICDs) using efficient ML algorithms. In the past decade, we refer to data mining (DM) algorithms to make some decisions from collected data sets. But, due to increasing data on a large scale, DM fails to handle this data. So, as a substitute for DM algorithms and to refine this information efficiently, we require traditional analytics algorithms, i.e., Machine learning algorithms with a cloud computing process. Cloud computing provides efficient computing resources and huge storage space and provides a secure system. People can accept IoT services via cloud servers anytime and anywhere in the IoT-based cloud computing environment. However, plenty of possible network attacks threaten the security of users and cloud servers. To implement effective access control and secure communication in the IoT-based cloud computing environment, identity authentication is essential. Consequently, the research proposed a secure IoT-based cloud computing platform for data storage, authentication and access control.

## 4. PROPOSED RESEARCH METHODOLOGY

IoT-based cloud computing technology is providing the best possible practices which are focused on market research. Cloud computing usually provides tools as entities on call, snappy transmission, and charge as a need to enhance user service. The basic idea behind IoT and cloud storage is to make day-to-day operations more efficient without compromising the quality of the stored or transmitted data. However, the study requires a secure method of sending data to the cloud. This paper aims to create a data security model based on cryptography and intrusion detection for data in cloud computing that seeks to reduce existing security and privacy concerns, such as data loss, data manipulation, and data theft. Cloud security depends on reliable computing and encryption. Only the approved user can access the data in the proposed work. However, if any intruder (unauthorized user) tries to attack the system to collect sensitive data, the original information will be secure and cannot be recovered. Authentication is the initial and imperative step in rendering access to approved users. Cloud storage is a new concept developed based on the concept of cloud computing with an efficient encryption algorithm in a hybrid manner. The core of the cloud computing platform system is to calculate and manage massive data. If a large number of storage-related devices are configured, the cloud computing platform system becomes the current one. The block diagram of the proposed work is presented in figure 1.

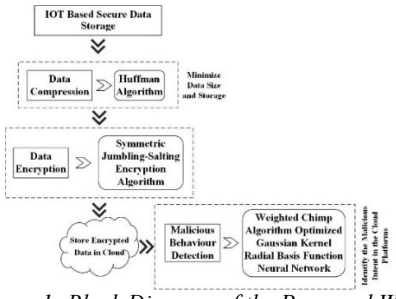


*Figure 1: Block Diagram of the Proposed Work*

A Cloud storage system is an integrated system that provides a large number of different types of storage-related devices together to provide massive data storage and business processing functions to the application layer through grid computing, parallel processing, distributed systems, cluster applications, application software and other functions. The core of the cloud storage system is to combine various application software and storage-related device management and transform storage-related devices into a storage service through various application software. In this work, a novel data encryption method is presented to encrypt the data to securely store it in the cloud. Accordingly, the weighted chimp algorithm optimized Gaussian kernel radial basis function neural network (RBFNN) is presented for identifying malicious behaviour in the cloud. Initially, data compression is presented to compress the data or minimize the data size and storage in the cloud. Subsequently, the above-mentioned methods are portrayed in the following sections.

### 4.1 Data Compression

Data compression is the technique used to restrict the redundancies of data in data representation and the way to decrease data storage requirements and reduce communication costs also. Reducing the storage requirement is equal to increasing the functionality of the storage medium and subsequently strengthening communication bandwidth. Data compression intends to limit the redundancy of the data present in the message to be capable to store or transmit data in an efficient form. However, Cloud storage allows organizations to store data at remote sites of service providers. Although cloud storage services offer numerous benefits, they also involve new risks and challenges concerning data security and privacy aspects. Data compression decreases data representation duplication, thus increasing usable data volume and helping to minimize text data size and storage, resource use, or transmission power. The classified unbalanced data are compressed by the Huffman algorithm.

#### 4.1.1 Huffman Algorithm

The Huffman algorithm is a data compression method. When using the Huffman algorithm for data compression, the average code length of the data will not change, so this advantage is the coding efficiency uniqueness, which can significantly reduce the data storage space and improve the speed of the data query in the storage mode. In addition, the Huffman algorithm constructs the code word with the shortest average length of different characters based on the character appearance probability, which is more accurate. After the above classification processing, the data is transformed by the wavelet decomposition method, and then Huffman coding. In the process of data compression using the combination of these two methods, the scale of wavelet decomposition should be smaller to reduce the amount of calculation of wavelet transform; binary coding of the transformed data can further improve the compression ratio.


*Figure 2: Flow Chart of Unbalanced Big Data Compression*

Huffman coding algorithm adopts optimized static coding technology, and the binary tree generated by the algorithm has the minimum weighted sum. The algorithm first arranges all the data in descending probability order, establishes a list, and then constructs a tree from bottom to top. Figure 2 is the flowchart of Huffman algorithm data compression. According to the unbalanced data compression process, as shown in figure 2, the Huffman algorithm of each leaf is placed in a tree, and then the tree determines the coding of the original data. All the obtained codes form a two-dimensional table. The size of the table is very small relative to the original data. The two-dimensional table and the encoded data are stored together or sent to the remote end through the communication network. The decoder does not need to traverse the tree but decompress it by looking up the table.

Huffman code is a type of optimal prefix code commonly used for lossless data compression. The output from Huffman's algorithm can be viewed as a variable-length code table for encoding any source symbol. The Huffman procedure of obtaining the variable length code is given in the following algorithm.

**Steps for Huffman Coding**

1. Building a Huffman Tree from the input character.
2. Assigning code to the characters by traversing the Huffman tree.

The steps involved in the building of the Huffman tree are as follows: -

**Step 1:** Create a leaf node for each character of the given text. The leaf node of a character contains the frequency of the ongoing character.

**Step 2:** All the nodes are arranged in increasing order of their frequency value.

**Step 3:** A new internal node is created. The sum of the frequency of two nodes is the frequency of the new node. The first node is considered a left child and the other node is considered the right child of the newly created node.

**Step 4:** Repeating steps 2 and step 3 in all nodes form a single tree. Finally, the desired Huffman tree is acquired.

**4.2 Data Encryption**

Data encryption enables the protection of sensitive data and therefore addresses many privacy-related issues in cloud computing. To preserve confidentiality, data must be encrypted before outsourcing to the cloud. Although this approach protects the security and privacy aspects of data, it also impedes regular functions such as executing queries and performing analytical computations. In this paper, a novel cryptographic technique has been presented that uses client-side data encryption for encrypting the data before uploading it onto the cloud. A novel symmetric key crypto-graphic technique has been used to encrypt/decrypt the data before uploading it onto the cloud. Consequently, the research proposed Symmetric Jumbling-Salting Encryption Algorithm, which deals with randomization, the password encryption technique forms a highly secured form of encrypted password which makes it difficult to decrypt reducing the probability of guessing the password. The standalone version of the algorithm is evaluated against the parameters like encryption time, decryption time, throughput, size of cipher text generated etc. The cloud version of the algorithm is implemented using virtual machine containers and will be evaluated using cloud-specific parameters like scalability, heterogeneity etc.

**4.2.1 Symmetric Jumbling-Salting Encryption Algorithm**

The JS algorithm uses a symmetric key encryption strategy. The algorithm is divided into two blocks: 1) a Jumbling block and 2) a Salting block. The jumbling block is further divided into three other sub-blocks: a) Addition block, b) Selection block and c) Reverse block. The input to the JS algorithm is the plain-text password which is stored in the Process array. The description of the three blocks is given below:

**Jumbling Block:** Jumbled Block is a block where the random values are generated from a pre-defined set of characters, digits and special symbols. Jumbling block is responsible for prepending some characters from the character set and jumbling them with the help mod function. The jumbling block itself is a combination of three sub-blocks:

**Addition Sub-Block:** Original plaintext is added with an additional sub-block consisting of random values in this block.

**Selection Sub-Block:** This block is about selecting characters from the predefined character set A. The size of the character array is large and the character set for a particular password entry is different. The selection of characters is based on the random values which are generated 'l' times.

**Reverse Sub-Block:** This block reverses the entire process array based on some predefined condition. The predefined condition is to check the value of 'l' is even or odd. If the value of 'l' is even, then we reverse the process array else we keep it as it is.

**Salting Block:** Salt Block is the block where the salt to be added to for ciphering password will be stored. The objective of the salting block is to add a random string along with a jumbled version of the password. The criterion for the selection of salt is the user's sign-up timestamp value. The salt is added to make the password more complicated thereby making it difficult for the attacker to obtain it.

Initially, the Jumbling-salting algorithm was used for passwords, DNS Encryption, and Payment gateway. The random nature of the jumbling salting algorithm proved that security is ubiquitous. The entire process contains 4 actors involved in the process of encryption and decryption of JS algorithm viz. Client, Network, Server, Database. The role of the Jumbling salting algorithm is described in each step below.

**Client**

- This is the initial phase of the Jumbling salting algorithm where a large file is divided into several chunks. For the simulation of this research, the chunk size is kept very small and the overall file size is taken with some initial limit. Practically, the chunk size is very large and hence the selection of an appropriate chunk size can be a future enhancement of this algorithm.
- The random number is also generated from the client side as the principal requirement of an algorithm. The random value will be a practically large value with its minimum value greater than or equal to the size of the character in the chunks.
- For the first time, the random key is sent through the network with an asymmetric encryption algorithm to the server for the decryption process of the file. The key is initially stored in the database with a hash value. The hashing algorithm can be used in future to ensure the integrity of the principal random number throughout the network.
- On the client side, the Jumbling-salting algorithm is implemented. The algorithm consists of two major processes, viz. Jumbling and Salting. Jumbling includes the addition of random characters from the characters set. To select the appropriate characters, we call the predefined user-coded random function. Each value in the integer will be the appropriate index of the position of the character in the character set. The modulus mathematical function is used to jumble the characters in a linear position. The salting function is done to prepend the unique timestamp value and the process of jumbling is repeated. The reverse function reverses the entire characters in the chunks using some predefined function. Jumbled and salted chunks are then sent to the server side as an encrypted file.

**Network**

- The network is used to secure sending of principal random values to the server's database and to send the encrypted version of file chunks sequentially.
- The performance of the network is unpredictable and depends on the quality of service (QoS) parameters. Hence, a strategy must be designed to secure the transmission of the encrypted string and principal random number. Some chunks may be lost during the transmission so there is a need to have a proper identification number from the sender and receiver.

**Server**

- The server stores the principal random value and encrypted chunks of file securely on the database of the server.
- The server also checks the hash value of the received strings to make sure that data has not been modified in the network.
- Various hashing algorithms like SHA and MD5 can be implemented to check the integrity of the algorithm.
- The server also performs the process of decryption which is followed by encryption. The principal random number which was sent by the sender is used as a principal random decryption key for the process of decryption. The algorithm undergoes the processes like reversing the string, removal of timestamp salt and de-jumbling the characters.

**Database or Storage**

- The database is used for storing the key and encrypted string.
- The hash value associated with the string is also stored in a database.

### 4.3 Malicious Behaviour Detection

Cloud service providers should be concerned about abuse management prevention, and the user's behaviour should be identified, preventing a malicious user to take advantage of a cloud computing system for unlawful violence to crack the password and DDOS attacks. However, hackers increasingly tend to abuse and nefariously use cloud services by injecting malicious code. This malicious code can be spread through infrastructures in the cloud platforms and pose a great threat to users and enterprises. The malicious

attacker has malicious intentions despite his benefit. This type of attacker can spread malware in the system and disrupt or even collapse the system. Several organizations are concerned about the privacy of data on the cloud due to sophisticated threats like malicious insiders. Therefore, in this study, the user's behaviour to identify the malicious intent in the cloud platforms is suggested by using A Weighted Chimp Algorithm optimized Gaussian Kernel Radial Basis Function Neural Network (RBFNN).

### 4.3.1 Gaussian Kernel Radial Basis Function Neural Network

The radial basis function is essentially good for function approximation and highly suitable for identifying malicious intentions in the cloud. It is a three-layer feedforward network consisting of the input layer, the hidden layer and the output layer. The hidden layer performs a non-linear transformation of the input parameters while the output is a linear combiner of the outputs of the hidden layer. The radial basis function is efficient because the training is faster as it has one single hidden layer and so reduces all forms of complexities. The function of each node can also be easily interpreted in the RBF network. The figure of the radial basis function neural network is presented in figure 3.

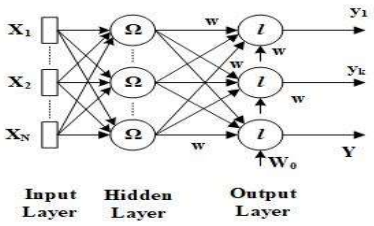


*Figure 3: Radial Basis Function Neural Network*

In an RBF network, the output of the input layer is obtained by the calculation of the distance from the network input to the hidden layer centre. The secondary layer is the linear hidden layer and the output of this layer is in the weighted form of the input layer output. Every neuron present in the hidden layer holds a parameter vector called the centre.

For data training and learning with the Gaussian kernel function. The output-weighted training points $\{t^p\}$. The radial basis function neural network $f(x)$ developed using the Gaussian kernel function as the weighted sum of Gaussian is given in (1)

$$f(x) = \sum_{n=1}^{K} W_p \emptyset_p(x) = \sum_{n=1}^{K} w_p exp\left(-\left\|\frac{x-v^n}{2\sigma^2}\right\|\right) \quad (1)$$

Where $\sigma$ is the width of the Gaussians, $w$ represents the weight of the function, $\emptyset$ is the interpolation matrix, the distances $\|x - v^n\|$ give the Euclidean distance between the input vector $x$ and the centre of the nth neuron in the hidden layer.

Minimization in the RBF achieved as given in (2)

$$E = \frac{1}{N}\sum_{i=1}^{N}\left\|y_p - w^T \emptyset_P\right\|^2 \quad (2)$$

Where, $y = [y_1, y_2, \cdots, y_N]$. Accordingly, the design process of the RBFNN involves determining the number of neurons present in the hidden layer. After that, the desired output obtained for RBFNN $w$, $\alpha$, $c$, and $\beta$ parameters may be accustomed appropriately. Reference-based error metrics namely mean square error (MSE) or sum square error (SSE) could be utilized to assess the network performance. Expression for error calculation of an RBF network is defined in the following equation (3):

$$E^{SSE}(w, \alpha, c, \beta) = \sum_{j=1}^{J}\left(y_i - i - \hat{y}_j\right)^2 \quad (3)$$

Here, $y_i$ denotes the required output and $\hat{y}_j$ denotes the RBFNN output. The training procedure of the RBFNN includes the reduction of the error function. RBF network is extensively implemented in approximation function for a given input-output pattern. Usually, the conventional RBFNN might be considered costlier to apply in any computational term for a huge quantity of training data. The performance of any trained RBF network is influenced by the quantity and locality of the RBFs, their shape and the methods utilized for understanding the mapping of input-output. Most familiar existing learning techniques for RBF NNs are of the following categories (i) strategy for random choosing of the RBF centres within the given training dataset, (ii) strategy deploying an unsupervised process for choosing the RBF centres, and (iii) strategy deploying supervised process for choosing the RBF centres.

In both learning strategies, the selection of the parameter highly influences the performance of the classification of RBFNN. The parameters are the number of neurons in the hidden layer, coordinates in the centre of the hidden layer, the radius of every RBF function and the weights

among the hidden and output layer. In an NN, the hidden unit forms a class of "functions" that comprises a random "basis" for the pattern of inputs and these functions are known as RBFs. RBF network is well-known for the requirement of a very short training period; at the same time, it has the limitation that it requires the best coverage of the input space by RBFs. In such a situation where RBFNN is trained by massive repetitive information, then the network instruction will be highly loaded hence training time will also be maximized; finally, the generalization ability of the network will also be decreased. The key limitation of RBFNN is the high dimensionality that needs an optimal, however, the weighted Chimp optimization algorithm is presented to this algorithm for solving the optimization problem. Subsequently, it identifies the malicious attacks in the cloud, accordingly, the weighted Chimp optimization algorithm is mentioned in the following section.

#### 4.3.2 Weighted Chimp Optimization Algorithm

Chimp Optimization Algorithm (ChOA) is a standard Nature-Inspired Algorithms (NIA) inspired by the hunting mechanism of chimps in nature. There are four kinds of chimp driver, chaser, attacker, and barrier, so they are in charge of attracting the other chimps towards the prey (optimal solution). Consequently, with a regular equilibrium between exploration and exploitation of a search space, the best optimization problem solution will be achieved. In standard ChOA, only the first four solutions of ChOA (i.e., driver, chaser, attacker, and barrier) are utilized to update the positions of other chimps. In other words, the other chimps are attracted to these four best solutions (driver, chaser, attacker, and barrier).

The solutions related to driver, barrier, chaser, and attacker are considered the best solutions, and all of the other chimps would be guided by these four chimp groups during exploration (searching) and exploitation (hunting).

$$\overrightarrow{D_A} = \left|\vec{C}_1\overrightarrow{X_A} - \vec{M}_1\vec{X}\right|, \overrightarrow{D_B} = \left|\vec{C}_2\overrightarrow{X_B} - \vec{M}_2\vec{X}\right|,$$
$$\overrightarrow{D_C} = \left|\vec{C}_3\overrightarrow{X_C} - \vec{M}_3\vec{X}\right|, \overrightarrow{D_D} = \left|\vec{C}_4\overrightarrow{X_D} - \vec{M}_4\vec{X}\right| \quad (4)$$

$$\overrightarrow{X_1} = \overrightarrow{X_A} - \vec{A}_1\left(\overrightarrow{D_A}\right), \overrightarrow{X_2} = \overrightarrow{X_B} - \vec{A}_2\left(\overrightarrow{D_B}\right),$$
$$\overrightarrow{X_3} = \overrightarrow{X_C} - \vec{A}_3\left(\overrightarrow{D_C}\right), \overrightarrow{X_4} = \overrightarrow{X_D} - \vec{A}_4\left(\overrightarrow{D_D}\right) \quad (5)$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}+\overrightarrow{X_4}}{4} \quad (6)$$

Where $t$ denotes the current iteration. $\vec{X}_A$, $\vec{X}_B$, $\vec{X}_C$, and $\vec{X}_D$ vectors indicate the current positions of the attacker, barrier, chaser, and driver, respectively. $\vec{X}$ vector is the current position of other chimps. Also, $\vec{C}$, $\vec{M}$ and $\vec{A}$ vectors contribute greatly toward ChOA.

Although attackers have a natural ability to forecast the prey's progression route, there is no main reason that the solution of attackers is always the best because chimps sometimes leave their tasks during the process of hunting or keep their same duty during the entire process. As a result, if the position of the other chimps is updated based on attackers, they may become trapped in local optima and cannot explore new areas in search space because their solution space significantly concentrates around the attacker's solutions. Also, there are reasons for the other best solutions (driver, chaser, and barrier). To tackle this issue, our proposed WChOA offers a position-weighted relationship based on proportional weights.

Equations (4) to (6) are utilized to update the position of other chimps. What it boils down to is that the other chimps are forced to update their position based on the positions of driver, chaser, attacker, and barrier. Therefore, if the mentioned reasons in previous paragraphs are noticed, it opens the door to new approaches to update the position of other chimps. The corresponding weighting method is proposed based on the Euclidean distance of step size as follows:

$$w_1 = \frac{|\overrightarrow{X_1}|}{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}+\overrightarrow{X_4}} \quad (7)$$

$$w_2 = \frac{|\overrightarrow{X_2}|}{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}+\overrightarrow{X_4}} \quad (8)$$

$$w_3 = \frac{|\overrightarrow{X_3}|}{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}+\overrightarrow{X_4}} \quad (9)$$

$$w_4 = \frac{|\overrightarrow{X_4}|}{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}+\overrightarrow{X_4}} \quad (10)$$

Where, $w_1$, $w_2$, $w_3$ and $w_4$ are called the learning rates of other chimps from the attacker, barrier, chaser, and driver, respectively. Also, $|\cdot|$ indicates the Euclidean distance. Nonetheless, the position-weighted relationship is as follows:

$$\overrightarrow{X}(t+1) = \frac{1}{w_1+w_2+w_3+w_4} \times \frac{w_1\overrightarrow{X_1}+w_2\overrightarrow{X_2}+w_3\overrightarrow{X_3}+w_4\overrightarrow{X_4}}{4} \quad (11)$$

In WChOA, the position-weighted relationship equation (11) can be utilized instead of equation (6) in the standard ChOA. As is obvious, the main difference between equation (11) and traditional position-weighted relationship equation (6) is to apply the corresponding learning rate. As mentioned previously, since there is a possibility

that some chimps do not have any sexual motivation in the process of hunting, a probability of 50% can be considered to choose whether the position-weighted strategy of chimps will be normal (equation (11)) or not (chaotic model). Thus, the following relationship is applied:

$$\overrightarrow{X_{chimp}}(t+1) = \begin{cases} \overrightarrow{X}(t+1) & if\ \mu < 0.5 \\ Chaotic - Value & if\ \mu \geq 0.5 \end{cases} \quad (12)$$

The process of updating the position of other chimps by the first four best solutions (attacker, barrier, chaser, and driver). In other words, the final position of other chimps will randomly be a circle in the vicinity of the prey that is determined by the attacker, barrier, chaser, and driver.

*Table 1: Pseudo-Code of WChOA*

| **Algorithm:** WChOA |
|---|
| Initialize the chimp population $X_i(i = 1,2, \dots, n)$<br>Initialize $f$, $m$, $a$ and $c$<br>Calculate the position of each chimp<br>Divide chimps randomly into independent groups<br>**Until** the stopping condition is satisfied<br>Calculate the fitness of each chimp<br>$X_{Attacker}$ is the best search agent<br>$X_{Chase}$ is the second-best search agent<br>$X_{Barrier}$ is the third-best search agent<br>$X_{Driver}$ is the fourth-best search agent<br>**while** ($t$ < maximum number of iterations)<br>**for** each chimp:<br>Extract the chimp's group<br>Use its group strategy to update $f$, $m$, and $c$<br>Use $f$, $m$, and $c$ to calculate $a$ and then $d$<br>**end for**<br>**for** each search chimp<br>**if** ($\mu < 0.5$)<br>**if** ($\|a\| < 1$)<br>Update the position of the current search agent by using equation (11)<br>**else if** ($\|a\| > 1$)<br>Select a random search agent<br>end if<br>**else if** ($\mu > 0.5$)<br>Update the position of the current search by using equation (12)<br>**end**<br>**if**<br>**end for**<br>Update $f$, $m$, $a$ and $c$<br>Update $X_{Attacker}$, $X_{Driver}$, $X_{Barrier}$, $X_{Cha}$<br>$t = t + 1$<br>**end while**<br>**return** $X_{Attacker}$ |

It is noteworthy that the learning rates in the position-weighted relationship change dynamically. It means that these parameters are not constant during every iteration of WChOA. For more explanation, table 1 depicts the pseudo-code of WChOA. The proposed weighted chimp algorithm-based optimized Gaussian kernel RBFNN identifies the malicious attacks in the cloud. Consequently, it enhances the speed of convergence and avoidance of local optima where attackers, barriers, chasers, and drivers are less likely to be knowledgeable about the position of the prey.

## 5. EXPERIMENTATION AND RESULT DISCUSSION

The performance of the proposed work is evaluated using Python software. Table 2 shows the simulation configuration of the Python software. The version of the Python software is 3.8.0, it is utilized in the python Jupiter. The operating system of this software is Ubuntu, and its memory capacity is 4GB DDR3. The processor utilized capacity is an Intel Core i5 @ 3.5GHz, and the time taken for simulation is 200 seconds.

*Table 2: Table of System Configuration for Simulation*

| **Simulation System Configuration** | |
|---|---|
| Python Jupiter | Version 3.8.0 |
| Operation System | Ubuntu |
| Memory Capacity | 4GB DDR3 |
| Processor | Intel Core i5 @ 3.5GHz |
| Simulation Time | 200 seconds |

The performance evaluation is classified into two categories, functional comparison and performance analysis, and they are described in different sections. The scheme is compared with the existing Fully Homomorphic Encryption (FHE), Ciphertext Policy-Attribute based Encryption (CP-ABE), and Quasi Modified Levy Flight Distribution Reversed Sheamir Algorithm (QMLFD-RSA).

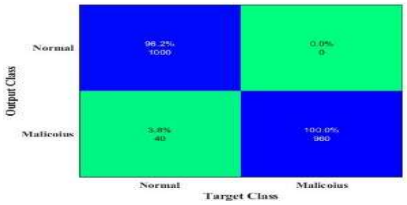


*Figure 4: Confusion Matrix of Malicious Detection of the Proposed Method*

Figure 4 demonstrates the confusion matrix of malicious detection for the proposed work. The confusion matrix shows the accuracy of the experiment at a glance. The captured confusion matrix for all in the figure in which all two types of data (normal and malicious) are combined. The numbers of the correct responses are shown in blue squares (left squares in the first rows and right squares in the second rows) while the numbers of the incorrect responses are shown in the green squares (right squares in the first row and left squares in the second rows). Accuracy determines how well the learning model functions. The system put the proposed method to the test by doing experiments and comparing the outcomes to those obtained using an established secure blockchain based on federated hybrid machine learning models. To accomplish this, we must first determine the proportion of attacks that were correctly classified as true positives, trusted nodes that were correctly classified as true negatives, false positives that were incorrectly classified as true negatives, and false negatives that were incorrectly classified as true positives. The simulation result shows that the proposed system achieved the localization accuracy of the routing attacks is 96.2% by training the extracted features as shown in Figure 5. This shows that the localization accuracy of the attack is effective compared to Kim [27] with a localization accuracy of 82.17%.

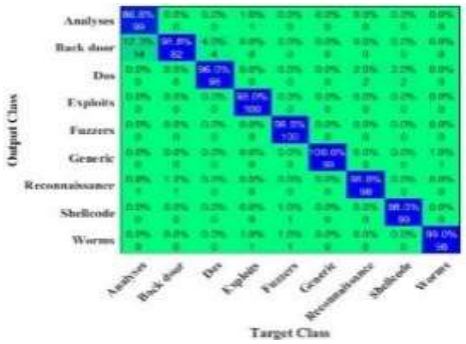


*Figure 5: Confusion Matric of Output and Target Class*

The result of the confusion matrix is displayed in figure 5. The proposed optimized method can detect several attacks such as Back door, Dos, exploits, fuzzes, generic, reconnaissance, shellcode, and worms. TDMA type detection, only 9 examples are misclassification to Normal attack. This confusion lies in the attacks and their similarity, as the information captured by the network monitoring instrument is similar. The values reflected in the dataset attributes will also have this similarity, thereby negatively affecting the metrics of the model.

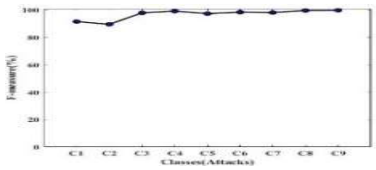


*Figure 6: F-Measure for Proposed Method*

Figure 6 shows the F-measure of all attack types using specific-attack classes. Class "0" is the normal class that includes the regular flow of network traffic. An incorrect classification of this class produces a false positive. The proposed model achieves zero false positives, and successfully detects all attacks; however, some attacks are misclassified from one attack type to another. Class 1-9 represent attacks used in this work. Incorrect analysis of attack class as normal generates a false negative. The proposed model achieves an improved detection rate for all attack classes. The false negative rate of our proposed model was zero for all attack classes. However, the proposed model achieves better recall value. Muhammad Nouman 2023 [28] determines the single value that balances both precision and recall. The F1 scores of the classifiers differ for the balanced and original datasets, the classifier achieves the highest F1 score of 97%. This shows that the proposed method has the highest F-measure value of 98% compared with existing ones.

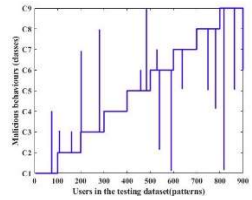


*Figure 7: Malicious Behaviour of the Testing Dataset*

Figure 7 shows the classification distribution of 900 patterns in the testing dataset, the Y-axis shows the different classes from C1 to C9 (attacks) in the testing dataset and the X-axis shows how patterns in the testing dataset are classified to

their belonging classes (attacks). The horizontal blue lines show correct classifications, for instance, in this figure the blue straight line from 0 to 100 shows the first 100 patterns in the testing dataset belonging to class 1 (Analyses attack). The turbulences (vertical blue lines) accrue in each class when there is misclassified pattern.

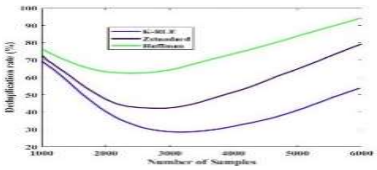

*Figure 8: Comparison Graph for Deduplication Rate*

Figure 8 portrays the deduplication rate for a different number of samples. The number of samples varied from 1000 to 6000, then the deduplication rate changed from 76% to 94%. However, the proposed method is compared with the existing Run length encoding with K-precision (KRLE), and Z-standard. Compare to these two existing methods, the proposed method produces higher values, therefore, the performance of the proposed method is efficiently improved.

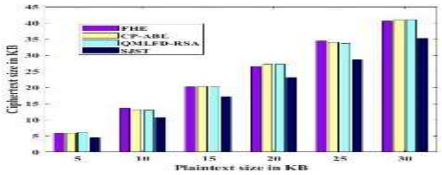


*Figure 9: Comparison of Different Models Based on Plaintext and Ciphertext*

Figure 9 shows a comparison of the plaintext size and ciphertext size based on different models. The X-axis represents the plaintext size and the Y-axis represents the cipher text size. In the proposed model, the ciphertext size is smaller than in other models, which indicates an important improvement for the proposed model. The size of the data is changed from 5 to 30 KB and ciphertext is calculated. The calculated cipher text file size for the proposed model is 35 KB for 30 KB, however, the existing FHE takes 41 KB, CP-ABE, and QMLFD-RSA takes 41.5 KB.

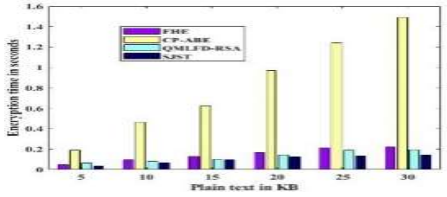


*Figure 10: Comparison Graph for Encryption Time*

The results of encryption time have been illustrated in figure 10. The efficiency of an encryption algorithm is inversely proportional to the encryption time. The lesser the encryption time of an algorithm, the higher would be its efficiency. In this work, the encryption time of the proposed SJST method is compared with the other symmetric algorithms like FHE, CP-ABE, and QMLFD-RSA. The results demonstrate that as compared to FHE, CP-ABE, and QMLFD-RSA methods, the proposed method takes less time for encrypting the same plaintext. Hence, the proposed SJST method is more efficient than these algorithms. Figure 10 shows the encryption time comparison between the proposed architecture with FHE [29], CP-ABE [30], and QMLFD-RSA [31] schemes. In all approaches, the encryption time increases with the number of attributes. This study can analyze that the proposed system requires less time to perform the encryption process than the existing literature. The proposed scheme uses a robust encryption process with a file which was used to obtain these readings of 500 lines. The results obtained on varying numbers of chunks while encrypting a file using the JS algorithm fetched the following details. The average encryption time is 140.0647124 ms. In contrast, the existing technique utilizes the symmetric session key to encrypt the plaintext, thus needing more time to share the same key for both encryption and decryption processes.

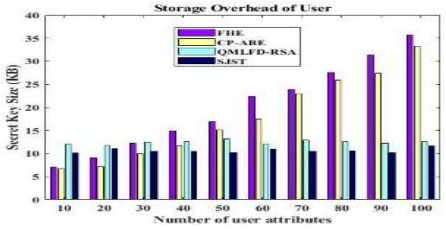


*Figure 11: Secret Key Size for Number of Users*

The overhead of the proposed method is measured at both owner and cloud server parts. Figure 11 showed that the proposed method occupies the constant memory to store the secret key. However, the secret key and encrypted file size increase linearly concerning the number of user attributes. Therefore, considering the number of secret key attributes (e.g., a value is three), the length of the secret key will increase quadratically with the number of N attributes as shown in Figure 10. The user attributes are taken from 10 to 100, respectively. Accordingly, the proposed method is compared with the existing FHE, CP-ABE, and QMLFD-RSA methods. While compare to these

existing methods, the proposed method utilizes low secrete key size, i.e., 12 KB for 100 user attributes.

From Figure 11, the study can discover that in the data updating phase, the running time cost of both the designed solution and the previous solution [32] grows with the quantity of outsourced data blocks. Furthermore, the proposed solution requires substantially less time overhead than the previous solution. For instance, when the quantity of outsourced sub-files is 5000, the time overhead of the proposed solution is nearly 1.3 milliseconds and the storage overhead is about 12KB. Nevertheless, the running time overhead of the previous solution is 55.3 milliseconds. Therefore, the study can trust that in the data updating phase, the designed solution is equipped with much higher efficiency than the previous solution.

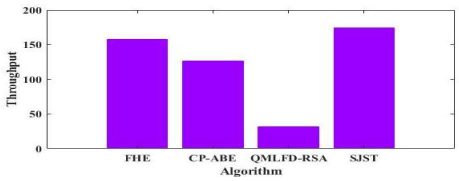


*Figure 12: Comparison Graph for Throughput*

Figure 12 portrays the comparison graph for throughput. The Performance of an algorithm can be analysed directly through its throughput. The Performance of an algorithm is directly proportional to the throughput i.e. greater the throughput of the algorithm, the higher will be the performance. Based on the recorded encryption time, the throughput for all the techniques has been calculated. The proposed method is compared with the existing FHE, CP-ABE, and QMLFD-RSA. Figure 12, clearly demonstrates that the throughput of the proposed method is higher than all other techniques. Hence, it can be concluded that the proposed method performs better than other symmetric key algorithms.

## 6. RESEARCH CONCLUSION

Cloud storage allows organizations to store data at remote sites of service providers. Although cloud storage services offer numerous benefits, they also involve new risks and challenges concerning data security and privacy aspects. Although there are several advantages to cloud computing services, especially for business owners, various challenges are posed in terms of the privacy and security of information and online services. A threat that is widely faced in the cloud environment is the on/off attack, in which entities exhibit proper behaviour for a given period to develop a positive reputation and gather trust, after which they exhibit deception. Another threat often faced by trust management services is a collusion attack, which is also known as collusive malicious feedback behaviour. In this work, a Machine learning-based algorithm is proposed for malicious behaviour detection. Initially, the Huffman algorithm is proposed for data compression, which compresses the data and decreases the data representation duplication. However, the Symmetric Jumbling-Salting Encryption Algorithm is presented for data encryption, this method encrypts the data before uploading it to the cloud. Finally, a Weighted Chimp Algorithm optimized Gaussian Kernel Radial Basis Function Neural Network based machine learning method is introduced to identify the malicious behaviour in the cloud environment, respectively. Accordingly, the proposed work is implemented using Python software.

- The result revealed that the accuracy of the experiment is shown in the confusion matrix, which produces the detected normal values are 96.2%, and the malicious data are 3.8%, respectively.
- The F-measures of the proposed work produce a higher approximate value of 98%, however, the deduplication rate of the proposed method is 94%.
- The ciphertext size of the proposed method produces 35 KB for a plaintext size of 30 KB. The encryption time of the proposed work is 0.15 seconds for plaintext in 30 KB.
- Accordingly, the proposed method outperforms the other existing methods like FHE, CP-ABE, and QMLFD-RSA.

The proposed work performs better than the other existing works. Subsequently, in future work, a robust detection method is proposed, which can be applied to the detection tasks of various malicious codes such as ransomware and Android malware. Moreover, also plan to evaluate the effect of this method on improving model robustness in other issues of cybersecurity with limited data scales. The results also reveal that when the picture distortion is minimized, the quantity of data concealed in the image rises. For diverse companies of varied sizes, objectives, and demands, the suggested methodology is more flexible, adaptable, and efficient for safeguarding cloud data. In comparison to other comparable efforts, the approach additionally assures cloud data redundancy. The qualities of the proposed model

make it suited for data exchange in the cloud, financial, and healthcare environments. The model can safeguard the confidentiality, privacy, and integrity of cloud data by employing the approaches described. As a result, since the model verifies data integrity, it can be concluded that the goal of this work, which was to improve data security and the privacy of cloud data, has been met.

## REFERENCES

[1] K.O.B.O. Agyekum, Q. Xia, E.B. Sifah, C.N.A. Cobblah, H. Xia, J. Gao, "A proxy re-encryption approach to secure data sharing in the Internet of things based on blockchain", *IEEE Systems Journal*, Vol. 16, No. 1, 2021, pp. 1685-1696.

[2] Y. Alvarez, M.A. Leguizamón-Páez, T.J. Londoño, "Risks and security solutions existing in the Internet of things (IoT) in relation to Big Data", *Ingeniería y competitividad*, Vol. 23, No. 1, 2021.

[3] P. Panahi, C. Bayılmış, U. Çavuşoğlu, S. Kaçar, "Performance evaluation of lightweight encryption algorithms for IoT-based applications", *Arabian Journal for Science and Engineering*, Vol. 46, No. 4, 2021, pp. 4015-4037

[4] S. Atiewi, A. Al-Rahayfeh, M. Almiani, S. Yussof, O. Alfandi, A. Abugabah, Y. Jararweh, "Scalable and secure big data IoT system based on multifactor authentication and lightweight cryptography", *IEEE Access*, Vol. 8, 2020, pp. 113498-113511.

[5] S. Namasudra, "Data access control in the cloud computing environment for bioinformatics", *International Journal of Applied Research in Bioinformatics (IJARB)*, Vol. 11, No. 1, 2021, pp. 40-50.

[6] S. Bhatt, T.K. Pham, M. Gupta, J. Benson, J. Park, R. Sandhu, "Attribute-based access control for AWS internet of things and secure Industries of the Future", *IEEE Access*, Vol. 9, 2021, pp. 107200-107223.

[7] M. Cui, S.S. Baek, R.G. Crespo, R. Premalatha, "Internet of things-based cloud computing platform for analyzing the physical health condition", *Technology and Health Care*, Vol. 29, No. 6, 2021, pp. 1233-1247.

[8] S. Ravikumar, D. Kavitha, "IoT based home monitoring system with secure data storage by Keccak–Chaotic sequence in cloud server", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, No. 7, 2021, pp. 7475-7487.

[9] W. Li, Y. Chai, F. Khan, S.R.U. Jan, S. Verma, V.G. Menon, X. Li, "A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system", *Mobile Networks and Applications*, Vol. 26, No. 1, 2021, pp. 234-252.

[10] T. Frikha, A. Chaari, F. Chaabane, O. Cheikhrouhou, A. Zaguia, "Healthcare and fitness data management using the IoT-based blockchain platform", *Journal of Healthcare Engineering*, Vol. 2021, pp. 2021.

[11] Z. Eghbali, M.Z. Lighvan, "A hierarchical approach for accelerating IoT data management process based on SDN principles", *Journal of Network and Computer Applications*, Vol. 181, 2021, pp. 103027.

[12] D. Unal, A. Al-Ali, F.O. Catak, M. Hammoudeh, "A secure and efficient Internet of Things cloud encryption scheme with forensics investigation compatibility based on identity-based encryption", *Future Generation Computer Systems*, Vol. 125, 2021, pp. 433-445.

[13] Y. Qiu, G. Liu, B.A. Muthu, C.B. Sivaparthipan, "Design of an energy-efficient IoT device with optimized data management in sports person health monitoring application", *Transactions on Emerging Telecommunications Technologies*, 2021, pp. e4258.

[14] T. Munirathinam, S. Ganapathy, A. Kannan, "Cloud and IoT based privacy preserved e-Healthcare system using secured storage algorithm and deep learning", *Journal of Intelligent & Fuzzy Systems*, Vol. 39, No. 3, 2020, pp. 3011-3023.

[15] V. Prabhakaran, A. Kulandasamy, "Hybrid semantic deep learning architecture and optimal advanced encryption standard key management scheme for secure cloud storage and intrusion detection", *Neural Computing and Applications*, Vol. 33, No. 21, 2021, pp. 14459-14479.

[16] L. Haghnegahdar, S.S. Joshi, N.B. Dahotre, "From IoT-based cloud manufacturing approach to intelligent additive manufacturing: Industrial Internet of Things—An overview", *The International Journal of Advanced Manufacturing Technology*, 2022, pp. 1-18.

[17] F. Sajid, M.A. Hassan, A.A. Khan, M. Rizwan, N. Kryvinska, K. Vincent, I.U. Khan, "Secure and Efficient Data Storage Operations by

Using Intelligent Classification Techniques and RSA Algorithm in IoT-Based Cloud Computing", *Scientific Programming*, Vol. 2022, 2022.

[18] S. Mishra, A.K. Tyagi, "The role of machine learning techniques in the internet of things-based cloud applications", *In Artificial Intelligence-based Internet of Things Systems* (pp. 105-135). Springer, Cham. 2022.

[19] N. Almurisi, S. Tadisetty, "Cloud-based virtualization environment for IoT-based WSN: solutions, approaches and challenges", *Journal of Ambient Intelligence and Humanized Computing*, 2022, pp. 1-23.

[20] G.G. Deverajan, V. Muthukumaran, C.H. Hsu, M. Karuppiah, Y.C. Chung, Y.H. Chen, "Public key encryption with equality test for Industrial Internet of Things system in cloud computing", *Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 4, 2022, pp. e4202.

[21] A. Ali, M.A. Almaiah, F. Hajjej, M.F. Pasha, O.H. Fang, R. Khan, J. Teo, M. Zakarya, "An Industrial IoT-Based Blockchain-Enabled Secure Searchable Encryption Approach for Healthcare Systems Using Neural Network", *Sensors*, Vol. 22, No. 2, 2022, pp. 572.

[22] D.B. Babu, M.K.J. Krishna, M.A.J. Sree, M.B.V. Prasanti, M.D. Srivalli, M.K. Sahithi, "Cloud-based data storage and sharing with dual access Control", *Specialusis Ugdymas*, Vol. 1, No. 43, 2022, pp. 6209-6215.

[23] Z. Lu, Y. Guo, J. Li, W. Jia, L. Lv, J. Shen, "Novel Searchable Attribute-Based Encryption for the Internet of Things", *Wireless Communications and Mobile Computing*, Vol. 2022, 2022.

[24] S. Das, S. Namasudra, "A Novel Hybrid Encryption Method to Secure Healthcare Data in IoT-enabled Healthcare Infrastructure", *Computers and Electrical Engineering*, Vol. 101, 2022, pp. 107991.

[25] K.S. Patil, I. Mandal, C. Rangaswamy, "Hybrid and Adaptive Cryptographic-based secure authentication approach in IoT based applications using hybrid encryption", *Pervasive and Mobile Computing*, Vol. 82, 2022, pp. 101552.

[26] J. Mamvong, G. Goteng, Y. Gao, "Low-cost client-side encryption and secure Internet of things (IoT) provisioning", *Frontiers of Computer Science*, Vol. 16, No. 6, 2022, pp. 1-3.

[27] T. H. Kim, R. Goyat, M. K. Rai et al., "A novel trust evaluation process for secure localization using a decentralized blockchain in wireless sensor networks," *IEEE Access*, Vol. 7, 2019, pp. 184133–184144.

[28] M. Nouman, U. Qasim, H. Nasir, A. Almasoud, M. Imran, and N. Javaid, "Malicious Node Detection using Machine Learning and Distributed Data Storage using Blockchain in WSNs", *IEEE Access*, 2023.

[29] R. Hamza et al., "Towards secure Big Data analysis via fully homomorphic encryption algorithms", *Entropy (Basel)*, Vol. 24, No. 4, 2022 p. 519.

[30] P. Sharma, R. Jindal, and M.D. Borah, "Blockchain-based cloud storage system with CP-ABE-based access control and revocation process", *The Journal of Supercomputing*, 2022, pp.1-29.

[31] P. Kaliyamoorthy, and A.C. Ramalingam, "QMLFD based RSA cryptosystem for enhancing data security in public cloud storage systems", *Wireless Personal Communications*, Vol. 122, No. 1, 2022, pp.755-782.

[32] H. Yu, Z. Yang, M. Waqas et al., "Efficient dynamic multi-replica auditing for the cloud with geographic location", *Future Generation Computer Systems*, Vol. 125, 2021, pp. 285–298.