IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 11, 2020, accepted June 22, 2020, date of publication June 24, 2020, date of current version July 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004711

A Secure Industrial Internet of Things (IIoT) Framework for Resource Management in Smart Manufacturing

KHALED ALI ABUHASEL[1] AND MOHAMMAD AYOUB KHAN[2], (Senior Member, IEEE)
[1]Mechanical Engineering Department, College of Engineering, University of Bisha, Bisha 61421, Saudi Arabia
[2]College of Computing and Information Technology, University of Bisha, Bisha 67714, Saudi Arabia

Corresponding author: Mohammad Ayoub Khan (ayoub.khan@ieee.org)

ABSTRACT The Industrial Internet of Things (IIoT) can transform an existing isolated industrial system to a connected network. The IIoT and the related wireless connectivity requirements for industrial sensors are very significant. The deployed sensors in IIoT monitors the conditions of the industrial devices and machines. Therefore, reliability and security become the most important concerns in IIoT. This introduces many familiar and ever-increasing risks associated with the industrial system. The IIoT devices can be vulnerable to vast array of viruses, threats, and attacks. Therefore, an efficient protection strategy is required to ensure that the millions of IIoT devices are safe from these risks. However, resource constraint IIoT devices have not been designed to have effective security features. Due to this, in recent years, cloud, fog, and edge-based IIoT has received great attention in the research community. The computationally intensive tasks such as security, data analytics, decision making, and reporting are performed at the cloud or fog using a powerful computing infrastructure. The data security of the IIoT device has been provided by employing improved Rivest-Shamir-Adelman (RSA) and hash signatures. The proposed RSA algorithm has a four-prime number of 512- bits. The device authentication is performed by employing a hash signature. For long network life, an efficient clustering technique for the sensor devices which is based on node degree(N), distance from the cluster(D), residual energy(R), and fitness (NDRF) has been proposed. The fitness of the sensor nodes is computed using the Salp swarm algorithm (SSA). In order to reduce the latency and communication overhead for IIoT devices a resource scheduling using SoftMax deep neural network (DNN) is proposed. All the requests coming from the cluster head are classified using SoftMax-DNN for best resource scheduling on the basis of storage, computing, and bandwidth requirements. The proposed framework produces superior results, especially in terms of energy consumption, latency, and strength of security.

INDEX TERMS IoT, industrial IoT, wireless sensor networks (WSN), SoftMax, RSA, SHA-512.

I. INTRODUCTION

Technologies like the Internet of things (IoT), wireless sensor network (WSN), Cloud computing, edge computing, cyber-physical system, and fog computing, brings new business model and market. These technologies have several advantages in increasing automation, production, performance, reliability, and safety in industrial sectors. There is a growing possibility of adding more and more efficient, complex IP-based devices which use advanced sensors, wireless network, and microprocessors. In many sectors, the IoT has drawn the attention that includes retail, logistics, healthcare, supply chain, manufacturing, and pharmaceuticals. On the other hand, progress in wireless communications and sensor network technologies involves more and more connected objects in the Industrial Internet of things (IIoT) [1], [2]. Industry 5.0 standards promote the entire industry with regard to productivity, efficiency, promoting heterogeneous data, increased production, automation, and information integration [3]–[6]. In addition, the number of devices connected between them will drastically increase, and devices will communicate constantly with local cloud services in order to function smartly and flexibly. In particular, the IIoT system has a frequency, exchange of information and an independent financial transaction, which have always been very stagnant and highly isolated [7], [8]. Smart manufacturing is facilitated

The associate editor coordinating the review of this manuscript and approving it for publication was Chunsheng Zhu.

by IoT devices, smart sensors, robotics, actuators, communication devices, machine learning, technologies, and data analytics to automate and enhance the productivity. This brings numerous challenges to industrial network systems [9], such as the protection of devices, privacy and architectural flexibility. Cloud computing provides infrastructure centralization and resource sharing. However, still unable to meet all the demands of deployment for distributed IoT particularly given the drastic increase in time and power restricted in the sensing devices.

In order to extend the cloud's ability to store, process, and network at the edge of the network, this evolving computing paradigm is called fog computing [11]. The purpose of fog computing is to reduce time delays when the system is in use in which the traffic that is complex in processing is moved to the cloud data center [12]. The fog computing includes the virtualized layer which is located between end-users and the cloud data centers context [13]. There are numerous advantages of the cloud and fog-based model which reduces latency, reduced networking traffic, and improved energy efficiency. One of the important advantages is the allocation of resources and the scheduling of tasks. The fog computing matches the most desirable resources to the tasks of the applications. Due to its involvement in the assignment of the task [14], the technology can be controlled on the best way to match resources for the tasks of the application, which does not go beyond the minimum set times aiming at meeting quality of service (QoS) needs of the IIoT devices [15]. This improves the performance of the fog computing, and helps to implement the planning of load balancing.

Due to the varying range of requests submitted by IIoT devices, fog computing assigns the resources in such a way that the device needs are met. It seeks to identify the best appropriate resources for IIoT devices such as reducing process delays and improved use of resources in order to reach optimal planning goals [16]. Furthermore, this requires the design of secure and robust security in IIoT devices [17]. It is therefore of the greatest practical importance to obtain efficient use of resources and performance of a high level from the computing environment of the fog system. Therefore, this work presents the SoftMax based deep neural network and improved RSA algorithms for effective resource schedules and secure data transmission for smart manufacturing using the IIoT framework. The contributions of this work are as follows:

- First, the data security of the IIoT device has been provided by employing improved RSA and hash signatures.
- The second contribution is the clustering technique for the sensor devices using the NDRF-SSA technique which is based on sensor node degree, distance from cluster head, residual energy, and fitness.
- The third contribution is to devise classification using SoftMax-DNN for best resource scheduling on the basis of storage, computing, and bandwidth requirement.
- The results from the simulation confirms that making use of the given algorithm is more desirable than the existing approaches. The proposed algorithm has low latency, low energy consumption, efficient resource allocation, and strong security.

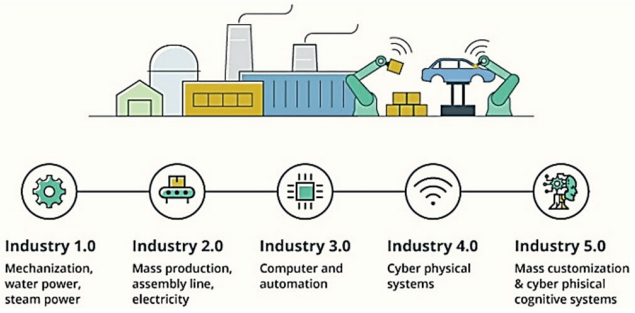

FIGURE 1. Evolution of industry ages [2].

The article is structured as follows: Section II presents existing works related to the proposed method. A detailed discussion of the proposed work is presented in section III. In section IV, the results of the simulation are explored and compared. The conclusion is contained in section V.

II. RELATED WORK

Many studies have been carried out to provide efficient resource scheduling and security in the industrial network. For instance, Shivi *et al.* [12] have focused on the detection of duplicate tasks capable of reducing the capacity of storage and the latency of the cloud server. To enhance the security of the data, the ECC based HM algorithm has been used to encrypt data. Further, Li *et al.* [15] present a study on resource scheduling in fog environment. The authors have proposed the FCAP algorithm for clustering the resource in the fog to match the resource. The best characteristics of the particle swarm optimization coupled with those of the fuzzy clustering are used to ensure that resource allocation is optimal. The simulation result demonstrates an efficient allocation of resources. Bu *et al.* [19] proposed and simulated a secure and reliable model for enabling information sharing in IoT networks. Authors have used threshold-based secret sharing that divides into parts to be stored by the IoT devices on the network to retrieve the information by the devices in a collaborative fashion. In another work, Bhatia *et al.* [20] have proposed a quantumized scheme for scheduling tasks in the fog computing environment. A node-specific metric for measuring the level of computing of the fog system was specifically denoted as the node computing index. The authors have performed a comparative analysis with existing models and found to be superior [20].

Alzubi *et al.* [21] have devised a scheme to provide security for IIoT data transfer using cloud services with a Hashed Needham Schroeder (HNS) cost-optimized deep machine Learning (CODML) technique that shows the need to deliver IIoT security.

A public key using HNS is computed that provides access to the device. This way they improve the execution time as

TABLE 1. Comparative analysis of related work.

S. No.	Authors	Methodology	IoT	Security	Task Allocation	Fog Computing	Cloud Computing	Optimization
1.	Shivi Sharma [12]	Task allocation and secure deduplication	IIoT	Yes	Yes	Yes	Yes	Whale Optimization algorithm
2.	Guangshun Li [15]	Methods of Resource Scheduling	No	No	Yes	Yes	No	Fuzzy-PSO
3.	Lake Bu et al [19]	A Secure and Robust Scheme	IoT	Yes	No	No	No	No
4.	M. Bhatia [20]	QCI-prediction for task allocation	No	No	Yes	Yes	No	QCI-Neural Network
5.	J. A. Alzubi [21]	Hashed Needham Schroeder Industrial IoT	IIoT	Yes	No	No	Yes	No
6.	Pegah Gazori [22]	Fog-based IoT based scheduling	IoT	Yes	Yes	Yes	No	Double Deep Q-Learning
7.	Huaiying Sun [23]	Energy and time efficient task offloading/allocation	IoT	No	Yes	Yes	No	ETCORA algorithm
8.	Wei Wang [24]	Data Scheduling and Resource Optimization	IIoT	No	Yes	Yes	No	No

only authenticated devices in the cloud were able to transmit the message on a securer channel. The findings of the simulation have shown that, compared to existing techniques, the HNS-CODML approach produces superior results by minimizing the overhead communication. In another wok, Gazori *et al.* [22] have presented the task scheduling mechanism for IoT application to reduce long time operation delays and measurement costs within the resources and time limit. The authors have devised a double deep Q-learning for task scheduling techniques. In terms of a service delay, the measurement costs and, energy consumption as well as task completion, the evaluation results showed that the algorithm has surpassed other simple algorithms and also manages single-point failure along with challenges of load balancing. Further, Sun *et al.* [23] have devised a fog-cloud enabled IoT architecture that has used the best features of fog and cloud. An ETCORA algorithm has been applied to improve energy consumption and complete applications [23]. The authors have shown the simulation results that are able to minimize the energy expenditure and response time. However, this paper has no discussion is present on the security of the data. In another work, Wang *et al.* [24] have combined the decentralized resources of fog nodes into a cluster that has the processing capacity to handle a complex task allocation. Thereafter, authors have applied a multi-channel data planning strategy that was developed to minimize real-time processing delays and enhance system stability. Simulation results demonstrate that the optimal data scheduling strategy for performance gains could be accomplished according to various scenarios. A comparative analysis of various related work has been shown in Table 1.

III. PROPOSED FRAMEWORK

In this work, a resource scheduling, clustering of sensor nodes and, secure transmission of IIoT data are proposed using SoftMax and improved RSA algorithm. The proposed resource scheduling and security system for IIoT data contain four layers. The layers are described as: sensing devices for physical IIoT, gateway, fog, and cloud. The proposed framework is depicted in figure 2.

TABLE 2. Layer functions.

Layers	Function
Could Layer	Device authentication, Data storage, Data analytics, Decision making
Fog Layer	Pre-processing (Data standardization and normalization) Task classifier and scheduling, (SMDNN), Hashing, Data encryption (Improved RSA), Resource scheduling
Gateway Layer	Connectivity aggregation (standards, protocols, bridges)
IIoT Sensing Layer	Connection, Device registration, Login, Clustering (NDRF-SSA)

The proposed function for each layer is shown in Table 2. The detailed description of each layer is as follow:

IIoT sensing device layer: This has deployed devices and sensors to sense the data that is sent to the cloud layer via the gateway and fog layer. In this layer, to avoid unauthorized access to the IIoT data, first, the IIoT device registered and login operation is performed. Upon successful authentication by the cloud layer, the device is get connected to the cloud server. The clusters are formed where cluster head is selected based on the node degree(N), the distance between nodes(D), residual energy(R), and their fitness(F).

The fitness is tested using the Salp swarm algorithm (SSA), this proposed algorithm is named as NDRF-SSA. The cluster head aggregates data from nodes and transmits it to the fog layer via the gateway layer.

Gateway layer: The gateway layer is responsible for connectivity aggregation that enables the communication between various heterogeneous devices and sensors. This layer also provides interoperability among different standards, protocols, and systems.

Fog services layer: Fog is distributed, unlike cloud which is centralized. In the fog layer, the task in the form of

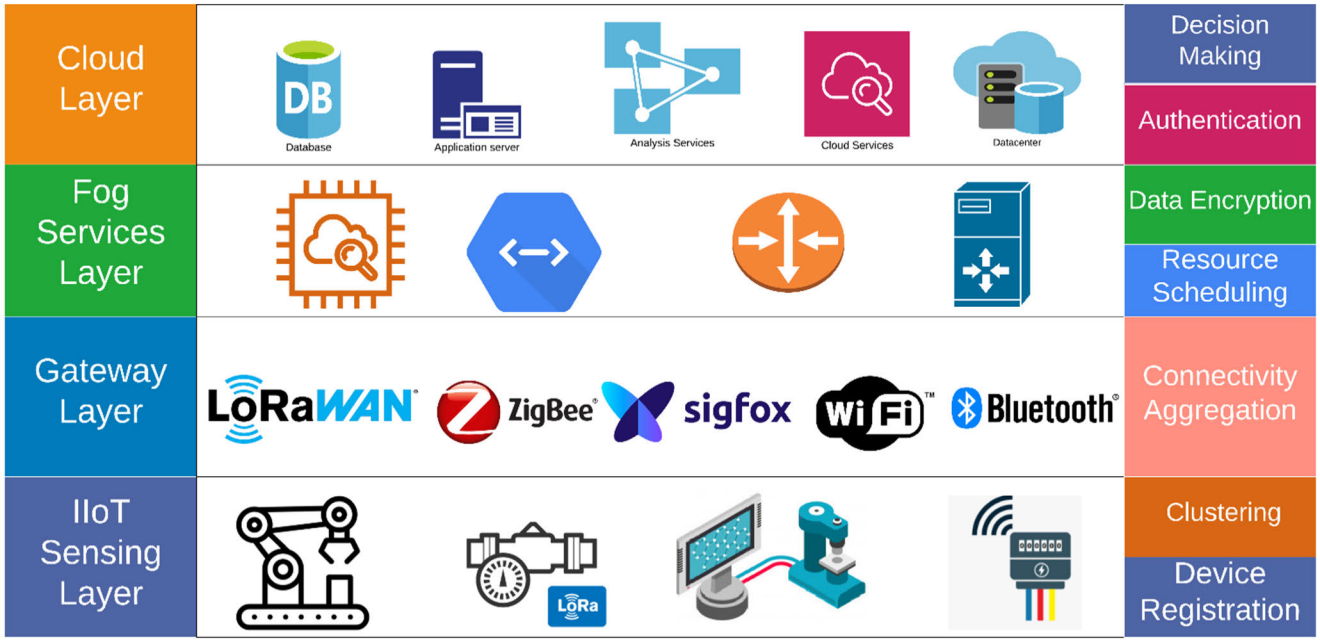

FIGURE 2. Proposed secure industrial IoT framework.

requests from the cluster heads are continuously received. These received tasks are allocated to a cloud server using the machine learning algorithm called SoftMax deep neural network. That means the received tasks are first classified into three classes: memory resources, bandwidth resources, and storage resources. Then, these resources are securely allocated to the cloud server (cloud layer) using the resource scheduling algorithm which is discussed further in later section. Here for avoiding the data deduplication and improving the security, the proposed method uses the SHA-512 algorithm and improved RSA algorithm.

Cloud layer: This layer is end-user layer which responsible for device authentication, data storage, data analytics, and decision making. The user and decision makers can interact through this layer.

A. DEVICE AUTHENTICATION

Device authentication is an important task in the IIoT environment as the deployed devices read the sensor values which is forwarded for processing to the cloud server through the fog computing layer. In order to avoid the unauthorized access of IIoT device data, device authentication is proposed using three steps: registration, login, and verification. Each of the steps of the authentication is performed using affine cipher-based SHA 512 algorithm [18], [25], [26], which is elucidated as below:

Step 1: The registration phase is performed using device information such as unique identification (*Dev_Id*), device password (*Dev_Pw*), device type (*Dev_Type*), device MAC address (*Dev_Mac*), and device location (*Dev_loc*). Here, first, the affine cipher is used to create the registration code. The registration code is generated by combining the device Id and the device password then encryption is performed using equations (1) and (2) [26].

$$E(x) = (ax + b) \ mod \ m \tag{1}$$

$$D(x) = a^{-1}(x - b) \ mod \ m$$
$$s.t, \ 1 = a.a^{-1} \ mod \ m \tag{2}$$

where:

$$E(x) - encryption\ function$$
$$D(x) - decryption\ function$$
$$a, b - coprime/key\ numbers$$
$$mod - modulus$$

Then, the SHA-512 algorithm creates the hash value for that registration code. Lastly, this converted hash value $\boldsymbol{H_c}$ is stored in the server at the cloud layer.

$$H_c = SHA512(E(Dev\_Id \parallel Dev\_Pw) \tag{3}$$

Step 2: Subsequent to registration, for attaining sensor values, the device shall connect to the server using a login mechanism. During login, the device Id along with the password for device authentication will be transmitted to the cloud server. Thereafter, the registration code is again generated by combining the device Id and password. Then, for this registration code, the hash value is again generated using the SHA-512 algorithm.

Step 3: The verification step is performed at the cloud server which checks whether the hashed value of the transmitted device value is similar to the hashed value generated during registration. If yes, then the IIoT sensor device is turned on, and the cloud server attains the data concurrently,

otherwise, the verification phase is redirected to the login phase.

B. NDRF-SSA CLUSTERING METHOD

The NDRF-SSA steps for the creation and selection of the cluster heads are explained as follows:

Step 1: The initial population starts with k number of swarms as follows [27]:

$$S_i = \{S_1, S_2, \ldots .S_k\}, \quad i = 1, 2, \ldots .k \tag{4}$$

The S_i denotes the population k swarms, S_1, S_2, and S_k denotes the first, second, and k^h swarm in the population.

Step 2: After initialization, initialize the global optimum using a following D-dimensional vector.

$$G_j = \{G_1, G_2, \ldots .G_N \ldots, G_D\} \tag{5}$$

Step 3: Now, compute the fitness value (FV) of all salps and thereafter choose the salp which has the best FV to assign as a global optimal position of the population. The FV for IIoT devices is calculated based on the residual energy, degree, and distance. The below equation (6) specifies the fitness function for performing the CH selection.

$$f_t = \left\{max\left(n_R\right), max\left(n_{Deg}\right), min\left(n_d\right)\right\} \tag{6}$$

Any device in which maximum n_R, n_{Deg} and minimum n_d is chosen as cluster head. The fitness function f_t, is computed using equation (7) - (11).

Node Degree: The IIoT device P_i shall join a new cluster head C_j that has higher node degree n_{Deg} than C_k in the vicinity as shown in equation (7).

$$ft\left(P_i, C_j\right) \propto \frac{1}{n_{Deg}\left(C_j\right)} \tag{7}$$

Residual Energy: The IIoT device P_i shall join a new cluster head C_j that has higher residual energy n_R than other C_k in the vicinity as shown in equation (8).

$$ft\left(P_i, C_j\right) \propto \frac{1}{n_R\left(C_j\right)} \tag{8}$$

Distance: The IIoT devices that has minimum distance from any cluster head C_j will be selected sine it will consume less energy to transmit the data to cluster head. The FV can be expressed in equation (9):

$$ft\left(P_i, C_j\right) \propto \frac{1}{n_d\left(P_i, C_j\right)} \tag{9}$$

The equations (7) - (9) can be joined together to derive final fitness values as shown in equation (10) and (11).

$$ft\left(P_i, C_j\right) \propto \frac{n_R\left(C_j\right)}{n_d\left(P_i, C_j\right) \times n_{Deg}\left(C_j\right)} \tag{10}$$

$$ft\left(P_i, C_j\right) = \lambda \times \frac{n_R\left(C_j\right)}{n_d\left(P_i, C_j\right) \times n_{Deg}\left(C_j\right)} \tag{11}$$

The λ is a constant symbol. Equation (11) computes the fitness values of the cluster that is based on the parameters and sends requests to cluster head to the network which has the maximum weight.

Step 4: Now update the leader and follower position equation (12) and (13) respectively [27]:

$$S_1^j = \begin{cases} G_j + k_1 * \left(\left(u_{bj} - l_{bj}\right) * k_2 + l_{bj}\right), & k_3 \geq 0.5 \\ G_j - k_1 * \left(\left(u_{bj} - l_{bj}\right) * k_2 + l_{bj}\right), & k_3 < 0.5 \end{cases} \tag{12}$$

$$S_{i+1}^j = \frac{1}{2}\left(S_{i+1}^j + S_{i-1}^j\right) \tag{13}$$

$$k_1 = 2e^{\left(4I/P\right)} \tag{14}$$

where:

S_1^j − position of the leader
G_j − food source position
u_{bj}, l_{bj} − upper and lower bound
k_1, k_2, k − random numbers
I − current iteration
p − max. iteration

Step 5: Now, the global optimum position is updated. The *ft* of each individual is computed and compared with the *ft* of the global optimum. The individual position will replace the optimal position if it is better.

Step 6: Evaluate if the computed results satisfy the final goal, under normal circumstances, on the question of optimization assignment such as the maximum number of iterations, etc. If the goal is achieved then stop otherwise go back to step 4.

C. RESOURCE SCHEDULING AND ENCRYPTION

1) TASK SCHEDULING

After cluster formation, data in the cluster nodes is aggregated by the cluster head. After that, it is taken to the fog layer. The fog layer receives a massive amount of data or requests from the cluster heads. Therefore, it is important to schedule the data processing for transmission to the next layer. The task manager is to arrange incoming requests in a certain way to make good use of the resources available. Here, to perform task scheduling, first, when the cluster head submits the tasks to the fog layer, preprocessing operations, such as data standardization and data normalization can be carried out to improve the classification accuracy, and thereafter the preprocessed tasks are classified into sub-tasks. The scheduler of tasks, contained in the fog environment must receive the task. The task scheduler gathers scheduling data from the cluster heads, the monitors, and the cloud. These tasks are then assigned to the appropriate node of the fog. Here, to avoid storage space and securely scheduling the resources to cloud virtual machines, the classified resources are checked for data deduplication using SHA-512. Thereafter, the encryption using the improved RSA algorithm is performed and then scheduled to the cloud server virtual machines.

Let's the number of tasks as n and the number of resources as m. The collection of tasks can be represented as

$S = \{s_1, s_2, s_3, \ldots .s_n\}$, the fog resources set is are represented as $H = \{h_1, h_2, h_3, \ldots .h_n\}$. The features of task i is represented following the 1-dimensional array by equation (15):

$$S_i = \{s_{id}, s_{len}, s_{com}, s_b, s_s, s_d\} \tag{15}$$

The task identification is represented by s_{id}, the task length is represented by s_{len}, computing requirements by s_{com}, s_b is the bandwidth requirement, the storage requirement is denoted by s_s and s_d are the task data. Fog computing is achieved by virtualization technology to abstract physical resources from virtual resources. If the number of resources in the ith set is m of fog resource i^{th} resource can be denoted by H_j as in equation (16):

$$H_j = \{h_{id}, h_{com}, h_b, h_s\} \tag{16}$$

Here, h_{id} is the resource number; and h_{com}, h_b and, h_s represent resource identification, resource computing power, resource bandwidth, and resource capacity respectively. The preprocessing, classification, encryption, and scheduling of tasks are performed which is explained in the next sections.

Data Standardization and Normalization: In fog computing, the impact on classification results will be unevenly affected by the different measurements of fog resource characteristics when raw data are processed directly. Thus, in order to resolve the negative effects that are as a result of this situation, the resource matrix data is standardized by the standard deviation (SD). The set of fog resources $A = \{a_1, a_2, \ldots a_m\}$ represents indicates the m nodes of the fog resource with n elements shown as the matrix in equation (17) [15]:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots\ldots & a_{1n} \\ a_{21} & a_{22} & \ldots\ldots & a_{2n} \\ . & . & & \\ . & . & & \\ a_{m1} & a_{m2} & \ldots\ldots & a_{mn} \end{bmatrix} \tag{17}$$

The a_{ij} represents the j^{th} attribute of resource r_{ij}.

$$a'_{ij} = \frac{a_{ij} - \bar{a}_{ij}}{b_j} \tag{18}$$

$$\bar{a}_{ij} = \frac{1}{m} \sum_{j=1}^{n} a_{ij} \tag{19}$$

$$b_j = \sqrt{\frac{1}{m} \sum_{i=1}^{n} \left(a_{ij} - \bar{a}_j\right)^2} \tag{20}$$

$$a''_{ij} = \frac{a'_{ij} - min\left(a'_{ij}\right)}{max\left(a'_{ij}\right) - min\left(a'_{ij}\right)} \tag{21}$$

where:

$\bar{a}_{ij}$ – avg. value of resource

b_j – standard deviation

a''_{ij} – normalized value

$min\left(a'_{ij}\right)$ – min. value of $\left(a'_{1j}, a'_{2j}, \ldots\ldots a'_{mj}\right)$

$max\left(a'_{ij}\right)$ – max. value of $\left(a'_{1j}, a'_{2j}, \ldots\ldots a'_{mj}\right)$.

The processed data should be is normal. Therefore, it has a mean of 0 and SD 1. Thus, the data in the matrix is normalized between 0 and 1.

2) SOFTMAX CLASSIFIER

After preprocessing, the preprocessed tasks are classified using the SoftMax function [30] based on a deep neural network (DNN). In general, DNN is a complex neural network of more than two layers. The DNN contains input, output, and hidden layers. When an input in neurons increases, followed by an increase in the hidden layer, the resultant neural network is complicated. Furthermore, the time of execution is increased while the accuracy is reduced. Time is reduced since the DDN is trapped in the local minima. So, the proposed method uses the SoftMax layer with a rectified activation function in the output layer to ensure that the speed of computation and prediction remains high. The significant advantage of SoftMax is the range of output probabilities. The range between 0 and 1 is equal to the total of all probabilities. So, the proposed resource classification model is named SoftMax function based DNN(SoftMax-DNN) and the architecture is depicted in figure 3.

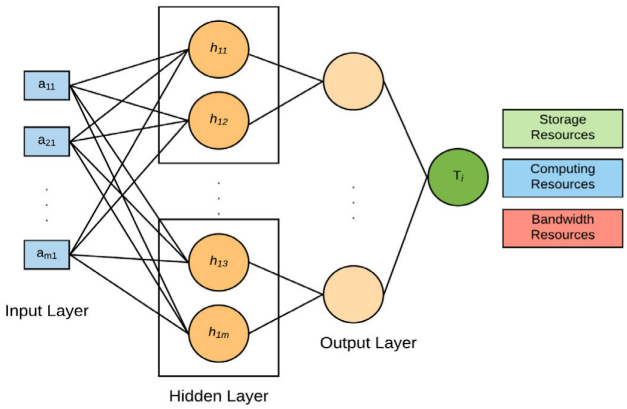

FIGURE 3. Structure of SoftMax.

The algorithm of SoftMax-DNN is given as follows:

Step 1: First in the input layer, the received preprocessed tasks of cluster heads are given as input.

Step 2: Generate the weight values for each input data that is given in the input layer, and thereafter, assign to all the neurons in the hidden and output layer an input data. Finally, ensure that weight in all neurons of the input layer is maintained.

Step 3: Compute the output of the hidden layers (Hidden layers 1, 2, and 3) using equations (22), (23), and (24). The proposed method uses three hidden layers for resource classification.

$$h_{1i} = b_{1i} + \sum_{i=1}^{N} m_i w_{1i} \tag{22}$$

$$h_{2i} = b_{2i} + \sum_{i=1}^{N} h_{1i} w_{2i} \tag{23}$$

$$h_{3i} = b_{3i} + \sum_{i=1}^{N} h_{2i} w_{3i} \tag{24}$$

where h_{1i}, $h_{2i,}$ and h_{3i} specifies the outcome of the 1st, 2nd, and 3rd layers, b_{1i}, $b_{2i,}$ and b_{3i} denotes the bias values of 1st, 2nd, and 3rd layers while as w_{1i}, w_{2i}, and w_{3i} denotes the weight values of 1st, 2nd, and 3rd layers and m_i refers to the input data values from the clustering unit.

Step 4: To find the final output unit, here, the SoftMax layer is used as an output layer, which uses the rectifier's activation function to compute the weight value of the final hidden layer. The rectifier allows the network to converge very quickly. The SoftMax activation function for DNN is expressed as follows:

$$T_i = \frac{h_{3i} + B_v}{R_f'(x)} \tag{25}$$

where T_i denotes the final SoftMax output $R_f'(x)$ and B_v denotes the weight and bias values of the final hidden layer. Here, the weight value of the final hidden layer is given based on the rectifier's activation function, which is expressed as follows,

$$R_f(x) = max(0, x) * w_i' \tag{26}$$

Here, w_i denotes the weight value and $R(f) \rightarrow x$ for positive value of x and $R(f) \rightarrow 0$ for negative value of x. Then, equation (26) is applied to the SoftMax function, which classifies the resources effectively as storage, memory, and computing resources. After, task scheduling, to detect the duplicate task a SHA-512 algorithm is employed. Using SHA-512, the cloud server produces the hash code for incoming device requests. It then checks whether or not the hash value is present in a cloud server's hash table. If so, then the cloud server to the file location route stored otherwise the encryption for storing the data is carried out [26].

3) IMPROVED RSA ENCRYPTION

In this phase, encryption is performed to provide an additional level of difficulty for the intruder even if it peeps out of the authentication then it can't decrypt the data. Here, for encryption, the proposed method uses the enhanced version of the RSA cryptographic algorithm. In RSA, two prime numbers are considered initially [26]. In the key generation process, these two prime numbers are multiplied. Therefore, if the intruder can find these factors using the various types of attacks, the security level will decrease. So, here, the RSA algorithm is enhanced with four random prime numbers to increase the security level of the system. RSA algorithm with four-prime numbers are used to increase attack time. Therefore, improved RSA will provide strong results by increasing the security level with a small key size. The improved RSA consist of three phases. They are key generation, encryption, and decryption. The step by step explanation of improved RSA is explained as follows:

Step 1: choose 512-bit long numbers as p, q, r, s
where $p \neq q \neq r \neq s$.
Step 2: compute $n = p \times q \times r \times s$
Step 3: $\beta = (p-1)(q-1)(r-1)(s-1)$
Step 4: $gcd(m, \beta) = 1$, where $1 < m < \beta$, m $\in (-n, 0, n)$
Step 5: compute $p = m \times r \times s$
Step 6: $w \equiv p^{-1} (mod\ \beta)$

$$p \times w \equiv 1 (mod\ \beta), \quad 1 < w < \beta$$

where:
(p, n) - public key
(w, n)- private key
w, p, q, r, s, β- secret numbers
Step 7: encrypt data using $E_c \equiv D^p (mod\ v)$
Step 8: decrypt the encrypted using $D \equiv E_c^w (mod\ v)$
We can observe that m is multiplied with the third and fourth prime number r and s to make it difficult for the attacker to find the value of m even if public key pair (p, n) is known.

4) RESOURCE ALLOCATION

Resource scheduling is an important aspect of any system. Resources in the class are matched with the appropriate resource category and the requirement of the devices. The simple weight matching is used to complete the resource scheduling as follows [29]:

$$G = \frac{\sum \|req_i - res_i\|}{\sum w_i} \tag{27}$$

where:
req_i – attributes of request
res_i – attributes of resources
w_i – attributes weight
The resource requirements for various devices are different. Therefore, they can be categorized as processing, storage requirements, and bandwidth for various task preferences. The above formula calculates the attribute and the resource attribute required by the device in accordance with the highest score obtained as a result of the resource scheduling.

IV. SIMULATION RESULTS AND DISCUSSIONS

In this paper, a resource scheduling and secure data transmission of IIoT data are proposed using SoftMax-DNN and improved RSA techniques. To validate and evaluate the model, the simulation has been done using JAVA and NS3 platforms. The simulation results of the proposed and existing techniques are analyzed for performance metrics. The metrics parameters that are used for the analysis are latency, network lifetime, energy consumption, and security strength.

A. ANALYSIS OF NETWORK PERFORMANCE

A comparison between the performance of the proposed algorithm and the existing techniques [12] in terms of the performance metrics, say average latency, and energy consumption is shown in Table 3.

Fig. 4 and 5 illustrate the graphs of the proposed and existing technique against the latency and energy. Metrics are measured by varying the IIoT devices from 0 to 20 devices.

In Fig. 5, the comparison of the techniques for energy consumption is shown. For 10 IIoT devices, the proposed algorithm gives the energy consumption of 0.15J whereas the existing FATA [12] attains the energy consumption of 0.3,

TABLE 3. Network latency & energy consumption.

# IIoT devices	Latency (ms)		Energy Consumption (J)	
	FATA [12]	Proposed	FATA [12]	Proposed
2	.5	.5	0.13	0.05
4	1	.8	0.15	0.06
6	1.2	.9	0.19	0.08
8	2.2	1	0.25	0.1
10	2.4	1.2	0.3	0.15
12	2.5	1.5	0.35	0.25
14	2.8	1.6	0.40	0.30
16	4.2	1.9	0.49	0.35
18	5.9	2	0.63	0.40
20	7	2.1	0.91	0.45

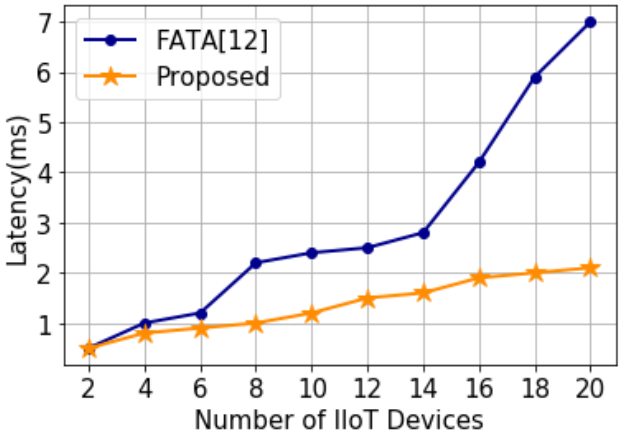

FIGURE 4. Latency analysis of IIoT devices.

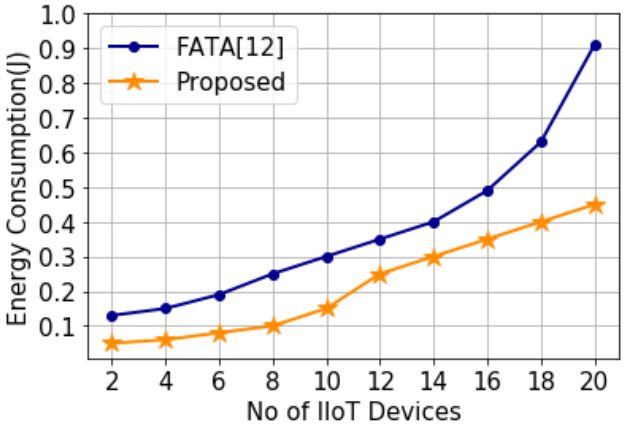

FIGURE 5. Energy consumption of IIoT devices.

which is higher than the proposed one. Similarly, for all IIoT devices, the proposed one achieves lesser energy consumption. The graph demonstrates that the given algorithm achieves the least latency compared to previously used systems, such as FATA [12]. Moreover, the given algorithm is energy-saving and durable, which defines its efficiency.

B. PERFORMANCE ANALYSIS OF IMPROVED RSA

The encryption time achieved by the improved RSA for different key size is compared as shown in Table 4.

TABLE 4. Performance of security algorithms.

Key size(bits)	M		E_t(ms) [12]	E_t(ms) Proposed	D_t(ms) [12]	D_t(ms) Proposed
64	1000		400	231	400	211
128	2000		800	268	800	267
512	3000		1200	312	1200	302
1024	4000		1600	390	1600	345
2048	5000		2000	431	2000	421

The encryption time is varied from 400ms to 2000ms for existing algorithms while as proposed algorithm takes less time ranging from 231ms to 431ms.

The decryption and encryption times are the same for the applied system. However, when it comes to the proposed system, the decryption and encryption times are decreased, with high security as the messages increase.

Analysis of Correlation: The correlation between the original plaintext and its code is used to investigate the existing relationship [31]–[34]. This correlation shows how strongly statistical attacks are protected by the encryption algorithm. To calculate the coefficient of correlation between the two variables, the following formula may be used. (28) [31]–[34]:

$$cc\,(x, y) = \frac{\sum_{i=1}^{n} (x_i - \mu(x))(y_i - \mu(y))}{\sigma\,(x)\,\sigma(y)} \tag{28}$$

$$\mu\,(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and } \mu\,(y) = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{29}$$

$$\sigma\,(x) = \sqrt{\sum_{i=1}^{N} (x_i - \mu(x))^2}, \quad \text{and}$$

$$\sigma\,(y) = \sqrt{\sum_{i=1}^{N} (y_i - \mu(y))^2} \tag{30}$$

where:

$\mu(x)$ denote the expected value of x
$\mu(y)$ denote the expected value of y
(x) denotes the independent variable (plaintext)
(y) denotes the dependent variable (ciphertext)
$\sigma\,(x)$ is the SD of the distribution of plaintext.

The values of the correlation indicate the strength with size and strength being directly proportional. Such values are as shown in Table 5.

TABLE 5. Correlation coefficient values.

Message	Description	Values
M_1	Establishment Data for Connection	0.019
M_2	Data for Device registration	0.014
M_3	Device login data	0.021
M_4	Machine data	0.029
M_5	Machine Control messages	0.072
Average		0.031

The correlation coefficient is calculated using different message typeset as shown in Table 5. The results show that the

proposed improved RSA correlation coefficient is near zero at around 0.031.

C. ANALYSIS OF SOFTMAX-DNN CLASSIFIER

A comparison between the proposed SoftMax-DNN for resource classification and the state-of-the-art forecasting models that include QCI-Neural Network is done in this section [20]. Statistical aspects, including but not limited to the precision, sensitivity, specificity, and the probability of coverage. Besides, the root relative squared error (RRSE), the mean absolute error (MAE), the relative absolute error (RAE), and root average square error (RASE) were investigated and their values are as shown in figure 6.

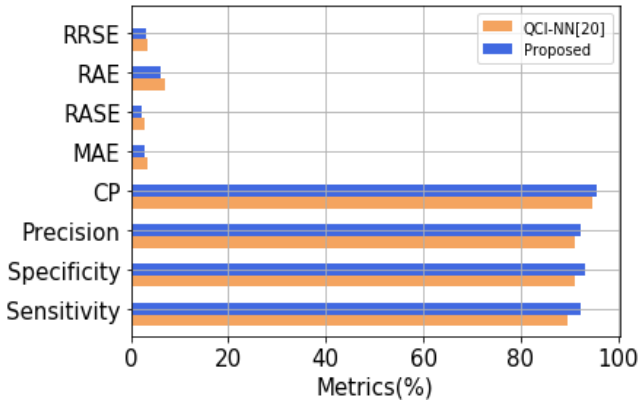

FIGURE 6. Performance of the proposed prediction efficiency.

TABLE 6. Prediction efficiency estimation of the techniques.

Metrics	Techniques	
	QCI-NN [20]	Proposed
Sensitivity	89.76	92.45
Specificity	90.99	93.12
Precision	91.15	92.34
Coverage Probabilities	94.56	95.47
MAE	3.3	2.75
RASE	2.67	2.32
RAE	6.88	6.15
RRSE	3.48	2.98

Table 6 shows that the proposed SoftMax-DNN model achieves performance values of precision (92.34%), sensitivity (92.45%), coverage probability (95.47%), and specificity (93.12%). Moreover, lower values of MAE (2.75%), RASE (2.32%), RAE (6.15%), and RRSE (2.98%) which is better than the existing QCI-NN [20] technique.

D. RESOURCE SCHEDULING AND USER SATISFACTION

After the fog resources were classified, encryption was performed and the resource size was needed in the scheduling process. The specifications of users can be divided into various groups. The appropriate resources are identified and matched with the user requirements, provided they are available.

TABLE 7. Resource schedule.

Classification	Schedule
$\{r_3, r_4, r_7, r_9, r_{18}, r_{17}, r_{20}\}$	$\{t_2 \rightarrow r_4\}$ $\{t_5 \rightarrow r_9\}$ $\{t_9 \rightarrow r_{17}\}$
$\{r_2, r_5, r_{11}, r_{13}, r_{15}, r_{16}, r_{19}\}$	$\{t_3 \rightarrow r_{12}\}$ $\{t_6 \rightarrow r_{14}\}$ $\{t_8 \rightarrow r_2\}$
$\{r_1, r_8, r_7, r_{10}, r_{12}, r_{14}\}$	$\{t_{10} \rightarrow r_{15}\}$ $\{t_1 \rightarrow r_5\}$ $\{t_4 \rightarrow r_{11}\}$ $\{t_7 \rightarrow r_1\}$
$\{r_6, r_{21}, r_{25}, r_{32}, r_{25}, r_{30}, r_{23}\}$	$\{t_{10} \rightarrow r_{16}\}$ $\{t_{12} \rightarrow r_{12}\}$ $\{t_{21} \rightarrow r_{13}\}$ $\{t_{16} \rightarrow r_{15}\}$
$\{r_{22}, r_{31}, r_{29}, r_{28}, r_{26}, r_{24}, r_{27}\}$	$\{t_{11} \rightarrow r_{21}\}$ $\{t_{17} \rightarrow r_{19}\}$ $\{t_{14} \rightarrow r_{18}\}$ $\{t_{15} \rightarrow r_{21}\}$

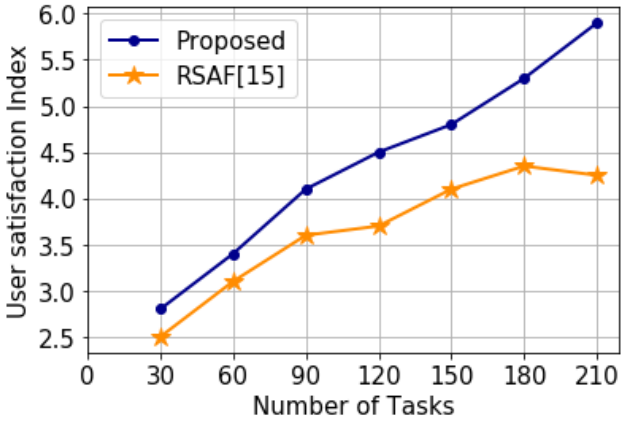

FIGURE 7. User satisfaction index.

Since users have varying needs as far as resources are concerned, the computational, bandwidth, and storage requirements are grouped together. As such, the system incorporates all three types of requirements, which are characterized by application of different resources. Equation (27) measures the attribute that the user requires and the resource attribute and gives the user the highest score as the final result of the resource schedule as shown in Table 7.

The user satisfaction index is calculated with the following equation (31) to investigate the essence of resource scheduling. On the other hand, the user's satisfaction index depends on the s_{com}, s_b and s_s parameters, which represents the task's requirements, storage, and bandwidth attribute. Moreover, computing attributes, bandwidth, and storage attributes are represented by h_{com}, h_b and h_s. The attributes are further presented by μ, ν and τ as the coefficients [28].

$$U_S = \left(\frac{\mu\,(s_{com})}{h_{com}} + \frac{\nu\,(s_b)}{h_b} + \frac{\tau\,(s_s)}{h_s} \right) \tag{31}$$

Figure 7 shows the user's satisfaction index performance varying with the number of tasks for the proposed weight matching based resource scheduling algorithm and the RSAF [15]. For 30 tasks, the proposed algorithm achieves the user satisfaction index of 2.8 whereas the existing RSAF [15] only 2.5, which is lower than the proposed method, and also for the remaining number of tasks (60, 90, 120, 150, 180 and 210), the proposed algorithm achieves the user satisfaction index of 3.4, 4.1, 4.5, 4.8, 5.3 and 5.9, which are higher than the RSAF [15]. The RSAF [15] system ensures that the fastest resources are assigned with the shortest task for faster completion of the process. Consequently, there may

be an occurrence of imbalance in loads, which causes lower satisfaction for the user. Therefore, the proposed algorithm is great is ensuring that the requirement of users is well matched with the fog resources.

V. CONCLUSION

This paper has discussed a secure task scheduling system that is given as an alternative to the previously applied algorithms. This algorithm makes use of the SoftMax-DNN and improved RSA techniques for IIoT applications. For fast data transmission and data deduplication, the proposed task scheduling algorithm uses the NDRF-SSA clustering and SHA-512 algorithm. The outcomes of the proposed algorithm are evaluated and its performance analyzed by comparing the techniques with the existing one. The proposed algorithm attains the lowest latency, the lowest energy consumption, and the highest network lifetime. When comparing to the performance of the improved RSA, the proposed one attains the highest level of security potency when compared to the existing FATA [12] and the proposed improved RSA attains the highest security level when performing both encryption and decryption. In addition, the proposed resource classification algorithm SoftMax-DNN outperforms others. The SoftMax-DNN obtains the highest sensitivity (92.45), specificity (93.12), precision (92.34), coverage (95.47) and attains the lowest error rate for the measures, such as RASE (2.32), RRSE (2.98), and MAE (2.75) when compared to QCI-NN [20]. Therefore, the proposed system produces more desirable results, especially in terms of the speed of data transmission and energy saving. These desirable aspects are made possible with the help of NDRF-SSA clustering and SHA-512 data deduplication.

REFERENCES

[1] W. Hou, L. Guo, and Z. Ning, "Local electricity storage for blockchain-based energy trading in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3610–3619, Jun. 2019.

[2] Olesia Martynova. *Industry 5.0: Announcing the Era of Intelligent Automation*. Accessed: Jun. 1, 2020. [Online]. Available: https://www.intellias.com/industry-5-0-announcing-the-era-of-intelligent-automation/

[3] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial Internet of Things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, Oct. 2018, pp. 1–12, doi: 10.1016/j.compind.2018.04.015.

[4] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.

[5] R. Basir, S. Qaisar, M. Ali, M. Aldwairi, M. I. Ashraf, A. Mahmood, and M. Gidlund, "Fog computing enabling industrial Internet of Things: State-of-the-art and research challenges," *Sensors*, vol. 19, no. 21, p. 4807, Nov. 2019.

[6] R. W. L. Coutinho and A. Boukerche, "Modeling and analysis of a shared edge caching system for connected cars and industrial IoT-based applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2003–2012, Mar. 2020.

[7] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019.

[8] A. E. Kalor, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski, "Network slicing in industry 4.0 applications: Abstraction methods and End-to-End analysis," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5419–5427, Dec. 2018.

[9] M. A. Khan and K. A. Abuhasel, "Advanced metameric dimension (AmD) framework for heterogeneous industrial Internet of Things (HetIoT)," *Comput. Intell.*, to be published, doi: 10.1111/coin.12378.

[10] A. Q. Ansari and M. A. Khan, "Fundamentals of industrial informatics and communication technologies," in *Handbook of Research on Industrial Informatics and Manufacturing Intelligence: Innovations and Solutions*, M. A. Khan and A. Q. Ansari, eds. Hershey, PA, USA: IGI Global, 2012, pp. 1–19, doi: 10.4018/978-1-4666-0294-6.ch001.

[11] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, "A development approach for collective opportunistic edge-of-things services," *Inf. Sci.*, vol. 498, pp. 154–169, Sep. 2019, doi: 10.1016/j.ins.2019.05.058.

[12] S. Sharma and H. Saini, "Fog assisted task allocation and secure deduplication using 2FBO2 and MoWo in cluster-based industrial IoT (IIoT)," *Comput. Commun.*, vol. 152, pp. 187–199, Feb. 2020.

[13] T. Choudhari, M. Moh, and T.-S. Moh, "Prioritized task scheduling in fog computing," in *Proc. ACMSE Conf. ACMSE*, 2018, pp. 1–8.

[14] B. M. Nguyen, H. T. T. Binh, T. T. Anh, and D. B. Son, "Evolutionary algorithms to optimize task scheduling problem for the IoT based Bag-of-Tasks application in cloud-fog computing environment," *Appl. Sci.*, vol. 9, no. 9, p. 1730, Apr. 2019.

[15] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," *Sensors*, vol. 19, no. 9, p. 2122, May 2019.

[16] L. Liu, D. Qi, N. Zhou, and Y. Wu, "A task scheduling algorithm based on classification mining in fog computing environment," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, Aug. 2018.

[17] S. Garg, K. Kaur, G. Kaddoum, and K.-K.-R. Choo, "Toward secure and provable authentication for Internet of Things: Realizing industry 4.0," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4598–4606, May 2020.

[18] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020, doi: 10.1109/ACCESS.2020.2980739.

[19] L. Bu, M. Isakov, and M. A. Kinsy, "A secure and robust scheme for sharing confidential information in IoT systems," *Ad Hoc Netw.*, vol. 92, Sep. 2019, Art. no. 101762.

[20] M. Bhatia, K. S. Sood, and S. Kaur, "Quantumized approach of load scheduling in fog computing environment for IoT applications," *Comput.*, no. 5, pp. 1–19, Jan. 2020, doi: 10.1007/s00607-019-00786-5.

[21] J. A. Alzubi, R. Manikandan, O. A. Alzubi, I. Qiqieh, R. Rahim, D. Gupta, and A. Khanna, "Hashed needham schroeder industrial IoT based cost optimized deep secured data transmission in cloud," *Measurement*, vol. 150, Jan. 2020, Art. no. 107077.

[22] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," *Future Gener. Comput. Syst.*, vol. 110, pp. 1098–1115, Sep. 2020.

[23] H. Sun, H. Yu, G. Fan, and L. Chen, "Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture," *Peer Peer Netw. Appl.*, vol. 13, no. 2, pp. 548–563, Mar. 2020.

[24] W. Wang, G. Wu, Z. Guo, L. Qian, L. Ding, and F. Yang, "Data scheduling and resource optimization for fog computing architecture in industrial IoT," in *Proc. Int. Conf. Distrib. Comput. Internet Technol.* Cham, Switzerland: Springer, 2019, pp. 141–149.

[25] *Secure Hash Standard (SHS)*, Standard TR- FIPS PUB180-4, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2015, doi: 10.6028/NIST.FIPS.180-4.

[26] S. Singh, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, Anchor; Edición: Reprint, 2000.

[27] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[28] W. J. Li, Q. F. Zhang, L. D. Ping, and X. Z. Pan, "Cloud scheduling algorithm based on fuzzy clustering," *J. Commun.* vol. 33, no. 33, pp. 146–154, 2012.

[29] W. Hong-Qiang, L. Xiao-Yong, F. Bin-Xing, and W. Yi-Ping, "Resource scheduling based on improved FCM algorithm for mobile cloud computing," in *Proc. IEEE 22nd Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2016, 128–132.

[30] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing. NATO ASI Series* (Series F: Computer and Systems Sciences), vol. 68, F. F. Soulié and J. Hérault, Eds. Berlin, Germany: Springer, 1990, doi: 10.1007/978-3-642-76153-9_28.

[31] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems—CHES* (Lecture Notes in Computer Science), vol. 3156, M. Joye and J. J. Quisquater, eds. Berlin, Germany: Springer, 2004, pp. 16–29.
[32] I. Elashry, O. Allah, A. Abbas, S. El-Rabaie, and F. El-Samie, "Homomorphic image encryption," *J. Electron. Imag.*, vol. 18, no. 3, 2009, Art. no. 033002.
[33] N. El-Fishawy and O. M. A. Zaid, "Quality of encryption measurement of bitmap images with RC6, MRC6, and rijndael block cipher algorithms," *Int. J. Netw. Secur.*, vol. 5, pp. 241–251, Nov. 2007.
[34] A. I. Sallam, O. S. Faragallah, and E.-S.-M. El-Rabaie, "HEVC selective encryption using RC6 block cipher technique," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1636–1644, Jul. 2018.

KHALED ALI ABUHASEL received the B.Sc. and M.Sc. degrees from the University of Central Florida, Orlando, FL, USA, in 2009 and 2010, respectively, and the Ph.D. degree from New Mexico State University, Las Cruces, NM, USA, in 2012, all in industrial engineering. He is currently an Associate Professor with the Mechanical Engineering Department, University of Bisha, Saudi Arabia. He holds three U.S. patents, and more than 46 publications in journals and proceeding of very reputable conferences. His research interests include optimization, systems engineering, health care systems, intelligent systems, artificial neural network methodologies, and statistical analysis.

MOHAMMAD AYOUB KHAN (Senior Member, IEEE) received the Master of Technology degree in computer science and engineering from Guru Gobind Singh Indraprastha, New Delhi, India, and the Ph.D. degree in computer engineering from Jamia Millia Islamia, New Delhi. He is currently an Associate Professor with the University of Bisha, Saudi Arabia, with interests in the Internet of Things, RFID, wireless sensor networks, ad hoc networks, smart cities, industrial IoT, signal processing, NFC, routing in network-on-chip, and real time and embedded systems. He has more than 14 years of experience in his research areas. He has published many research articles and books in reputed journals and international IEEE conferences. He contributes to the research community by undertaking various volunteer activities in the capacity of editor for many journals and as a Conference Chair.