HHS Public Access

Author manuscript

IEEE Trans Industr Inform. Author manuscript; available in PMC 2022 July 01.

Published in final edited form as:

IEEE Trans Industr Inform. 2021 July ; 17(7): 5128–5137. doi:10.1109/tii.2020.3037872.

A Secured and Intelligent Communication Scheme for IIoT-enabled Pervasive Edge Computing

Fazlullah Khan [Senior Member IEEE]

Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan.

Institute of Social and Economic Research, Duy Tan University, Da Nang, 550000, Vietnam.

Mian Ahmad Jan* **[Senior Member IEEE]**, **Ateeq ur Rehman [Member IEEE]**

Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan.

Spyridon Mastorakis [Member IEEE],

Computer Science Department, University of Nebraska Omaha, NE, USA 68182-0002.

Mamoun Alazab [Senior Member IEEE],

Charles Darwin University 59, Chataway Cr Casuarina, NT, AUS 0811.

Paul Watter [Senior Member IEEE]

School of Engineering and Mathematical Sciences, La Trobe University Melbourne, VIC, AUS 3086.

Abstract

Industrial Internet of Things (IIoT) ensures reliable and efficient data exchanges among the industrial processes using Artificial Intelligence (AI) within the cyber-physical systems. In the IIoT ecosystem, devices of industrial applications communicate with each other with little human intervention. They need to act intelligently to safeguard the data confidentiality and devices' authenticity. The ability to gather, process, and store real-time data depends on the quality of data, network connectivity, and processing capabilities of these devices. Pervasive Edge Computing (PEC) is gaining popularity nowadays due to the resource limitations imposed on the sensor-embedded IIoT devices. PEC processes the gathered data at the network edge to reduce the response time for these devices. However, PEC faces numerous research challenges in terms of secured communication, network connectivity, and resource utilization of the edge servers. To address these challenges, we propose a secured and intelligent communication scheme for PEC in an IIoT-enabled infrastructure. In the proposed scheme, forged identities of adversaries, i.e., Sybil devices, are detected by IIoT devices and shared with edge servers to prevent upstream transmission of their malicious data. Upon Sybil attack detection, each edge server executes a parallel Artificial Bee Colony (pABC) algorithm to perform optimal network configuration of IIoT devices. Each edge server performs the job migration to their neighboring servers for load balancing and better network performance, based on their processing and storage capabilities. The experimental results justify the efficiency of our proposed scheme in terms of Sybil attack

*corresponding author. mianjan@awkum.edu.pk.

detection, the convergence curves of our pABC algorithm, delay, throughput, and control overhead of data communication using PEC for IIoT.

Keywords

Pervasive Edge Computing; Industrial Internet of Things; Artificial Intelligence; Artificial Bee Colony algorithm; Security

1 Introduction

The Industrial Internet of Things (IIoT) is a new paradigm in Information and Communication Technologies (ICTs) that enables the interconnection of existing industrial devices with additional intelligence. These devices sense, process, and communicate data with each other via the Internet [1]. For efficient data processing in IIoT, Pervasive Edge Computing (PEC) has attained significant attention in recent years. Rather than forwarding the data to the cloud for processing, it is processed at the network edge for quick responses to the IIoT devices [2]. PEC reduces the amount of required bandwidth and improves the computation, communication and storage capabilities of these resource-constrained devices [3].

In PEC, the edge servers process the data using Artificial Intelligence (AI) techniques for making certain decisions, e.g. load optimization, channel assignment, etc [4]. The AI-based techniques such as machine learning, deep learning, genetic algorithms, and evolutionary algorithms can learn from the environment by gathering the data, analyzing it, and taking intelligent decisions based on it. For efficient analysis and quicker decisions, IIoT-enabled PEC gains popularity in the use of these algorithms. The Artificial Bee Colony (ABC) [5] is an evolutionary algorithm that aims to embed intelligence and strong inference capabilities in intelligent systems. The variants of this algorithm are used in numerous industrial applications for designing complex systems. For example, in [6], the authors proposed an improved ABC algorithm and evaluated it using different benchmark functions, mainly focused on industrial applications, e.g., infinite impulse response design and industrial image segmentation testing. Authors in [7] proposed a discrete ABC algorithm for disassembling the sequences to recycle and re-manufacture the industrial products using PEC. In [8], an ensemble ABC (En-ABC) was proposed for anomaly detection in an edge-enabled IIoT environment using different datasets. In [9], the ABC algorithm was modified to achieve a QoS-aware secured scheduling in a cloud-based environment for IIoT applications. The use of ABC algorithm in an IIoT-enabled PEC has the potential to make human lives comfortable, however, at the same time, it faces threats in the context of data confidentiality and users privacy [10].

In recent years, the main focus of research for IIoT-enabled PEC has been the design of smart systems [11]. These systems are studied in the context of machine-to-machine communication [12], smart automation [13], intrusion detection [14], and bandwidth utilization [1]. However, limited attention has been given to privacy-preservation and secured communication systems for industrial applications. The low-powered IIoT devices are prone to various adversarial threats that misuse the limited resources of the devices and

cause the Quality of Service (QoS) degradation. Among these threats, Sybil attack is a prominent one, in which an intruder either fabricate or steal the identities of legitimate devices to infiltrate the network [15]. These forged and stolen identities of the intruder have the ability to disrupt the operations of entire network. In [3], the authors proposed a Sybil attack detection scheme using a channel-based machine learning approach for IIoT-enabled PEC. In [16], the authors proposed a secured edge-enabled framework for industrial applications. A lightweight encryption algorithm is executed at low-powered IIoT devices to detect forged identities of Sybil devices, whereas, the resource-intensive operations are performed at the edge servers. In [17], a Blockchain-enable edge framework was proposed to secure industrial applications from Sybil attack to guarantee service availability in an IIoT network.

In this paper, we aim to integrate security, AI, and edge computing to enhance the performance of IIoT-enabled PEC. The proposed scheme uses a three layer architecture. The bottom layer is IIoT architecture layer that is composed of low-powered IIoT devices for data collection and Sybil attack detection. The middle layer comprises edge servers for load optimization and job migration using our proposed parallel ABC (pABC) algorithm. Finally, the top layer is the core layer that includes the industrial cloud for data accumulation and storage. The main contributions of this paper are as follow.

- There does not exist any scheme for a secured and intelligent IIoT-enabled PEC in the existing literature. This scheme is unique because it secures the IIoT architecture layer, and performs optimal configuration at the edge layer using pABC. This results in efficient resource utilization by improving the performance of IIoT-enabled PEC.
- To protect the low-powered IIoT devices from Sybil attack, we propose a lightweight Sybil attack detection protocol. The Sybil devices are detected by IIoT devices and the results are shared with neighboring devices and also reported to the edge servers. This ensures the blockage of upstream transmission of fabricated data towards the industrial cloud.
- For efficient utilization of the network resources, the pABC algorithm is used to optimize the core layer using PEC. Upon Sybil attack detection, the pABC algorithm is executed in parallel at each edge server to perform their optimal configuration with the IIoT devices. The heavy loaded servers mutually migrate jobs to the neighboring servers that result in efficient resource utilization and network performance.
- Finally, we perform extensive simulations in terms of accuracy, sensitivity, specificity of Sybil device detection, and convergence curves of pABC algorithm to prove the efficiency of the proposed scheme. Moreover, we evaluated delay, throughput, and control overhead of the data communication within the network.

The rest of the paper is organized as follows. The system model is discussed in Section 2. In Section 3, the proposed scheme is illustrated followed by results and discussion in Section 4. Section 5 concludes the paper with future research directions.

2 System Model

In this section, we discuss the system model of our proposed IIoT-enabled PEC scheme. The system model is composed of three layers, the IIoT architecture layer, the edge layer, and the core network layer, as shown in Fig. 1. The bottom layer is composed of low-powered systems and sensor-embedded industrial devices. These devices need to be utilized efficiently with authentic data exchanges and interoperable connectivity to the rest of network entities. The middle layer consists of edge servers (E_s) that performs different functions, e.g. data exchanges, storage, and computation. The top layer has two sub layers, i.e., core networks, and the cloud services. The core networks sub-layer is similar to a conventional network with an exception of traffic profile. It is responsible to perform routing, data exchanges, and management of network information among different sub-networks. The cloud service is responsible for hosting various applications that provide different services.

In the system model, the IIoT architecture layer has no security mechanism to protect the low-powered IIoT devices from malicious threats such as DoS and Sybil attacks. The low-powered IIoT devices need to be secured, and an optimal configuration of these devices with E_ss need to be maintained for efficient resource utilization and network performance. To protect the IIoT devices, we proposed a Sybil attack detection scheme to prevent the malicious entities from the transmission of compromised data. For efficient utilization of the network resources at the edge layer, we propose a pABC algorithm. In Section 3, we discuss them in detail.

3 The Proposed Scheme

In this section, we design a Sybil attack detection protocol to securely utilize the resources of low-powered, sensor-embedded industrial devices. PEC is used to process and securely forward the data to the industrial cloud from these devices. Upon network initialization, our pABC algorithm runs in parallel at each E_s. This algorithm performs optimal configuration of IIoT devices and E_s that results in faster delivery of information to the industrial cloud. The quick delivery of information is not only due to optimal configuration but also due to Sybil attack detection, which prevents the flow of compromised data in the network. Besides optimal configuration, each E_s performs job migration when the gathered data exceed its storage and processing capability. This improves the efficient utilization of the network resources, resulting in higher throughput and minimum delay during data transmission.

3.1 Sybil Attack Detection and Prevention Model

Our proposed Sybil attack detection protocol uses signalprints, which is a vector of RSSI values received from multiple sources. Transmissions from the same location share the same channel response, i.e., the same RSSI, while it varies for transmissions coming from different locations. Like all other IIoT devices, the transmission from a Sybil device $device_s$ is omnidirectional. Neighboring IIoT devices, e.g. $device_i$ and $device_j$ receive the transmission from $device_s$ and they coordinate and evaluate the signal strength of this

transmission. When the $device_i$ and $device_j$ receive a control signal from $device_s$ with illicit identity h, then the received power $R^h_{device_i}$ can be computed using Eq. (1).

$$R^h_{device_i} = \frac{T_x \cdot c}{d^\rho_{device_i}} \tag{1}$$

Here, T_x is the transmitted power, c is a constant, $d^\rho_{device_i}$ is the Euclidean distance between sender and receiver, and ρ is the path-loss exponent and is equal to $4\pi d_{device_i}/\lambda$ [18]. In ρ, λ represents the wavelength of a radio signal. Similar to $device_i$, the neighboring $device_j$ also computes its received RSSI $R^h_{device_j}$.

The main motive of this protocol is to detect Sybil devices using their untrusted and false RSSI values. These RSSI values mostly have limited transmission power T_x, which is related to received power R_x, as shown in Eq. (2).

$$T_x = \frac{R_x}{\left(1/d_{device_i}\right)^\rho} \tag{2}$$

The location of $device_s$ with respect to $device_i$ can be computed using Euclidean distance given in Eq. 3.

$$d_{device_i} = \sqrt{\left(x_{device_i} - x_{device_s}\right)^2 - \left(y_{device_i} - y_{device_s}\right)^2} \tag{3}$$

Next, $device_i$ computes and appends $R^h_{device_i}$ to its own control packet and transmits it to $device_j$. As discussed earlier, $device_j$ has calculated $R^h_{device_j}$ after receiving a similar control packet from $device_s$ at time t_1. Putting the values of Eq. (2), (3), and ρ in Eq. (1), we can write

$$\frac{R^h_{device_j}}{R^h_{device_i}} = \left(\frac{R_x \cdot c}{d^\rho_{device_j}}\right) / \left(\frac{R_x \cdot c}{d^\rho_{device_i}}\right). \tag{4}$$

Further solving Eq. 4 results in

$$\frac{R^h_{device_j}}{R^h_{device_i}} = \left(\frac{d_{device_i}}{d_{device_j}}\right)^\rho, \text{where} \, t = t_1 \, . \tag{5}$$

At time t_0+t_1, $device_s$ again broadcasts control packets with a different identity $\hbar$. The two IIoT devices repeat the aforementioned operations and compute the RSSI ratio using Eq. 6.

$$\frac{R^{\hbar}_{device_j}}{R^{\hbar}_{device_i}} = \left(\frac{d_{device_i}}{d_{device_j}}\right)^{\rho}, \text{when} \, t = t_0 + t_1 \, . \tag{6}$$

At this point, $device_j$ compares the RSSI ratio of Eqs. (5) and (6), and declares if the control packet received is from $device_s$ or not? A control packet belongs to $device_s$ iff Eq. (7) holds.

$$\frac{R^{h}_{device_j}}{R^{h}_{device_i}} - \frac{R^{\hbar}_{device_j}}{R^{\hbar}_{device_i}} \approx 0. \tag{7}$$

An IIoT device that does not trust other IIoT devices and believes in their RSSI values (genuineness) is known as the initiator. The initiator can label other IIoT devices as Sybil or genuine IIoT devices. The initiator becomes an observer after sharing its knowledge with neighboring devices. In this way, any device is classified as Sybil if the observer and initiator have computed the same RSSI value for that device. In other words, a Sybil device $device_s$ has presented two illicit identities, h and $\hbar$ to its neighboring IIoT devices. The RSSI values calculated by neighboring devices are equal at different time periods that reflect the incoming transmission from the same location.

At the IIoT architecture layer , all the devices perform a similar operation to detect Sybil devices and their illicit multiple identities. Once the Sybil devices are detected, the IIoT devices share their forged or fabricated identities with their connected edge servers E_s to prevent upstream transmission of malicious data towards the core network using PEC. Once the Sybil devices and their forged identities are prevented from network participation, the IIoT devices transmit their own data upstream towards the E_s. The proposed pABC algorithm is executed in parallel at each E_s for optimal configuration of IIoT devices to each E_s. In the next section, we discuss ABC and pABC algorithms in the context of our proposed scheme.

3.2 Artificial Bee Colony Algorithm

The ABC algorithm is a probabilistic searching evolutionary algorithm used to solve optimization problems. It has three basic components, food sources, employed foragers, and unemployed foragers [5]. The food sources refer to solutions in the optimization problems that depend on proximity, richness, and ease of extraction, i.e., the objective function. The quality of food is considered as a fitness value. The employed foragers are responsible for collecting honey, which refers to finding an optimal solution for solving the optimization problems. The unemployed foragers are categorized into onlookers and scouts bees. The onlooker bees observe the employed bees on the dancing area to know about a food source (solution set), whereas the scout bees perform random search about a food source.

The ABC algorithm has numerous advantages over other intelligent evolutionary algorithms. For example, it has better exploration ability as the scout bees always generate new solutions that are different than the previous ones. In this way, a diverse set of solutions is produced that can prevent the premature convergence issue in the network. The ABC algorithm is

extremely simple as it uses only two control parameters. It is also efficient in local search as compared to GA, ACO, and PSO, due to its simplicity. However, its search time is longer and new solutions are generated based on the old solutions, which reduce the local optimization accuracy and convergence time. The process of collecting honey and optimization problems are given in Table 1. In the following sections, we describe the mapping of ABC algorithm to our proposed system model, the phases of this algorithm, and our proposed pABC algorithm that runs in parallel at E_s.

3.2.1 Mapping—The applications of ABC algorithm need the design of chromosomes, which is a set of parameters required to solve a particular problem. In other words, the chromosomes is a possible arrangement to find an optimal solution. In our proposed scheme, we have represented chromosomes with scalar values. In this section, we map the decision variables of our objective function to solve the target problem. The decision variables, i.e., chromosomes, below are used to find an optimal solution.

- S_c: Storage capacity of the edge server
- P_c: Processing capability of the edge server
- L_c: Location of the edge server

Like other evolutionary algorithms, in our scheme, a population is a large set of individuals, where each individual is represented by E_s. The E_s uses chromosomes for reaching to an optimal solution.

3.2.2 Population Initialization—Before initializing the population of ABC algorithm, we need to define the swarm size S, the number of cycles T, and the limit variable. We determine the food sources F_s, the number of employed bees E_b, the onlooker bees O_b using S, i.e., $F_s = E_b = O_b = \frac{S}{2}$. Next, we initialize random solutions for the population within the boundaries of our objective function, as shown in Eq. (8).

$$x_i^j = x_{lb}^j + rand(0,1)\left(x_{ub}^j - x_{lb}^j\right) \tag{8}$$

Here, $x_{lb}^j$is the lower bound and x_{ub}^j is the upper bound of x_I in the j^{th} direction. Now, the fitness values of each solution can be computed using Eq. (9).

$$fit = \begin{cases} \frac{1}{1+obj()}, & \text{if obj}() \geq 0 \\ 1 + abs(obj()), & \text{if obj}() < 0 \end{cases} \tag{9}$$

In this equation, obj() is the objective function as defined in Eq. (10).

$$obj() = \sum_{i=1}^{n} x_i \tag{10}$$

Here, x_i is the decision variable. We also generate the initial trial vector that is used during the scout phase.

3.2.3 Employed Bee Phase—In the population initialization, a set of random solutions is generated using Eq. 8. From the generated solution set, choose the first solution (candidate solution) and a partner solution from the remaining solution set. Select a decision variable in the first chosen solution to generate a new solution using Eq. (11)

$$x_{new}^{j} = x_{min}^{j} + \phi_i^j\left(x_{min}^{j} - x_p^j\right) \tag{11}$$

where, $x_{new}^{j}$is the new generated decision variable, x^j is the old decision variable, ϕ is a function that generates the random numbers $\in [-1,1]$, and x_p^j is the corresponding decision variable in the partner solution. The employed bees check the bounds of x_{new}^{j}, evaluates the fitness using Eq. (9), and updates the solution using a greedy selection procedure, given in Eqs. (13) and (14). If x_{new}^{j} violates the bounds, then apply Eq. (12). The pseudo code of E_b phase is provided in Algorithm 1.

$$x_{new}^{j} = \begin{cases} max\left(x_{new}^{j}, lb^i\right), & \text{for lower bound violation} \\ min\left(x_{new}^{j}, ub^i\right), & \text{for upper bound violation} \end{cases} \tag{12}$$

$$x_{new}^{j} = \begin{cases} x = x_{new}^{j}, & \text{if fit}_{new} \geq \text{fit} \\ x = x, & \text{if fit}_{new} < \text{fit} \end{cases} \tag{13}$$

$$obj()_{new} = \begin{cases} obj() = obj()_{new}, & \text{if fit}_{new} \geq \text{fit} \\ obj() = obj(), & \text{if fit}_{new} < \text{fit} \end{cases} \tag{14}$$

Algorithm 1 Pseudocode of Employed Bee Phase

Initialization:
input: obj(), lb, ub, S, p, fit, trial.

```
procedure
    for i = 1 to S/2 do
      Randomly select a partner solution p
      Randomly select a variable for modification in
  partner solution p            ▷ variable x_p^j in Eq (11).
      Bound x_new^j using Eq. (12)
      Evaluate obj()_new and fit_new using Eq. (10) & (9)
      if fit_new >fit
       Accept x_new
       trial_i = 0
      else
       trial_i = trial_i + 1
      end if
    end for
end procedure
```

3.2.4 Onlooker Bee Phase—The onlooker bee phase resembles the employed bee phase as shown in Algorithm 2. In this phase, we need the probability of each solution $Prob_i$, which is calculated using Eq. (15),

$$Prob_i = 0.9 \times \left(\frac{fit_i}{max(fit)}\right) + 0.1. \tag{15}$$

After obtaining the probabilities of every solution, the onlookers select the first F_s, and compare the probability of first solution with a random number $r \in [0,1]$. If $r > Prob_i$, then the onlookers do not generate a new F_s, and they repeat this process continuously. If $r < Prob_i$, generate a new F_s by selecting a random decision variable in the current F_s using Eq. (11). Check bounds of the new F_s and fix it using Eq. (12), evaluate fitness using Eq. 9, and update the solution using greedy selection given in Eqs. (13) and (14).

3.2.5 Scout Bee Phase—The employed bees whose F_s are abandoned are known as the scout bees. Each scout bee uses a trail vector that is generated during the initialization phase. The trail vector keeps track of how many times it failed to generate a better solution in comparison to the existing solution. During the scout bee phase, new solutions are generated using Eq. (8). The previous best solutions stored in trial vector are compared with the newly generated solution based on its fitness value. The solution with the best fitness value is maintained in the trial vector. The pseudocode of the scout bee phase is provided in Algorithm 3.

3.2.6 Parallel Artificial Bee Colony Algorithm—The issues in the traditional ABC algorithm are its long search time, slow convergence rate, and proneness to local optimum on a large number of decision variables. To overcome these issues, we have made changes to all the three phases, i.e., employed bee, onlooker bee, and scout bee. The employed and onlooker bee phases use Eq. (11) to generate a new solution and a new F_s, respectively. These newly generated solutions are not much different from the previous solutions, resulting in the least improvements after many iterations. It causes premature convergence when the optimal solution is not detected, which shows the inefficiency of the traditional ABC algorithm. To overcome this issue, we modify Eq. (11) to generate new solutions with a good convergence rate using Eq. (16).

$$x_{new}^j = x_i^j rand() + \phi_i^j\left(x_i^j - x_k^j\right) + \left(\psi_i^j - 0.5\right)^2\left(y_i^j - x_i^j\right) \tag{16}$$

where, $\left(\psi_i^j - 0.5\right)^2\left(y_i^j - x_i^j\right)$ is the *gbest* that helps in efficient optimal convergence, and ψ_i^j is the new mutation step size, calculated in Eq. (17).

$$\psi_i^j = 1 - \frac{t}{T} \tag{17}$$

where, t is the current iteration, and T is the total number of iterations. Furthermore, we modified Eq. (8) for the scout phase, as shown in Eq. (18). This equation generates a better solution than the original scout bee phase solution. Here, x_{best}^j is the previous best term saved.

$$x_{new}^{j} = \begin{cases} x_{min}^{j} + rand() * \psi_i^j * \left(x_{best}^{j} - x_i^j\right), & \text{for rand}() \leq 0.5 \\ x_{min}^{j} + rand() * \psi_i^j * \left(y_{best}^{j} - x_i^j\right), & \text{for rand}() > 0.5 \end{cases} \quad (18)$$

Algorithm 2 Pseudocode of Onlooker Bee Phase

Initialization: m=0, i=1
input: obj(), lb, ub, S, p, fit, trial, prob.

procedure
  while m ≤ S/2 **do**
    r = rand(0,1)
    if r < $prob_i$
     Randomly select a partner p
     Randomly select a variable for modification in partner solution p
     Bound x_{new}^{j} using Eq. 12
     Evaluate $obj()_{new}$ & fit_{new} using Eq. (10) & (9)
     if fit_{new} > fit
      Accept x_{new}
      $trial_i$ = 0
      m = m + 1
     else
      $trial_i$ = $trial_i$ + 1
      m = m + 1
     end if
  i = i + 1
  if i > S/2
    i =1
  end if
  end while
end procedure

Algorithm 3 Pseudocode of Employed Bee Phase

Initialization:
input: obj(), lb, ub, P, trial, limit.

procedure
  if trial of a solution > limit
   Replace the entire solution within P using Eq. (8)
    Evaluate obj() and fit
  end if
end procedure

Our proposed pABC algorithm needs to be executed simultaneously by all edge servers for optimal configuration of IIoT devices to the E_s. Each E_s needs a random selection of a group of networks, and evaluate their fitness and objective functions using pABC, respectively. Then, using Algorithms 1, 2, and 3, the new best solutions can easily be computed. Finally, these algorithms search the local best solution and decide to migrate the solution with the neighboring servers based on a user-defined threshold value (λ). After finding the best solution and optimal configuration, the E_s performs job migration to the neighboring E_ss if the gathered data from IIoT devices is larger than its storage and processing capabilities. The pseudo-code of the pABC is given in Algorithm 4.

4 Experimental Results and Discussion

In this section, the proposed scheme is analyzed using simulation-based results. We have compared our Sybil attack detection protocol with [19], [20], and [21] using the confusion matrix given in Table 2. On the other hand, the convergence of our pABC algorithm is compared against Genetic Algorithm [22], Artificial Bee Colony [5], and Particle Swarm Optimization [23] using mean values of different benchmark functions. Finally, the delay, throughput, and control overhead are compared with [1] and [5].

4.1 Sybil Attack Detection

The performance of our Sybil attack detection scheme is compared using Table 2, where the true positive T_P reflects the occurrence of Sybil attack that was detected instantly, while true negative T_N shows no attack and correctly detected. A false-positive F_P is the case when a genuine activity is detected as an attack and F_N is the case when an attack is detected as a genuine activity. The IIoT-enabled PEC uses the performance metrics such as accuracy, sensitivity, and specificity for comparison. Accuracy is the probability of accurate detection as shown in Eq. (19). Sensitivity is the percentage of positive events as shown in Eq. (20), whereas Specificity is the adverse events correctly detected as shown in Eq. (21).

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{19}$$

$$Sensitivity = \frac{T_P}{T_P + F_N} \tag{20}$$

$$Specificity = \frac{T_N}{F_P + T_N} \tag{21}$$

In Figs. 2, 3, and 4, the effect of Sybil devices on the aforementioned performance metrics is highlighted. The number of forged or fabricated identities of each device varies between 10 and 26. In comparison to the existing schemes, our scheme performs better in terms of these performance metrics for a varying number of forged identities. The values on the y-axis for all metrics increase by increasing the number of forged identities. The higher number of Sybil nodes increase the chances of their detection due to their increased fabricated identities. The main reason for better performance of our scheme is the use of RSSI values from multiple sources. The RSSI-enabled detection of our scheme allows it to detect even the smallest of fluctuation in the signal strength coming from the same physical location. The performance of [19–21] is slightly lower than our proposed scheme because they have used a single RSSI value for Sybil attack detection and also there is high fluctuation in these values experienced at IIoT devices.

Algorithm 4 Pseudocode of Parallel ABC Algorithm

Initialization:
input: obj(), lb, ub, T, limit, c_i=0.

procedure
  Initialize random population P for each colony (network) in parallel using Eq. (8)
  Evaluate obj() and fit of each population using Eq. (10) and (9)
  Set the trial counter of all solutions to zero
  for t=1 to T **do in parallel**
    Perform Employed Bee Phase for all solutions using Algorithm 1
    Determine the probability of each solutions
    Perform Onlooker Bee Phase to generate solutions using Algorithm 2
    Memorize the best solutions
    if trial of a solution > limit
      Perform Scout Bee Phase using Algorithm 3
    end if
  end for
  for each local best solution **do**
    if fit(x^i_{cur}) < fit(x^i_{pre}) **do**
      $c_i = c_i + 1$
    else-if $c_i \geq \lambda$ ▷ λ is a user-defined threshold.
      Migrate solutions to neighbors and update $network_i$
    end if
  end for
  for each E_s **do**
    if (collected data) > (the capacity of E_s) **do**
      Send request to neighboring E_s
      if *response* received in-time **do**
        submit job to neighboring E_s
      end if
    end if
  end for
end procedure

4.2 Optimization using Parallel ABC

In this section, we discuss the effectiveness of pABC algorithm using benchmark functions [24] of Table 3. These functions are used to compare our pABC algorithm with conventional ABC [5], PSO [23], and GA [22] as shown in Fig. 5, 6, and 7, respectively. In these figures, the values on y-axis decrease by increasing the number of iterations. The convergence curves of pABC are better because of the improved step size. Moreover, it has better new solutions every time in comparison to the previous solutions. The modifications at every phase of pABC result in faster convergence that gives an optimal network configuration in a shorter time. It has also a good effect on achieving a higher QoS of the network. Moreover, our proposed pABC algorithm runs in parallel at each E_s, and shares the results with neighboring E_ss that helps in the learning process and faster convergence.

4.3 Network Performance Metrics

In this section, we demonstrate various performance metrics, e.g. end-to-end delay, throughput, and control overhead of our IIoT-enabled PEC.

4.3.1 End-to-End Delay—The End-to-End Delay (E2E) is the total delay incurred by a packet from its transmission to its final reception at the industrial cloud. In Fig. 8, the E2E delay against the number of transmitting devices for different schemes is depicted. The largest delay is incurred when the network is under a Sybil attack. Moreover, each packet experiences a higher delay under normal network operations, i.e., without optimization. The delay of ABC algorithm is better than [1] due to its efficient convergence and in finding optimal solutions. In [1], authors have performed optimal configuration of IIoT devices with edge servers using PSO. They have also used high-bandwidth communication channels in the licensed band using cognitive radio. Our proposed scheme outperforms all these schemes because it efficiently detects the Sybil attack at the IIoT architecture layer, and achieves an optimal network configuration at the Edge layer using pABC. Besides, pABC executes simultaneously at different E_s that causes minimum delay and performs efficient job migration under heavy load.

4.3.2 Throughput—Throughput is defined as the number of data packets correctly transmitted over a network in a particular period. In Fig. 9, the throughput of different schemes against the number of transmitting devices is shown. The throughput is degraded when the network is under a Sybil attack. The throughput of [1] is better against the network without optimization due to the optimal device-to-gateway configuration and the use of high-bandwidth licensed channels. The ABC algorithm performs better than [1] due to efficient convergence and the fast optimal solution. Our pABC algorithm outperforms all these schemes due to Sybil attack detection, and optimal network configuration.

4.3.3 Control Overhead—Fig. 10 compares the control overhead of different schemes, which is the ratio of the control packet to data packets transmitted. During Sybil attack, the data is not successfully transmitted resulting in a higher overhead. In [1], the overhead is smaller when running PSO and it increases while detecting licensed channel in a spectral band. It is because the licensed channel detection needs various control messages. The pABC algorithm has higher control overhead in comparison to ABC because the proposed scheme transmits the control messages during Sybil device detection.

5 Conclusion

In this paper, we proposed a secured and intelligent communication scheme for the Industrial Internet of Things (IIoT)-enabled Pervasive Edge Computing (PEC). The proposed scheme relies on Sybil attack detection, Artificial Intelligence (AI)-based optimal configuration, and edge computing. Using a lightweight detection protocol, the Sybil devices were detected by the IIoT devices. After Sybil devices detection, their illicit identities are shared with the neighbors and edge servers to prevent upstream transmission of malicious data towards the industrial cloud. At the edge server, an optimal configuration of IIoT devices was achieved by executing a parallel Artificial Bee Colony (pABC) algorithm. This algorithm identified an optimal configuration of edge servers to the IIoT devices, and shared the results with the neighbors. Each edge server performed job migration to its neighbors when the incoming data from IIoT devices exceeded its processing and storage capabilities. The experiments results showed that our scheme is resilient against Sybil attack in terms of accuracy, sensitivity, specificity, and detection rate. Moreover, it has better convergence

curves against the existing approaches for various benchmark functions and achieved higher Quality of Service (QoS) of the network. In the future, we aim to perform mutual authentication among edge servers and IIoT devices prior to communication. Also, we aim to improve the performance of pABC by regulating the traffic from industrial cloud via Software-defined Networks (SDNs) controller.

Acknowledgements

This work is partially supported by a pilot award from the Center for Research in Human Movement Variability and the NIH (P20GM109090) and a planning award from the Collaboration Initiative of the University of Nebraska system.

Biographies

Dr. Fazlullah Khan is a faculty member at the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. He had been the recipient of various prestigious scholarships during his PhD studies and has been awarded the best researcher awarded for the year 2017. His research interests are Intelligent and robust protocol designs, Security and Privacy of Wireless Communication Systems, Internet of Things, Machine Learning, Artificial Intelligence. Recently, he has been involved in latest developments in the field of Internet of Vehicles security and privacy issues, Software-defined Networks, Fog Computing and Big Data Analytics. He has published his research work in top-notch journals and conferences. His research has been published in IEEE Transactions on Industrial Informatics, IEEE Internet of Things, IEEE Access, Elsevier Computer Networks, Elsevier Future Generations Computer Systems, Elsevier Journal of Network and Computer Applications, Elsevier Computers and Electrical Engineering, Springer Mobile Networks and Applications. He has served over 10 conferences in leadership capacities including General Chair, General co-Chair, program co-Chair, track Chair, session Chair, and Technical Program Committee member, including IEEE TrustCom 2017, 2018, EuroCom, GCCE 2019, ITNG 2018, Future5V 2017, CCODE-2017, IoT-BC2 2016. He has been an active reviewer for high-cited and highly ranked international journals, including IEEE Transactions on Dependable and Secure Computing (TDSC), Elsevier Computer Networks, Springer Mobile Networks & Applications and Wiley Concurrency and Computation: Practice and Experience.

Dr. Mian Ahmad Jan is an assistant professor at the department of computer science, Abdul Wali Khan University Mardan, Pakistan. He completed his PhD at the University of Technology Sydney (UTS), Australia. He had been the recipient of various prestigious

scholarships during his PhD studies. He was the recipient of International Research Scholarship (IRS), UTS and Commonwealth Scientific Industrial Research Organization (CSIRO) scholarships. He has been awarded the best researcher awarded for the year 2014 at the University of Technology Sydney Australia. His research interests include energy-efficient and secured communication in Wireless Sensor Networks and Internet of Things. Recently, he has been actively involved in machine learning, big data analytics, smart cities infrastructure and vehicular ad hoc networks. His research has been published in IEEE Transactions on Mobile Computing, IEEE Transactions on Cloud Computing, IEEE Transactions on Industrial Informatics, IEEE Transactions on Network Science and Engineering, IEEE Internet of Things Journal, IEEE Journal of Selected Areas of Communications and ACM Computing Surveys are few to mention. He has been guest editor of numerous special issues in various prestigious journals such as IEEE Transactions on Industrial Information, Springer Neural Networks and Applications, and Elsevier Future Generation Computer Systems etc.

Dr. Ateeq ur Rehman is Assistant professor of Computer Science at Abdul Wali Khan University Mardan, Pakistan. He received his BEng degree in Computer Science and Information Technology from the Islamic University of Technology Dhaka, Bangladesh, in 2009. He got his PhD degree in wireless communications from the University of Southampton in January 2017. His major research interests are next generation wireless communications and cognitive radio networks, Cooperative communication, Resource allocation, particularly cross layer approach and Hybrid ARQ. Currently, he is working on security and privacy of Internet of Things using machine learning algorithms. He published 20 quality scholarly articles. His most recent research achievements have been published in several highly cited IEEE Transactions and Elsevier journals including IEEE Internet of Things journal, Future Generation Computer Systems, and Computer Networks. His research contribution on network security is internationally recognized.

Dr. Spyridon Mastorakis is an Assistant Professor in Computer Science at the University of Nebraska Omaha. He received his Ph.D. in Computer Science from the University of California, Los Angeles (UCLA) in 2019. He also received an MS in Computer Science from UCLA in 2017 and a 5-year diploma (equivalent to M.Eng.) in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 2014. His research interests include network systems and protocols, Internet architectures, IoT and edge computing, and security.

Dr. Mamoun Alazab is an Associate Professor at the College of Engineering, IT and Environment at Charles Darwin University, Australia. He received his PhD degree in Computer Science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is a cyber security researcher and practitioner with industry and academic experience. AlazabâĂŹs research is multidisciplinary that focuses on cyber security and digital forensics of computer systems with a focus on cybercrime detection and prevention. He has more than 150 research papers in many international journals and conferences, such as IEEE transactions on Industrial Informatics, IEEE Transactions on Industry Applications, IEEE Transactions on Big Data, IEEE Transactions on Vehicular Technology, Computers & Security, and Future Generation Computing Systems. He delivered many invited and keynote speeches, 24 events in 2019 alone. He convened and chaired more than 50 conferences and workshops. He works closely with government and industry on many projects, including Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, the Australian Federal Police (AFP), the Australian Communications and Media Authority (ACMA), Westpac, United Nations Office on Drugs and Crime (UNODC), and the Attorney GeneralâĂŹs Department. He is a Senior Member of the IEEE. He is the Founding chair of the IEEE Northern Territory (NT) Subsection.

Prof. Paul Watters is Adjunct Professor of Cybersecurity at La Trobe University, and Honorary Professor at Macquarie University. He is a Chartered IT Professional, a Fellow of the British Computer Society, a Senior Member of the IEEE, and a Member of the Australian Psychological Society. Professor Watters is Academic Dean at Australasian Academies Polytechnic, an ASX-listed education provider. He is also Australia's leading trusted cybersecurity advisor, thought leader, and founder of Cyberstronomy Pty Ltd, home of the www.100pointcybercheck.com.

References

[1]. Yao W, Khan F, Jan MA, Shah N, Rahman I. ur, Yahya A, and ur Rehman A, "Artificial intelligence-based load optimization in cognitive internet of things," Neural Computing and Applications, pp. 1–11, 2020.

[2]. Sun W, Liu J, Yue Y, and Zhang H, "Double auction-based resource allocation for mobile edge computing in industrial internet of things," IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4692–4701, 2018.

[3]. Chen S, Pang Z, Wen H, Yu K, Zhang T, and Lu Y, "Automated labeling and learning for physical layer authentication against clone node and sybil attacks in industrial wireless edge networks," IEEE Transactions on Industrial Informatics, 2020.

[4]. Sangaiah AK, Medhane DV, Han T, Hossain MS, and Muhammad G, “Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics,” IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4189–4196, 2019.

[5]. Karaboga D, “An idea based on honey bee swarm for numerical optimization,” tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer, 2005.

[6]. Gao H, Shi Y, Pun C-M, and Kwong S, “An improved artificial bee colony algorithm with its application,” IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 1853–1865, 2018.

[7]. Tian G, Ren Y, Feng Y, Zhou M, Zhang H, and Tan J, “Modeling and planning for dual-objective selective disassembly using and/or graph and discrete artificial bee colony,” IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2456–2468, 2018.

[8]. Garg S, Kaur K, Batra S, Aujla GS, Morgan G, Kumar N, Zomaya AY, and Ranjan R, “En-abc: An ensemble artificial bee colony based anomaly detection scheme for cloud environment,” Journal of Parallel & Dist. Computing, vol. 135, pp. 219–233, 2020.

[9]. Thanka MR, Maheswari PU, and Edwin EB, “An improved efficient: Artificial bee colony algorithm for security and qos aware scheduling in cloud computing environment,” Cluster Computing, vol. 22, no. 5, pp. 10905–10913, 2019.

[10]. Makhdoom I, Abolhasan M, Lipman J, Liu RP, and Ni W, “Anatomy of threats to the internet of things,” IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1636–1675, 2018.

[11]. Chen J, Li K, Deng Q, Li K, and Philip SY, “Distributed deep learning model for intelligent video surveillance systems with edge computing,” IEEE Transactions on Industrial Informatics, 2019.

[12]. Wang T, Luo H, Jia W, Liu A, and Xie M, “Mtes: An intelligent trust evaluation scheme in sensor-cloud enabled industrial internet of things,” IEEE Transactions on Industrial Informatics, 2019.

[13]. Zhang Y, Qian C, Lv J, and Liu Y, “Agent and cyberphysical system based self-organizing and self-adaptive intelligent shopfloor,” IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 737–747, 2016.

[14]. Gao X, Shan C, Hu C, Niu Z, and Liu Z, “An adaptive ensemble machine learning model for intrusion detection,” IEEE Access, vol. 7, pp. 82512–82521, 2019.

[15]. Zhang S and Lee J-H, “Double-spending with a sybil attack in the bitcoin decentralized network,” IEEE Transactions on Industrial Informatics, vol. 15, no. 10, pp. 5715–5722, 2019.

[16]. Jan MA, Zhang W, Usman M, Tan Z, Khan F, and Luo E, “Smartedge: An end-to-end encryption framework for an edge-enabled smart city application,” Journal of Network and Computer Applications, vol. 137, pp. 1–10, 2019.

[17]. Xu J, Wang S, Zhou A, and Yang F, “Edgence: A blockchain-enabled edge-computing platform for intelligent iot-based dapps,” China Communications, vol. 17, no. 4, pp. 78–87, 2020.

[18]. Liu X, Qiu T, Zhou X, Wang T, Yang L, and Chang V, “Latency-aware path planning for disconnected sensor networks with mobile sinks,” IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 350–361, 2020.

[19]. Jan MA, Nanda P, He X, and Liu RP, “A sybil attack detection scheme for a forest wildfire monitoring application,” Future Generation Computer Systems, vol. 80, pp. 613–626, 2018.

[20]. Ssu K-F, Wang W-T, and Chang W-C, “Detecting sybil attacks in wireless sensor networks using neighboring information,” Computer Networks, vol. 53, no. 18, pp. 3042–3056, 2009.

[21]. Dong W and Liu X, “Robust and secure time-synchronization against sybil attacks for sensor networks,” IEEE Transactions on Industrial Informatics, vol. 11, no. 6, pp. 1482–1491, 2015.

[22]. Tang K-S, Man K-F, Kwong S, and He Q, “Genetic algorithms and their applications,” IEEE signal processing magazine, vol. 13, no. 6, pp. 22–37, 1996.

[23]. Kennedy J and Eberhart R, “Particle swarm optimization,” in Proceedings of ICNN’95-International Conference on Neural Networks, vol. 4, pp. 1942–1948, IEEE, 1995.

[24]. Toktas A and Ustun D, “A triple-objective optimization scheme using butterfly-integrated abc algorithm for design of multi-layer ram,” IEEE Transactions on Antennas and Propagation, 2020.

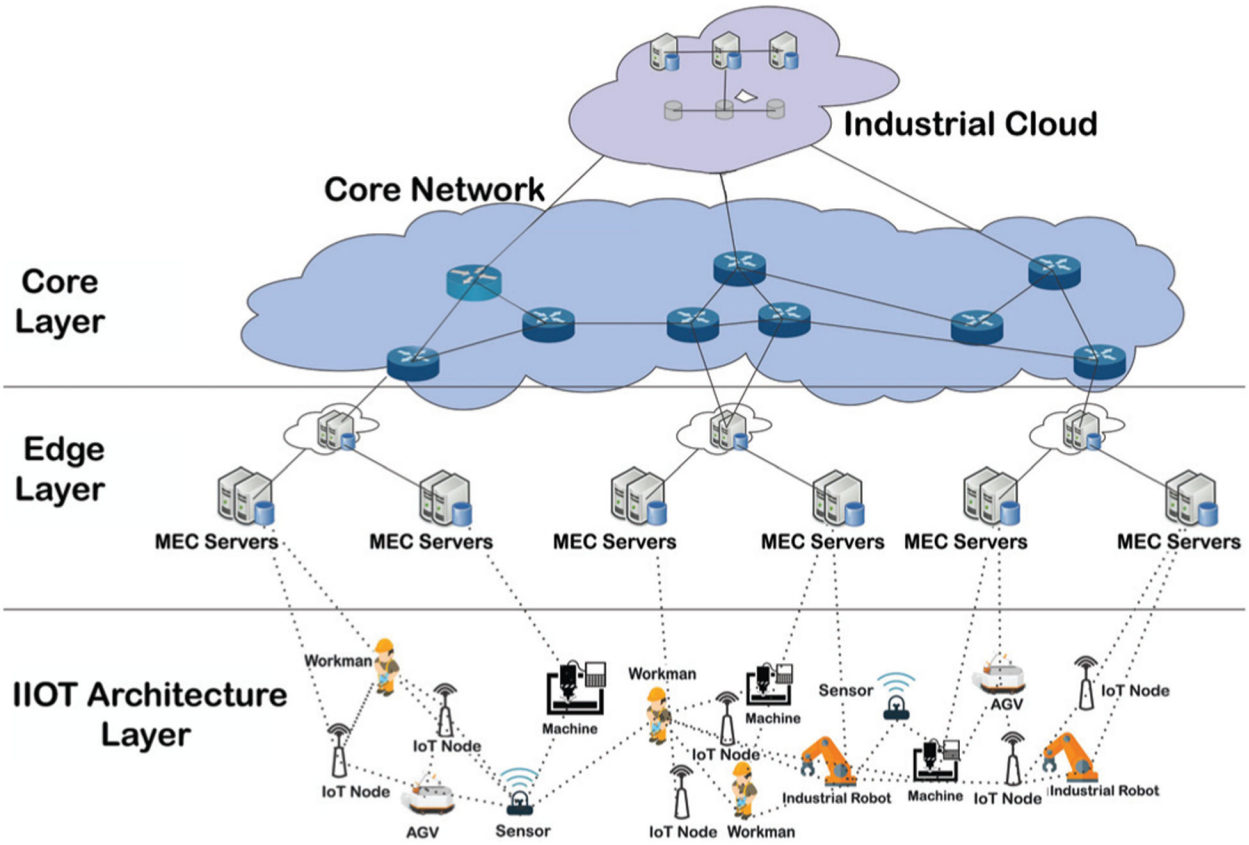

Fig. 1:
System model of the proposed scheme

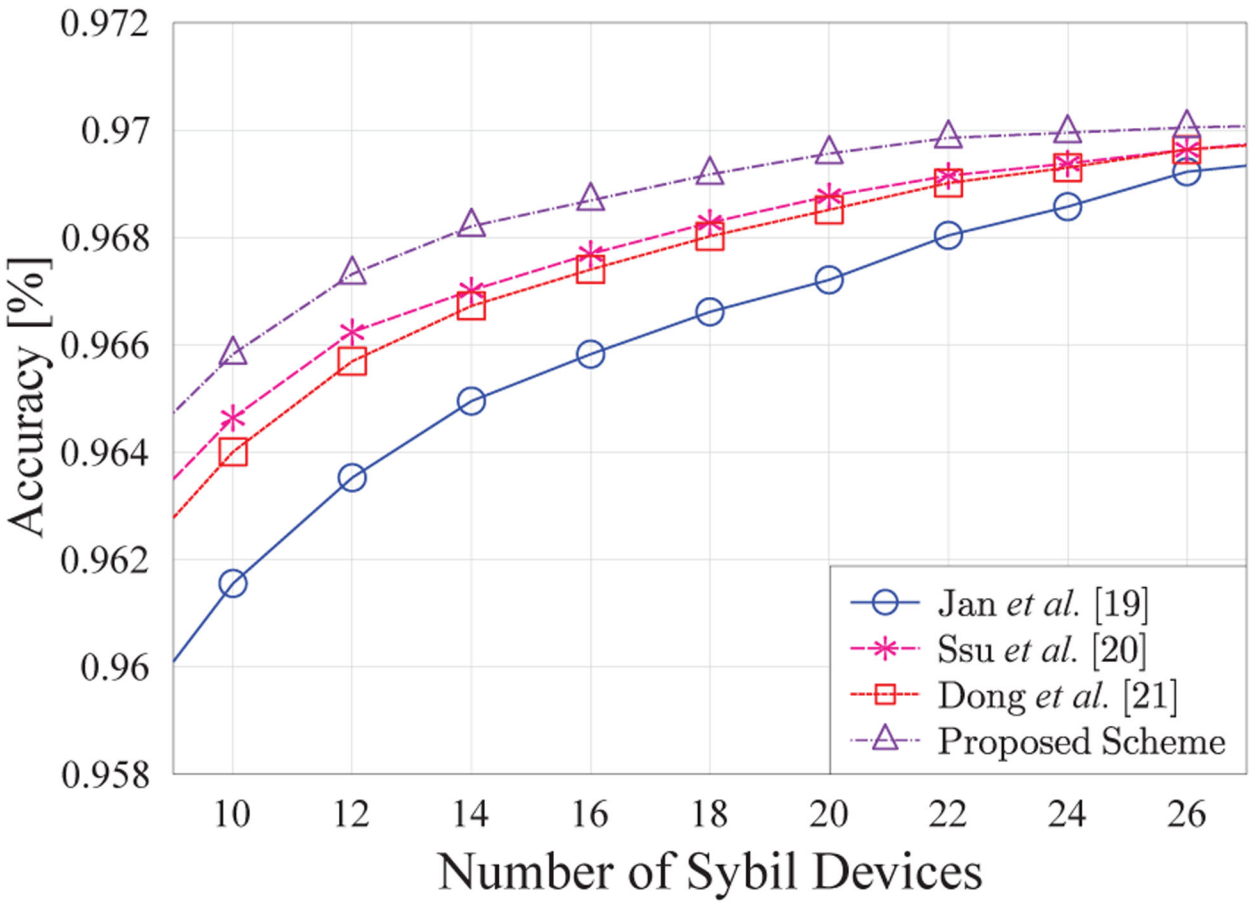

Fig. 2:
Accurate detection of Sybil devices

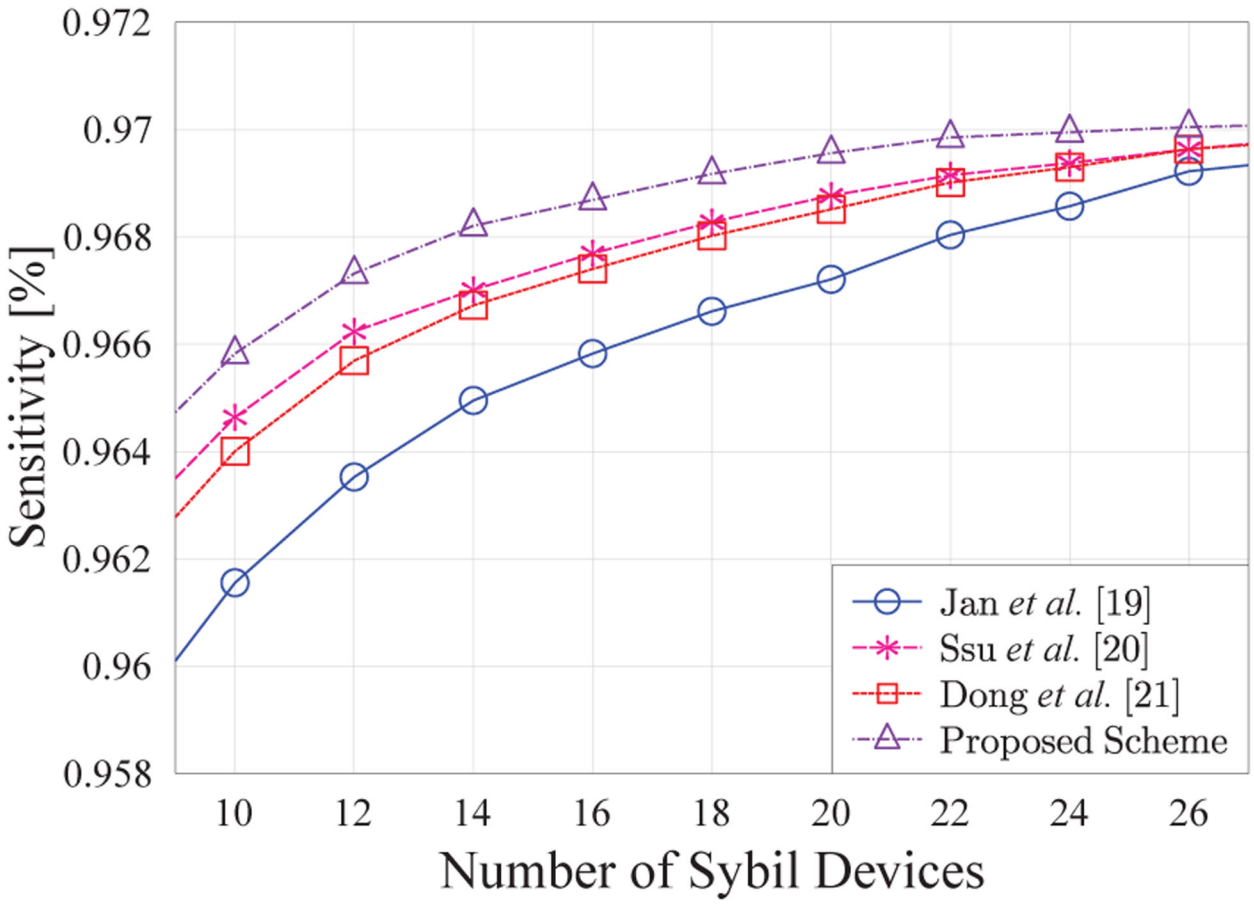

Fig. 3:
Sensitivity of Sybil devices detection

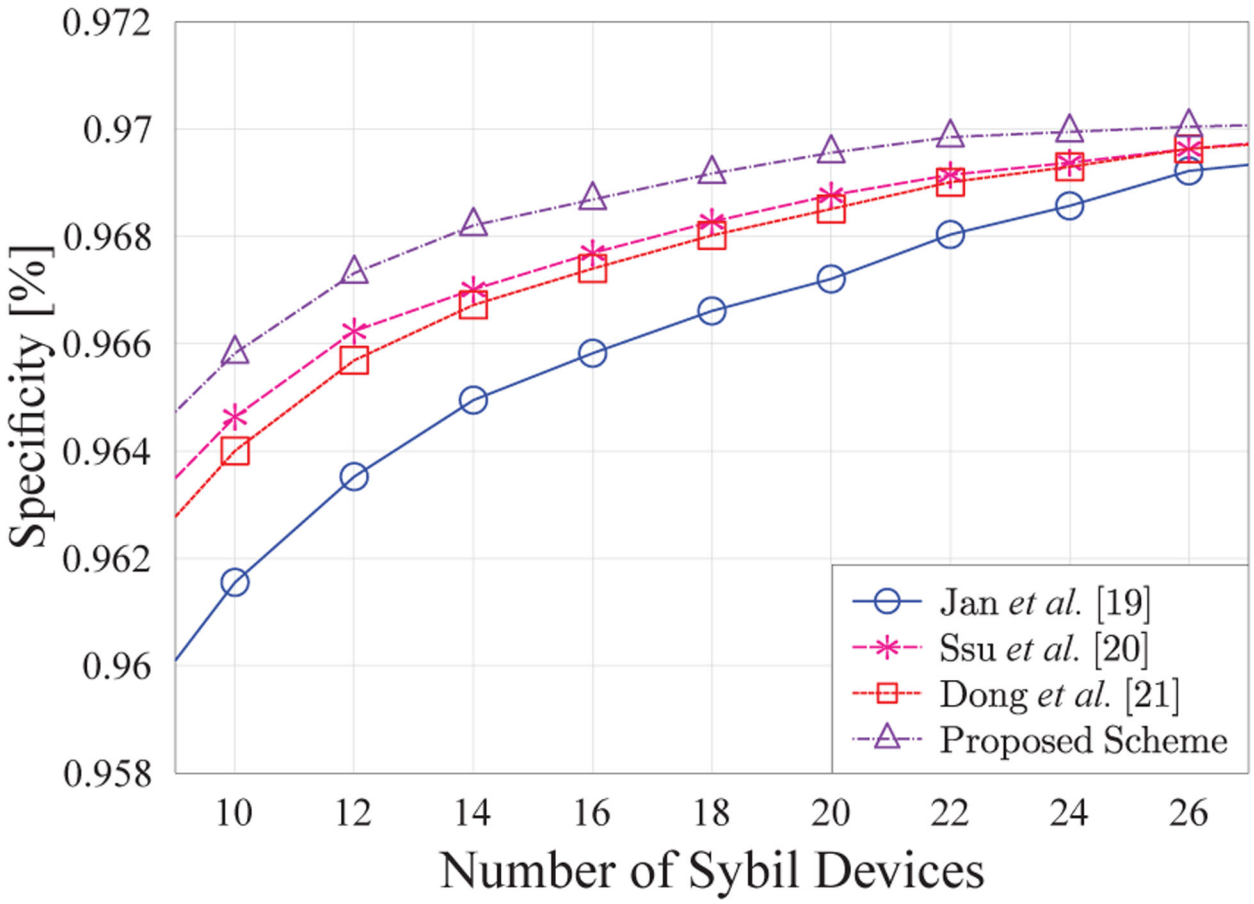

Fig. 4:
Specificity of Sybil devices detection

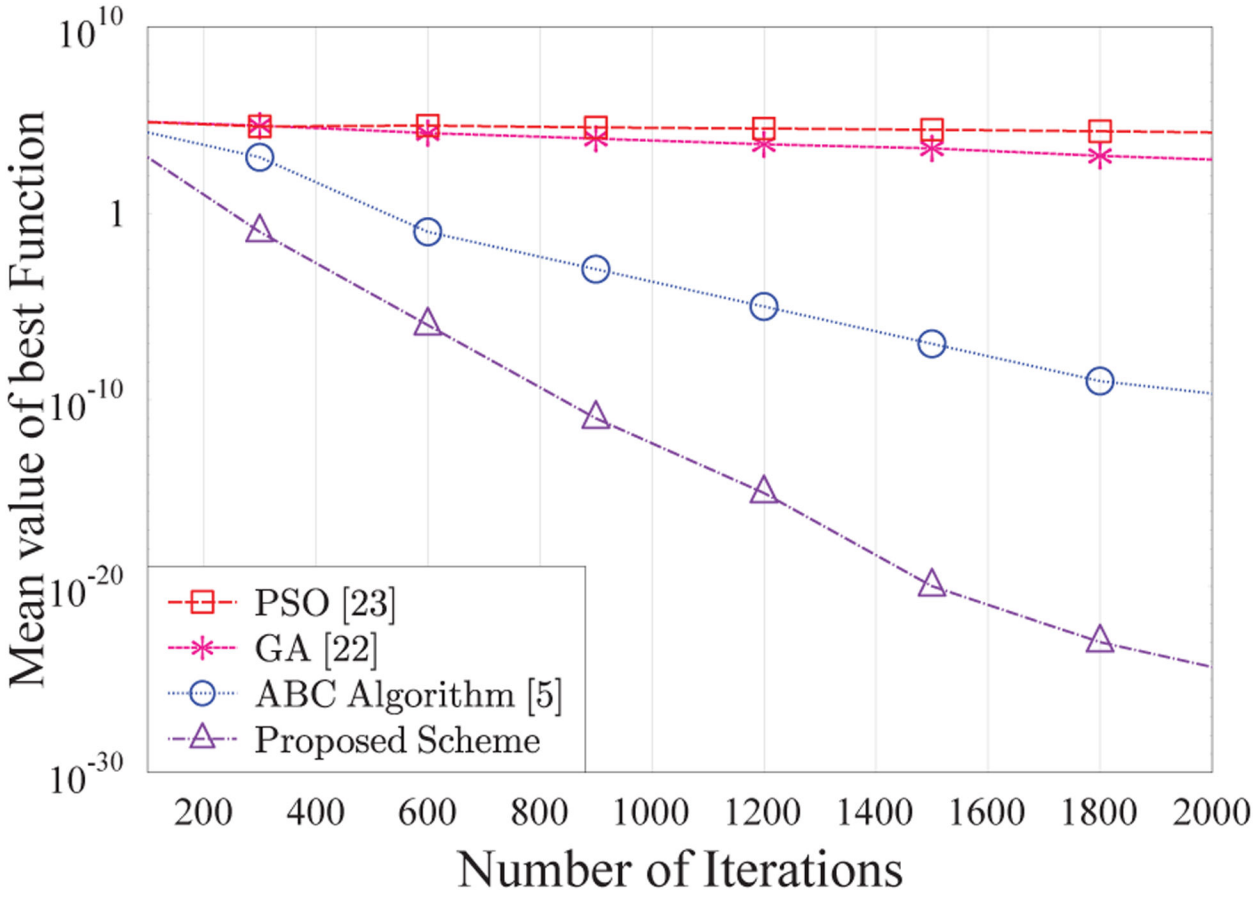

Fig. 5:
Convergence curves the Sphere function

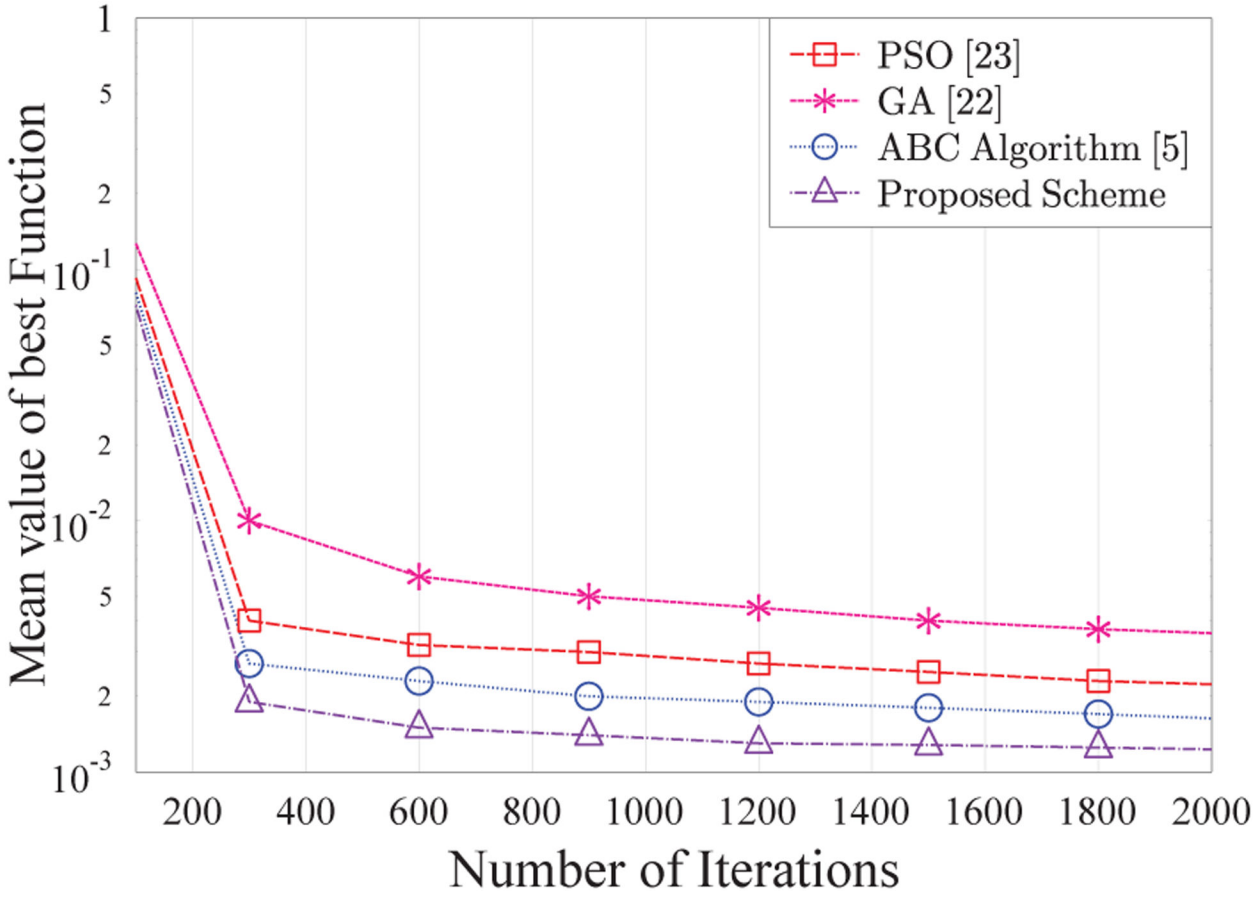

Fig. 6:
Convergence curves for the Schaffer function

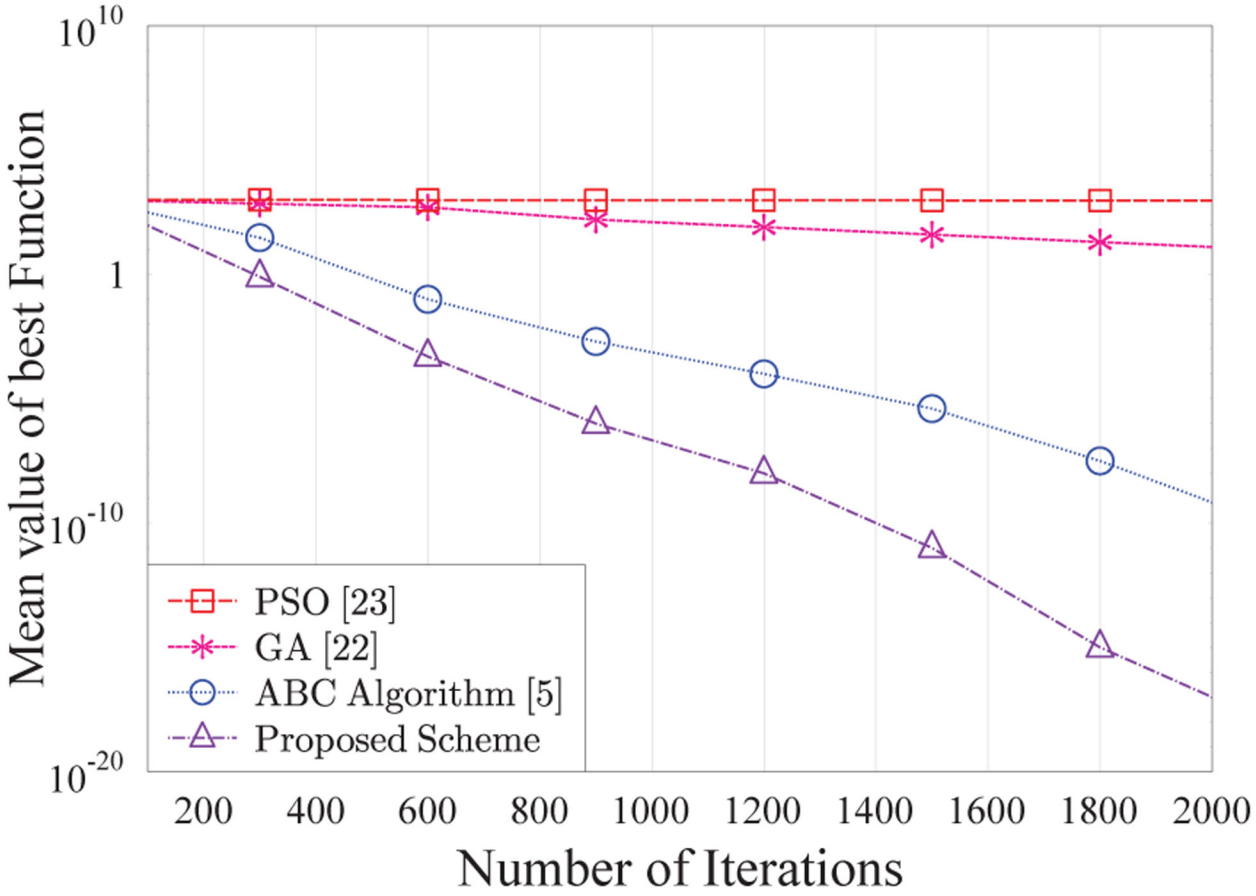

Fig. 7:
Convergence curves for the Griewank function

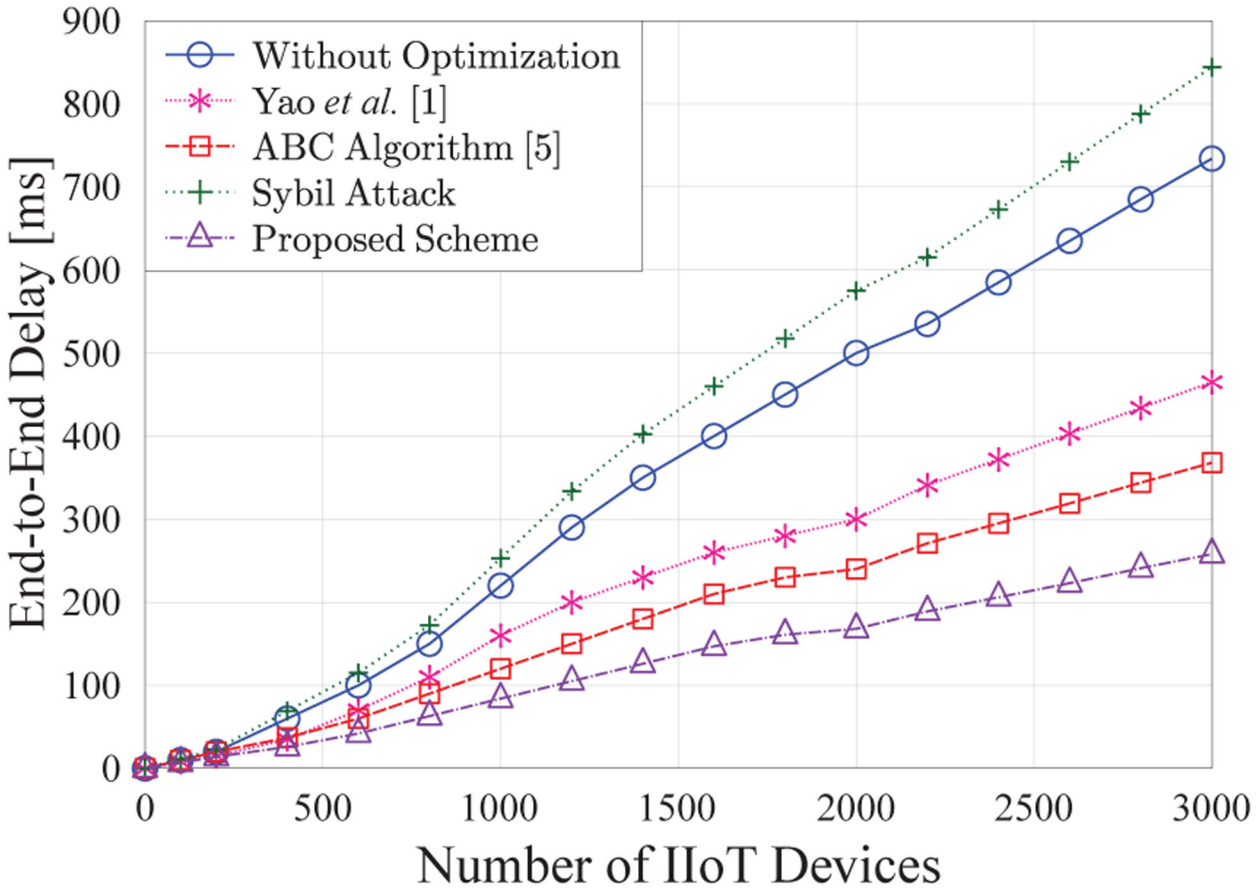

Fig. 8:
End-to-End Delay vs. Number of IIoT Devices

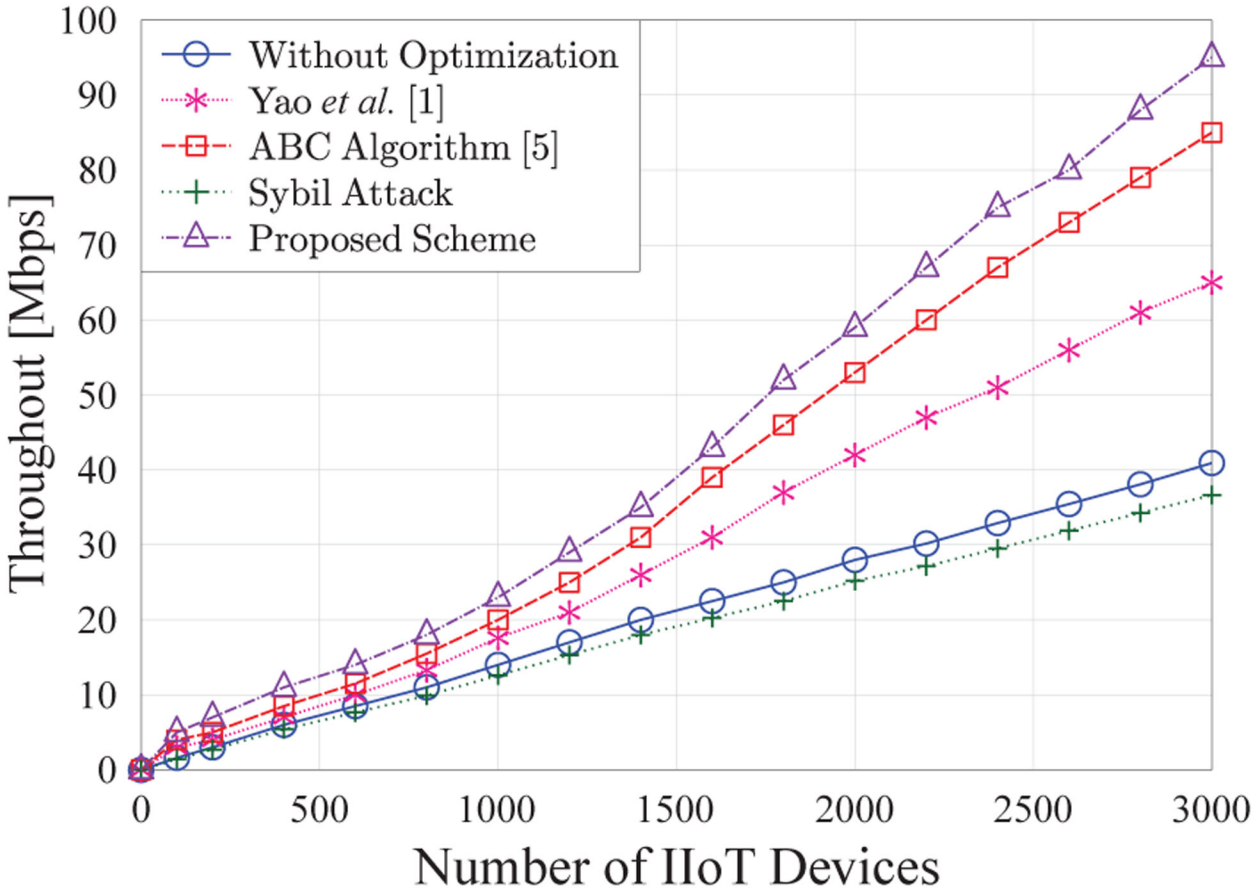

Fig. 9:
Throughput vs. Number of IIoT Devices.

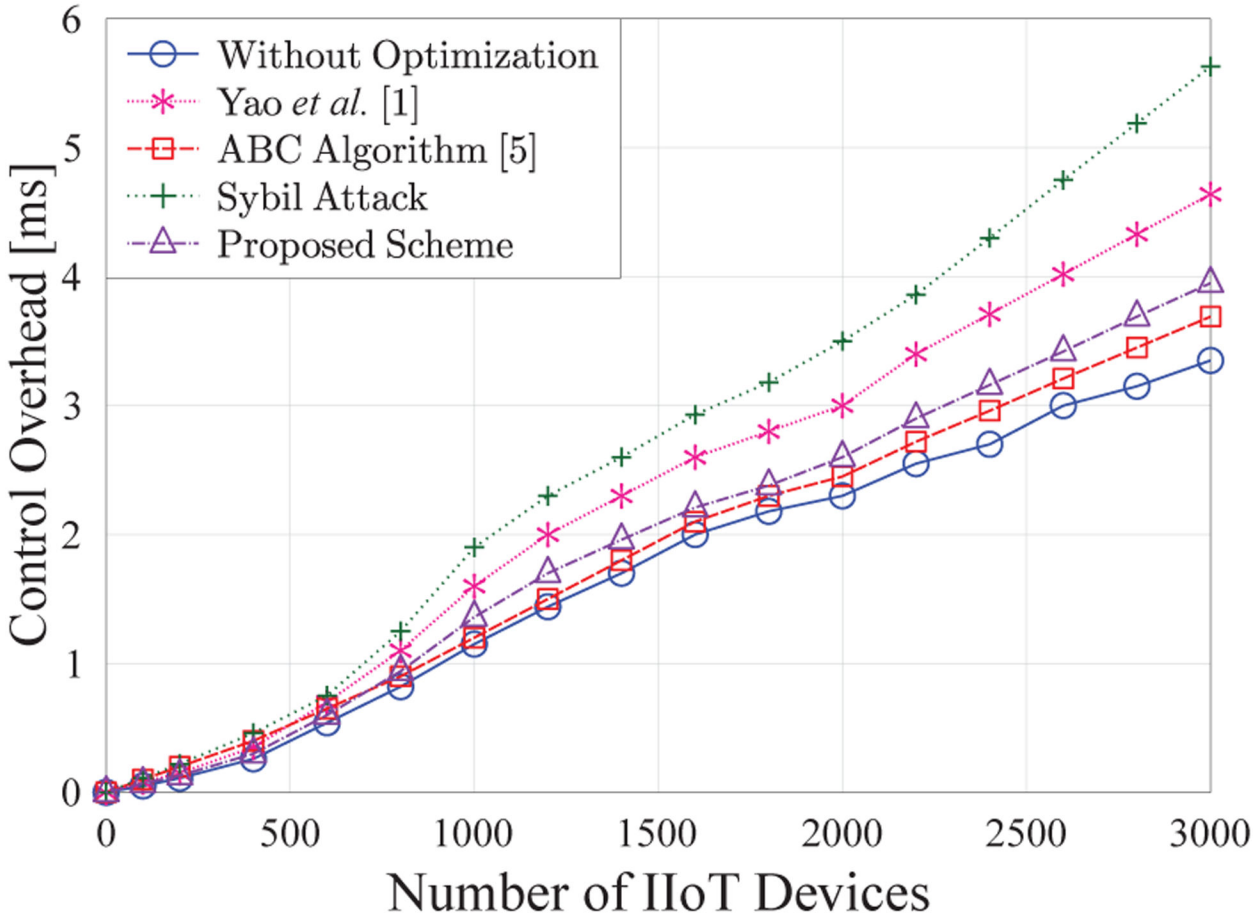

Fig. 10:
Control Overhead vs. Number of IIoT Devices

TABLE 1:

Honey collection vs optimization problems

Bees collect honey	**Optimization problem**
Proximity of the food source	Possible solution
Abundance of the food	Number of solutions
Ease of extraction	Easiness in getting a possible solution
Quality of the food	Fitness value
Maximum quality food	Optimum solution

TABLE 2:

Confusion Matrix

	Positive	**Negative**
Positive	T_P	F_N
Negative	F_P	T_N

TABLE 3:

Benchmark Functions

Name	Formula	Range
Sphere function	$f(X) = \sum_{j=1}^{D} X_j^2$	[–100,100]
Griewank function	$f(X) = \frac{1}{4000}\sum_{j=1}^{D}(X_j - 100)^2 - \Pi_{j=1}^{D} cos\left(\frac{X_j = 100}{\frac{1}{j\overline{2}}}\right) + 1$	[–600,600]
Schafer function	$f(X) = \frac{1}{2} + \left(\left(sin\left(\sum_{j=1}^{D} X_j^2\right)\right)^2 - \frac{1}{2}\right)/\left(1 + \frac{1}{1000}\left(\sum_{j=1}^{D} X_j^2\right)\right)^2$	[–100,100]