IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2021, accepted March 5, 2021, date of publication March 10, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065293

REFIT: Robustness Enhancement Against Cascading Failure in IoT Networks

MORTEZA BIABANI[1], NASSER YAZDANI[1], AND HOSSEIN FOTOUHI[2]
[1]School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14395-515, Iran
[2]School of Innovation, Design and Engineering, Mälardalen University, 721 23 Västerås, Sweden

Corresponding authors: Morteza Biabani (mbiabani@ut.ac.ir) and Hossein Fotouhi (hossein.fotouhi@mdh.se)

This work was supported in part by the Swedish Research Council (Vetenskapsrådet) through the MobiFog Starting Grant, in part by the Swedish Knowledge Foundation (KKS) through the FlexiHealth Prospekt, and in part by the EU Celtic_Next/Vinnova Project-Health5G (Future eHealth powered by 5G).

ABSTRACT There has been tremendous growth in the Internet of Things (IoT) technologies, and many new applications have emerged. However, cascading failure as one of the major issues in such constrained networks have been neglected. In this paper, we apply an effective clustering approach dubbed as REFIT to enhance network topology robustness via nodes' residual energy. The REFIT protocol divides the network processes into two stages, (i) set-up state and (ii) steady state. The Cluster Head (CH) selection method determines the supreme set of CHs that balances load distribution. The routing method is developed with the modified Particle Swarm Optimization (PSO) algorithm and the objective function to find the supreme set of Relay Nodes (RNs). These complete methods are combined into a set-up state to construct an optimal routing tree that links these CHs to the sink via RNs. In steady state, we model the routing tree to Conditional Directed Acyclic Graph (C-DAG) infrastructure that leads to shortcut routes. Simulation results on MATLAB Simulink have demonstrated that compared with the state-of-the-art works, REFIT can significantly promote network robustness against cascading failure.

INDEX TERMS IoT, cascading failure, robustness, clustering, particle swarm optimization, fault tolerance.

I. INTRODUCTION

Internet of Things (IoT) has become a platform for advanced solutions to the challenges of modern emerging technologies [1]–[3]. Wireless Sensor Network (WSN) is a distributed communication system [4], which is considered as a core component of the IoT network that has been widely exploited in many application domains including smart power grid [5], autonomous vehicles [6], disaster management [7], healthcare systems [8], etc. Wireless sensors gather environmental data autonomously and send them to a destination. In practical IoT networks, due to the low cost of hardware, sensor nodes generally have a bounded capacity [9], [10]. If the traffic load of the sensors exceeds their buffer capacity, the probability of their failure due to the buffer overflow increases [11]. When a failure occurs in one node, the other nodes will have to choose new routes to transmit data, so it is common to change the current routing paths. The re-routing process for redistribution load may cause some other nodes to fail due to the excessive traffic. This process may continue in the network, and the failure cycle can be expanded throughout the network [12]–[16]. This failure is considered as one of the essential items affecting the robustness of IoT networks, which is called *cascading failure* [17]–[19].

One of the obvious examples of cascading failure is the blackout in Indian power grids in 2012, when three regional grids collapsed, affecting over 400 million people [20]. A microgrid as an instance of a power grid is vulnerable to abnormalities due to small inherent inertia. Consequently, the stability of microgrids becomes a challenge after disrupting power systems. Thus, microgrids must be more reliable and more adaptive to load redistribution of the power grid [5], [20]. In other applications, the risk of smart city security increases by disrupting important infrastructure and damaging public trust. Most importantly, breakdowns in interconnected data-driven services can lead to system inefficiencies in a cascading manner [21]. In smart factories, information network components are vulnerable to attack and error due to numerous connections. Complex dependencies on physical devices and information networks complicates the detection of threats. These threats can cause cascading damage to the smart network of the factory [22].

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

In this work, we are focusing on microgrid applications as one of the most critical IoT applications that could be affected by cascading failure [23]. In general, IoT network is a proper solution for microgrid applications in order to manage energy and traffic load, especially when microgrids are equipped with renewable sources of energy [17], [18], [23]. These resources are usually located in harsh environments and complex places. Nodes in microgrid may fail due to several reasons, such as invader attack, hardware failure, and termination of energy [7], [24], [25]. Considering the cause of cascading failure, network traffic will meet certain challenges such as segmentation or paralyzation [13]. Moreover, there are still relatively few studies on how to optimize network topology against cascading failure [18].

In this paper, we consider combination of the clustering method with the Particle Swarm Optimization (PSO) algorithm for energy-efficient routing in IoT-based WSNs [7], [26]. The clustering method is the most common technique in controlling the energy restriction of nodes. This method divides the whole network into specific groups called clusters, in each cluster a leader called Cluster Head (CH) is selected to collect data from nodes in its cluster, known as Cluster Members (CMs), and transfer the data to the sink [9], [27]. CH selection is an NP-hard optimization problem [26]. PSO as an evolutionary algorithm with the help of multi-objective function seeks to find near-optimal solutions for CH selection [27]. In most studies, various objectives have been considered in CH selection, which is the key notice to the optimal trade-off between these factors [26]–[28]. However, the above-mentioned works have not paid attention to optimizing the hierarchical structure of clustering methods in improving the network resistance against cascading failure, which we try to utilize this energy-efficient topology to promote the network robustness.

We propose REFIT (Robustness Enhancement against cascading Failure in IoT networks), a fault tolerance mechanism to minimize the effect of cascading failure in IoT networks. On this basis, REFIT divides the network operations into two phases, (i) set-up state and (ii) steady-state. For clustering, we integrate important parameters affecting fitness function due to the supreme set of CH selection which includes energy, distance, delay, coverage, and node degree parameters. For routing, in a muli-hop communication, we apply the PSO algorithm as proven to be a promising optimization approach that causes clusters to follow multi-hop communication to reduce power consumption. Available energy in CHs, number of shortest paths and cluster size are three important factors that are integrated into the PSO fitness function. The clustering and routing are combined into a set-up state to construct an optimal routing tree that links these CHs to the sink. In the steady-state, we reduce the effect of cascading failure on the routing tree. In this phase, we first model the routing tree into Conditional Directed Acyclic Graph (C-DAG) infrastructure [29], and then propose a fault-tolerance mechanism by selecting effective paths based on the maximum residual energy. Finally, we simulate and evaluate the proposed method under the robustness and performance metrics in MATLAB Simulink. Extensive simulation results demonstrate that the proposed REFIT protocol not only maximizes the network lifetime of IoT applications but also effectively promote network robustness in the face of cascading failure.

The key contributions of this work are summarized as follows:

- Devised a fault tolerance mechanism for IoT networks to reduce the effect of cascading failure.
- Proposed a centralized CH selection to characterize the best set of CHs.
- Designed a novel multi-hop multi-objective evolutionary algorithm for routing tree construction to characterize the best set of RNs.
- Modeled C-DAG infrastructure for routing tree to enhance the topology robustness.
- Conducted extensive experiments to verify the correctness of the proposed algorithms and models.

The rest of the paper is arranged as follows. Related works are discussed in Section 2. The problem statement for IoT application is discussed in Section 3. Our proposed approach against cascading failure is presented in Section 4. The experimental results and discussions are presented in Section 5. Finally, we conclude the paper in Section 6.

II. RELATED WORK

With the increasing number of IoT-connected devices and consequently, additional network topology complexity, cascading failure has become more evident in IoT applications [17]. In this section, we first look into some of the related works in the failure effect in IoT networks, and then we specifically elaborate some of the solutions in the power grid domain as it is our targeting application domain in this paper.

A. RELATED WORKS ON IoT FAILURE EFFECTS

Fu *et al.* [18] designed a sink-oriented cascading model for WSNs based on a memetic algorithm to increase the robustness of network topology, which is so-called MA-TOSCA. In addition, this work achieved a stronger topological structure to spend less time than existing algorithms. In MA-TOSCA, modularity and clustering features have a positive effect on network robustness compared to the average shortest path length. Bao *et al.* [14] investigated load entropy that can be a measure of network heterogeneity in load distribution. This work claimed that load entropy can be optimized as an indicator to control and defend cascading failures in many complex real-life networks. Fu and Yang [17] proposed a realistic multi-sink-oriented cascade model for WSNs. The authors focused on the resistance of the network topology against node-link attacks. However, they also discussed that node attacks will cause more fragile failures than link attacks.

Zhong *et al.* [16] developed a generic method for examining network endurance with cascading overload failure in WSNs. In this paper, endurance refers to the duration time for the traffic load to touch a critical point with

a cascading failure. They also concluded that endurance is strongly dependent on the intensity of the cascading and disturbance. Moreover, the network endurance increases uniformly for uniform primary load distribution while for other types of load distributions, it shows more complex resistance behaviors. Yin *et al.* [25] modeled a fault-tolerance mechanism based on overloaded cascading failures in free-scale WSNs based on load and capacity. This work studied the relationship between variable load and cascading failure in a scale-free network. It concluded that when the load is more than a critical point, a random point failure in scale-free topology leads the entire network to collapse.

B. RELATED WORKS ON CASCADING FAILURE IN POWER GRIDS

Zhang *et al.* [20] considered new theories of stability in power grid systems. This study, which was inspired after India's major blackouts, states that if the load adjustable parameter grows, the probability of large cascading failure is more than a small or medium cascade failure. Adnan *et al.* [5] analyzed the cascading failure model in smart power grid. This work focused on reducing the uncertainty of the power grid due to the penetration of renewable energies. Although they provide optimal load balance with increasing transient stability in topology, the impact on power network is still unpredictable. In this study, a probabilistic analysis has been performed to prevent the spread of excessive cascade failures from one group to another group. Shuvro *et al.* [30] utilized a machine learning approach in power grid application to predict the cascading failure problems. They classified cascading failure in three main classes of (i) large, (ii) small, and (iii) no cascades. In this study, the authors also used linear regression to predict the quantity of missed transmission channels and the amount of load reduction. Adnan *et al.* [31] implemented an algorithm against cascading failure without loss occurrence on a standard IEEE-30 bus system for power networks. This study is based on fuzzy logic and uses vehicles to grid technology. It employs a mathematical mixture to identify critical points in heuristically and energy-based process. In this paper, to increase computational velocity, a network operator by a self-broadcast graph holds only vulnerable nodes.

Dui *et al.* [32] examined a multi-strategy evolutionary game-based cascading model for scale-free networks. Due to the game strategy, they analyzed the corresponding network against cascading failure by removing the failure nodes. Their ternary strategy, which is based on various law enforcement, demonstrates that with small network collaborators, network robustness will be improved. Zhai *et al.* [33] developed a mathematical model to calculate the worst case cascading failure in power grids. They also used iterative algorithms to explore worst-case outline. One of the main advantages of this formulation is the ability to identify disruptive turbulence. Wu *et al.* [13] proposed a sequential recovery method against cascading failure in complex networks that considers both the performance mechanisms of complex carrying networks and the possible cascading failures during the recovery manner. This study considers the only recovery after a major blackout in grids. Moreover, they utilized new graph-based recovery tools to identify critical points that by examining them sequentially, they would improve the recovery process.

To the best of our knowledge, existing works on cascading failure do not take into account the residual energy of non-failure nodes. Using this parameter, we return links to nodes with maximum residual energy that reduce the effect of the traffic overload. Furthermore, the relationship between network topology and node energy problem is still rarely investigated, which is a challenge for optimizing the robustness of cascading failure.

III. PROBLEM STATEMENT

Assuming that the real IoT network is made up of a set of nodes and a sink to collect the measured data, meaning that the ending hop of the routing protocol reaches the sink node [2], [3], [7]. Relay nodes may fail due to any reason when transmitting the collected data to the sink [7], [34]. This effect can be seen in Figure 1 when node Y fails, the routing traffic load is distributed on node X. Meanwhile, node X is also failed due to the traffic load of node Z. As shown in Figure 1, after a while, many network communication links are failed. This action is known as *cascading failure*, which leads to network segmentation [13], [17]–[19]. In [17], [20], the authors have modeled cascading failure where the main reason for network shutdown is the overloaded traffic. The load indicates the amount of current traffic that can be transferred by nodes. To estimate the traffic load, we assume that the Shortest Path (SP) algorithm is applied if the packets need to be re-transmit between two nodes in the network. Hence, the load on node i at time τ ($\ell_i(\tau)$) is the total number of SPs crossing via node i. To describe the real load propagation of WSNs, the design model of this metric is as follows:

$$\ell_i(\tau) = \left(\frac{\sum_{j \in \Psi, k \in \Psi^*} {}^{\Omega_{i,j,k}(\tau)}/_{\Omega_{j,k}(\tau)}}{N} \right)^{\sigma} \tag{1}$$

Here, parameter σ is the load-coefficient ($\sigma \geq 0$), $\Omega_{i,j,k}(\tau)$ is the total number of shortest paths from node j to node k by passing via node i at time τ. Similarly, $\Omega_{j,k}(\tau)$ is the total number of shortest paths from node j to node k at time τ, Ψ is the set of nodes from N sensor nodes and Ψ^* is the set of sink nodes, which in this work we have one sink node. Accordingly, the capacity can be defined as the maximum load maintained by a node. The traffic load on nodes in a network may vary over time. If these changes increment and overpass the capacity, a particular node will be prone to a malfunction. As a result, more nodes may be failed during load redistribution [11], [16], [32]. The Capacity formula is expressed as

$$\mathbb{C}_N = (1+\delta)\ell_N(0) = (1+\delta)\frac{\sum_{i=1}^{N} \ell_i(0)}{N} \tag{2}$$

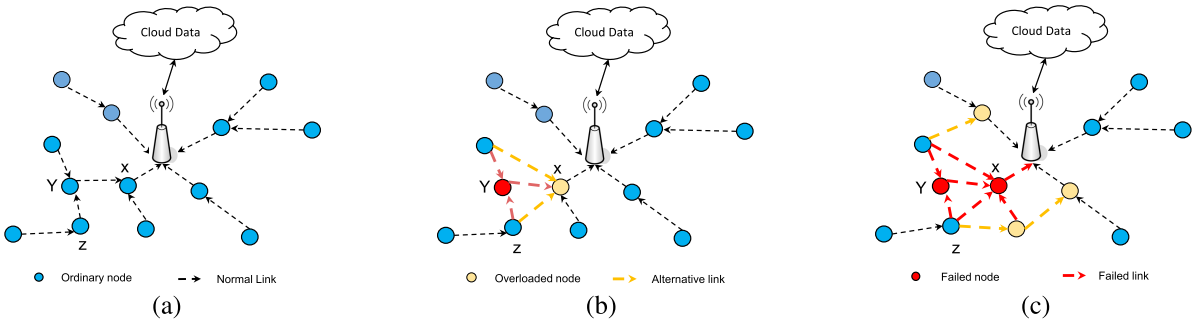

FIGURE 1. Cascading failure propagation process: a) normal state of the application without any node failure; b) the situation where node Y fails, the routing traffic load is converted to node X; c) the situation where Node X also fails due to the excessive traffic reaching from node Z. Thus this could lead to network collapse.

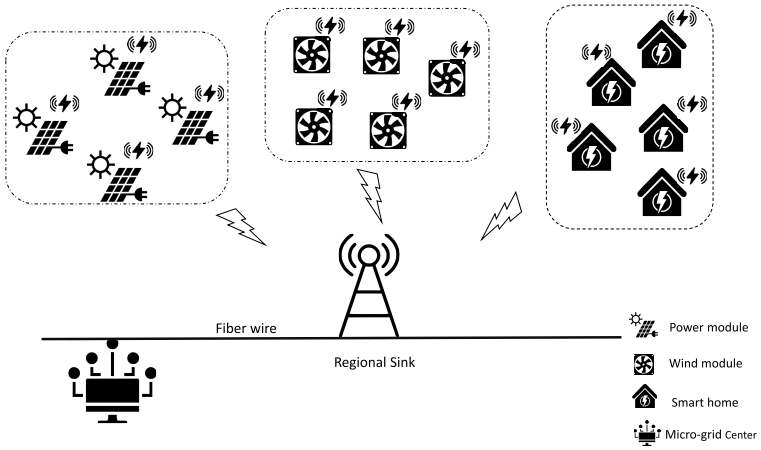

FIGURE 2. Our IoT-microgrid structure with renewable energy resources.

Here, N is the total number of nodes starting from a node with index i = 1, and $\ell_N(0)$ means that we assume the capacity of a node with its initial load ($\ell(0)$). Capacity ($\mathbb{C}$) in Equation 2 states that any small changes in overload-coefficient ($\delta \geq 0$) which leads to a change of traffic load will bring the violation of capacity. In our IoT application, a microgrid is an autonomous system that can control and manage itself [23]. Sometimes in an energy microgrid, consumers receive heat energy for heating and cooling in addition to electricity. One of the basic characteristics of microgrids is that they can work in two modes: (i) connected to a network or (ii) an island (separate from the network) [35]. When connected to the grid, the microgrid exchanges electrical energy with the grid. It can be seen in Figure 2, IoT-microgrid application with renewable energy resources is utilized in different levels. Microgrid resources may be depleted for reasons such as attackers, hardware failure, and power outages [5]. This failure, however small, may lead to the collapse of the entire network [20]. Further, Microgrids are vulnerable to abnormalities due to small inherent inertia. Therefore, the stability of microgrids becomes a challenge after disrupting power systems [10], [23].

Based on the discussions in the previous two sections, we conclude the following two problem declarations:

- How to provide an energy-efficient forwarding data to sink over cascading failure?
- How to develop a multi-hop communication in constrained networks such as microgrids for a more robust topology against cascading failure?

IV. REFIT APPROACH

In this section, we propose our load-balancing and energy-efficient routing method with respect to cascading failures (REFIT). This method divides the network operations into two stages, (i) set-up state and (ii) steady state. These stages are performed in series until the network lifetime ends. The set-up state itself includes three steps of: (i) bootstrapping, (ii) clustering, and (iii) routing. Figure 3 shows the stages of the proposed protocol. In fact, we address the issues of how to forward energy-efficient data in the set-up state and to develop a more robust topology against cascading failure in the steady state.

After distributing nodes in the network, all sensor nodes start the bootstrapping process. This part involves two processes. The first process is Neighbor Discovery (ND), where each sensor broadcasts a beacon frame consisting its identification (ID) and position. After the ND process, each sensor node knows its neighboring sensors. In the second process of the bootstrapping, the local data of each sensor

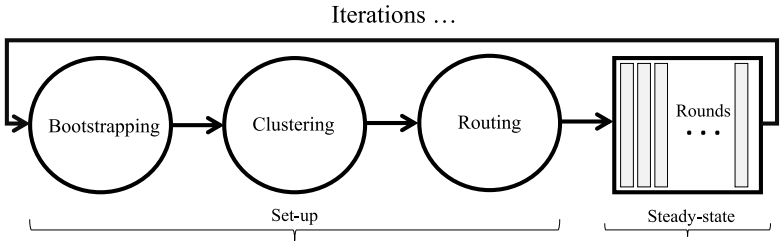

FIGURE 3. Overall schema of REFIT.

is transmitted to the sink by a *Hello* packet. Each sensor receiving a *Hello* packet behaves in the same manner. Then, the sink node executes the set-up state including clustering and routing methods in which it utilizes the data it delivers about the Area Of Interest (AOI). The sink node with the help of a centralized CH selection procedure constructs clusters. The modified PSO-based routing algorithm characterizes the relay node (RN) of each node (next-hop) towards the sink. After terminating the set-up state, the sink node broadcasts its outcomes by a controlled-flooding scheme, which makes each sensor not transmit a packet if it has already transmitted. Next, all the sensor nodes experience the steady-state, which is composed of multiple rounds. In each round, data transmission to the sink by RNs is intelligently implemented to balance network traffic load against cascading failure. The pseudo-code of the overall proposed protocol is given in Algorithm 1.

Algorithm 1 Overall Proposed Protocol

```
procedure REFIT( )
    Initialization
    for each iteration i do
        Bootstrapping
        CH Selection (Nodes, Net_info)                ▷ Section IV-B1
        Cluster formation (CHs, Nodes)                ▷ Section IV-B2
        Routing (Clusters, Net_info)                  ▷ Using Algorithm 2
        Broadcasting (CHs, RNs, Routing Tree)
        for each round j do                           ▷ Section IV-C
            Steady State(i,j)                         ▷ Using Algorithm 3
        end for
    end for
end procedure
```

After developing energy model, we describe the proposed clustering and routing algorithms in the next two subsections. Next, the steady state will be explained.

A. ENERGY MODEL

The current state of energy models in IoT networks is configured with the Heinzelman model [36]. This model indicates the energy consumption on the basis of sent bit length (L) and the distance of transmitter and receiver node in Equations 3–4. In addition, this model states that if the transmission distance (d) of Equation 3 is more than the threshold distance of Equation 5, energy is consumed fourfold, otherwise doubled. The features of antenna amplifiers are given by ε_{amp} and ε_{efs}, where ε_{amp} represents the energy loss for the multi-path model (Power dissipation d^4) and ε_{efs} is the energy model of free space (Power dissipation d^2). Also, E_{elec} indicates the energy utilized to activate the circuit of the transmitter and receiver.

$$\mathrm{E_{Tx}(L,d)} = \begin{cases} L.\left(\mathrm{E_{elec}}\right) + L.\left(\varepsilon_{\text{efs}} \cdot \mathrm{d}^2\right), & \text{if} \quad d \le \mathrm{d_0} \\ L.\left(\mathrm{E_{elec}}\right) + L.\left(\varepsilon_{\text{amp}} \cdot \mathrm{d}^4\right), & \text{if} \quad d > \mathrm{d_0} \end{cases} \tag{3}$$

$$\mathrm{E_{Rx}} = \mathrm{L.(E_{elec})} \tag{4}$$

$$d_0 = \left(\varepsilon_{efs}/\varepsilon_{amp}\right)^{1/2} \tag{5}$$

In our application, we need to have proper energy assessments based on the number of hops leading to the sink. The multi-hop routing delivers packets hop by hop from the source node to the destination. In cascade failure scenario, if a next hope of a route fails, selecting a successor link will be a critical decision as it should optimizes network traffic congestion. It is not trivial to simply calculate the amount of energy for transmitting a specific packet size from a source node to a destination. Consecutive calculations at this level and increasing the number of hops will cause our network to be inefficient in terms of energy and cost. In this section, we improve the radio-energy model [36] based on our own scenario. Since the proposed method is centralized and the sink knows the position of the nodes, we model the problem in the worst-case scenario by formulating the energy consumption for the longest path. After selecting the longest path, which will be described in the steady-state phase, if we set the maximum number of hops in the longest path and the distance between nodes in multi-path network –C and D parameters respectively– then the predicted amount of required energy of transmitting and receiving packets from a node to the sink for L bit data is computed as follows:

$$E_{TXnew} = \sum_{c=1}^{C} \left(L.(E_{elec}) + L.(\varepsilon.D)\right) \tag{6}$$

Subject to: $\forall c \in C$ *if* $(Cost\_Edge(c) \leq d_0) \rightarrow \varepsilon.D = \varepsilon_{efs}.d^2$ *else* $\varepsilon_{amp}.d^4$.

$$E_{RXnew} = \sum_{c=1}^{C} (L.(E_{elec})) \tag{7}$$

Thus, the total energy consumption of the network will be represented as follows:

$$E_{Total} = 2\sum_{c=1}^{C} L.(E_{elec}) + \sum_{c=1}^{C} L.(\varepsilon.D) \tag{8}$$

If we model this energy from collecting data by CHs, and transmit it to the sink, we will obtain the following formulas:

$$E_{CH\_agg} = Z.L.(E_{data\_agg}) \tag{9}$$

$$E_{Total\_CH} = E_{CH\_agg} + E_{Total} \tag{10}$$

where Z is the number of packets, E_{data_agg} is the energy consumed in data aggregation of 1-bit data, and E_{Total_CH} is the total energy of network from a CH to the sink, assuming the computation of data aggregation energy by CH.

B. CENTRALIZED PSO-BASED MULTI-HOP CLUSTERING AND ROUTING

This section addressed the proposed CH selection and routing protocol. The sink through the fitness function evaluates the quality of the solutions in the set-up state (CHs and RNs selection). The fitness function formula is the union of different efficiency factors that are combined to denote the maximum or minimum quality of a performance goal. Further, Multi-Criteria Decision Making (MCDM) formulas are one of the best tools for decision making [7]. In this paper, we employ the Adaptive Weighted Sum (AWS) approach to solve the multi-objective optimization problem (MOP) of clustering and routing [26], [27] which is expressed as Equation 11. The AWS method converted the MOP problem into a single optimization mathematical problem (SOMP) provided that all parameters have the same domain.

$$\min \sum_{i=1}^{N} \alpha_i.f_i \tag{11}$$

Here, $f_i : \Omega \rightarrow R^n, i \in \Omega, \alpha_i \geq 0, and \sum_{i=1}^{N} \alpha_i = 1$.

In general, the sub-objectives defined in the MOP formula are inconsistent. In such cases, related works indicate that there will be Pareto optimal solution. Under the convexity assumption, the solution of fitness function is Pareto optimal (if $\alpha_i > 0, \forall i = 1, 2, \ldots, N$)) [37]. However, such a combination of these different parameters to SOMP formula has been successfully tested, including [7], [27].

1) CH SELECTION

In general, CH is selected based on a number of parameters; including latency, power and distance. However, the proposed method is utilized for fault-tolerance, and thus to increase efficiency, coverage and degree of a node as two additional parameters are considered in the FP formula. Therefore, The proposed method not only maximizes the energy-efficiency for IoT systems, but also uniformly distributes CHs in the network to improve lifespan. In this paper, the sink selects CHs by considering different parameters together as shown in the Equations 12–14.

$$FF_1 = Power_f(1/Coverage_f) + Power_f(1/Degree_f) \tag{12}$$

$$FF_2 = \alpha(1/Distance_f) + (1-\beta)FF_1 \tag{13}$$

$$FF_3 = \alpha(FF_2) + (1-\alpha)(1/Latency_f) \tag{14}$$

Here, coefficients α and β are stationary values. Power and latency are factors that the sink can compute quickly for each node. Distance factor can also be achieved based on the Euclidean distance between nodes as shown in Equation 15.

$$Distance(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{15}$$

Now, the set of nodes with the lowest latency, the highest energy and the density are selected as the candidate CHs. To recovery fault, the two parameters (coverage, degree) make the best set of candidate CHs as the output of Equations 12–14, where these parameter are interpreted by using the following equations:

Node Degree: This parameter as shown in Equation 16 represents the number of sensor nodes accessible from a CH. It has been previously utilized for load balancing in CH selection [14].

$$Degree = \sum_{k=1}^{K} |Member(CH_k)| \tag{16}$$

Here, K is the number of candidate CHs and $|Member(CH_k)|$ is the number of cluster members in k_{th} CH.

Coverage: This parameter as shown in Equation 17 eliminates the non-CHs (non-clustered sensor nodes) and ensures participation of the remaining sensor nodes in the clustering process. Further, the nodes that have not joined any cluster will consume high energy in transmitting data to the sink node and thus it should be avoided [7], [26]. This parameter minimizes the number of remaining nodes that are unable to become part of any cluster. Thus, reducing the number of the non-CHs enhances network coverage considerably.

$$Coverage = \frac{N - K - Degree}{Degree} \tag{17}$$

After determining the sub-objective functions and forming the linear formulation as single objective function, the sink constructs clusters in a centralized method. The whole process of CH selection is shown in Figure 4.

2) CLUSTER CONSTRUCTION SCHEMA

After CHs selection, the sink node broadcasts a message that identifies the CH (ICH). The sensor node becomes CH with the same ICH, and neighbors as CH members, depending on the ICH, nodes will updates their values (Figure 4). Therefore, the construction of clusters extends in the system until

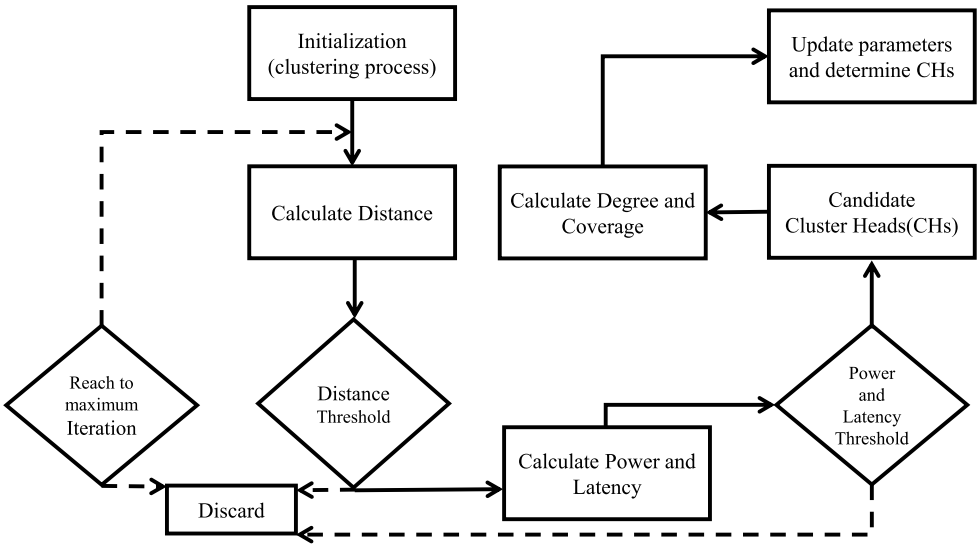

FIGURE 4. Flowchart of the proposed CH selection model.

each node reaches CH or cluster. Determining the optimal affiliation of cluster members is expressed as follows: (i) If the value of the neighboring node contains one CH, then join to that cluster. (ii) If the value of the neighboring node is higher than one CH, then join to CH which have maximum residual energy. Finally each Cluster member transmits the collected data to the CH using their TDMA slot. In TDMA, idle nodes are allowed to cross the sleep threshold in order to save energy [4], [9].

3) ROUTING TREE CONSTRUCTION SCHEMA

After cluster construction, initialization state by PSO starts and many particles are defined as solution. It should be noted that the particles are developed based on selected clusters. However, PSO receives a limited number of sensor nodes as the initial population. In addition, each solution is a set of a number of nodes in the cluster. In PSO, particles are evaluated by fitness functions and accordingly the best position value is assessed. Then, the best global value is specified among all of them. With the aim of these values, the position and velocity of the particle are updated. All this process is repeated until the termination status is reached. The output of this process will be routing with the source of the CHs and destination with the sink. Now the RNs must be found by PSO to transmit packets to sink hop by hop. The sub-objectives function to determine the optimal paths leading to the sink are interpreted as follows:

Available Energy in CHs: Fault inserts high traffic load to the network that raises the nodes' energy consumption. Therefore, the remained energy of a node can be a good factor to select RN [7]. The first fitness parameter that deals with energy efficiency is given by Equation 18.

$$En = \sum_{k=1}^{K} Residual.Energy(CH_k) \tag{18}$$

Number of Shortest Paths: Not to violate the capacity relation, it is necessary to control the network load. Therefore, when the failure occurs, the CH with the shortest path should send its information to the sink. The following normalized equation represents the network efficiency:

$$Pa = \frac{1}{N\,(N-1)} \sum_{k=1}^{K} \frac{1}{nsp\,(CH_k)} \tag{19}$$

In Equation 19, $nsp(CH_k)$ is the number of shortest paths from the CH to the sink. In fault acuteness situation, it is assumed when one node is annihilated and its path is not available. Thus, $0 \leq nsp \leq N-1$. It is evident that the lower the number of short paths between CHs and sink is the less robust the network becomes [20]. We use the Dijkstra's algorithm to detect the shortest path to access all nodes that drive exponential time $O(n^2)$ [11]. The shortest-path tree constructed can send data to the sink with a minimum number of hops. This criterion can also be called the factor of distance between CHs and sink.

Cluster Size: It is very important to keep the size of the cluster in the desired amount to avoid unbalanced energy consumption [27]. When the size of the cluster is large, the energy consumption by that cluster will be high. The third fitness parameter that deals with cluster size is given by Equation 20.

$$Si = \sum_{c=1}^{C} NumberOfNodes(Cluster_c) \tag{20}$$

Here, C is the total number of clusters. Cluster size and the shortest path criteria have a direct effect on load and capacity factors (Equation 1–2), so that by effectively controlling them, the fault tolerance mechanism can be improved. Finally, the fitness function for routing is estimated on the basis of stated criteria as follows:

$$F_{routing} = \alpha_1 \times En + \alpha_2 \times Pa + (1 - \alpha_1 - \alpha_2) \times Si \tag{21}$$

Here, $\alpha_1 + \alpha_2 = 1$ and also variables α_1 and α_2 are the weight parameters in Equation 21 and they are assessed and estimated during simulation based on AWS method. The fitness function represented by $F_{routing}$ in Equation 21 should be minimized to bring the network performance to the optimum value. The whole process of PSO applied in the proposed work is presented in Algorithm 2, which is discussed as follows:

Algorithm 2 The PSO-Based Multi-Hop Routing

Input: Clusters and Network information(ex. NoP)
Output: Optimal Routes

procedure MODIFIED PSO( )
  Initialize Particles ▷ Partition of set clusters with obtained positions and random velocity
  Define function Next-Hop()
  Set Number of particles: i = 1 to NoP
  for each particle *i* **of NoP do**
    Evaluate *fitness*(*particle*(*i*)) ▷ Using equation 21
    p_best = *particle*(*i*)
  end for
  g_best = min[*p_best* of all *particles*] ▷ minimum best-set of p_bests
  while maximum period or target objective is not attained do
    for each particle *i* **of NoP do**
      update position *particle*(*i*) ▷ Using equation 22
      update velocity *particle*(*i*) ▷ Using equation 23
      if *fitness*(*particle*(*i*)) < *fitness*(*p_best*) **then**
        p_best = *particle*(*i*)
      end if
      if *fitness*(*particle*(*i*)) < *fitness*(*g_best*) **then**
        g_best = *p_best*
      end if
    end for
  end while
  Return Locating Next-Hop($g\_best_{particle(i)}$)
end procedure

Algorithm 2 starts with initializing the population size as NoP and defines function Next_Hop() as the next hop for data transfer. Algorithm 2 is interpreting the following steps. Step 1, the number of particles is initialized. In step 2, the fitness value for the particle(i) is computed and its value is stored in *p_best*. In step 3, the global best value is computed. In step 4 inside the loop, the velocity and position are updated in equations 22–23 [26], [27] according to the fitness values obtained in the previous steps. Furthermore, local fitness (fitness(particle(i)) and global fitness (fitness(*p_best*)) are compared and the best value is selected as the global best. Finally, Next_Hop() is determined by the fitness values obtained in the previous steps. These steps are followed until the termination criteria are reached.

$$X_i(\tau + 1) = X_i(\tau) + V_i(\tau + 1) \tag{22}$$

$$V_i(\tau + 1) = \omega V_i(\tau) + c_1 r_1(p\_best - X_i(\tau)) + c_2 r_2(g\_best - X_i(\tau)) \tag{23}$$

C. STEADY STATE

After completing the set-up phase and broadcasting the configuration information to all nodes, we run the steady-state, which is executed at the sink. In IoT networks with multi-hop communication, nodes may fail due to the application state [30], [38]. In addition, nodes closer to the sink not only send their data messages to the sink, but also relay data messages from other nodes [27]. The occurrence of these problems leads to network segmentation. Hence, the routing algorithm must balance the energy consumption of nodes to increase network lifespan. To achieve a routing algorithm with energy-efficiency and energy-balanced, we select a number of effective paths as alternative routes. These paths change accordingly in each implementation of the algorithm. In other words, the algorithm determines another subset of links as effective paths in each execution. The sink node determines the effective paths by considering Conditional Directed Acyclic Graph (C-DAG) infrastructure with a formal definition of the problem [29]. To reduce the effects of cascading failure, the proposed method models the routing tree as follows:

1) Converting the routing tree to the DAG, so that each node of the tree is the vertex of DAG and each link between the nodes of the tree is a directional edge in DAG. If we reach the branch nodes, we label it Conditional node, and thus C-DAG is made.
2) Each vertex in C-DAG has the worst amount of energy required to handle packets. In fact, the residual energy of a node in the routing tree is considered the Worst Amount of Energy, known as WAE parameter.
3) If the formed C-DAG can be routed to the destination by WAEs, then routing data reliably reaches the sink.

The advantage of this schema is that when the network has a cascading failure, it can be switched to a more reliable link as the next-hop route with the priorities for sending the packet [39], [40]. Inspired by the problem model, a formal problem-solving approach emerges as a fault tolerance mechanism. The formed C-DAG with the following Algorithm 3 will configure the robustness of network topology with respect to cascading failure:

Algorithm 3 is interpreting the following steps based on C-DAG. In the first step, it receives *NULL* for the Threshold_isolated. This algorithm tries to save more nodes as isolated with the Threshold_isolated parameter. In fact, we consider two modes of overloaded and isolated for vertices. Noted that this algorithm is repeated to Maximum_Round. In the second step, it enters a loop that checks the next steps for each CH. In the routing phase, for each CH, RNs and routing trees are identified. Here, it checks

Algorithm 3 Pseudo Code of the Steady State Phase

Input: Formed C-DAG, Maximum_Round
Output: Find alternative links as new RNs

1: **procedure** GRAPH-BASED FAULT TOLERANT( )
2: Round←1
3: Threshold_isolated←NULL
4: **if** Maximum_Round<>Round **then**
5: ———— **CHs side** ————
6: Randomly start from a CH
7: **for** each *CH* **do**
8: RNs ← Locating Next-Hop(*CH*)
9: **if** RNs is not existed **then**
10: Isolate current *CH*
11: Broadcast(information)
12: go to line 5
13: **end if**
14: LP← Find_Longest_Path(C-DAG,*CH*) ▷ This function is described in the next part of subsection IV-C1
15: **if** Restrictions for LP are satisfied **then** ▷ Using Equations 25–26
16: Threshold_isolated← ⌊Average E_{LP}⌋
17: Broadcast(Threshold_isolated)
18: **end if**
19: ———— **Vertices side** ————
20: **if** vertices received Threshold_isolated **then**
21: **if** Threshold_isolated≥ $E_{residual}$ **then**
22: Isolate those vertices
23: Broadcast(information)
24: Update RNs
25: **end if**
26: Do forwarding based on RNs
27: **if** Data is reached to sink **then**
28: Save links for current *CH*
29: **end if**
30: **end if**
31: **end for**
32: Round←Round + 1
33: **end if**
34: **Return** New set of RNs for each *CH*
35: **end procedure**

if there is an RN for CH. In the third step, we find the longest path based on the topological order for that CH. This longest path as LP includes vertices and edges. We apply the constraints targeted in this paper to that path. If we are satisfied, we broadcast the average LP energy as Threshold_isolated. Vertices change or keep their state by receiving this parameter. If their residual energy is higher than this parameter, they will continue to transfer data. Otherwise, they isolate themselves. However, if the data reaches the sink in the next step, it returns the effective path as output for that CH. After performing this algorithm, each CH, in addition to RNs (from the set-up phase), may also have new RNs recommended for efficient routing over cascading failure.

1) FORMAL DEFINITION

In a DAG $G_i = (v_i, e_i)$, every v_i is a sequential node from routing tree. Arcs represent routing constraints between nodes. C-DAG consists of two types of nodes including normal node and conditional node (route branching) [29]. However, each v_i saves a WAE parameter and updates it at each execution of algorithm 3. The whole process of forming a modeled graph is shown in Figures 5 (a) and (b).

The pair (v_1, v_2) in a DAG $G_i = (v_i, e_i)$ is conditional nodes if the following definitions are satisfied:

As shown in Figure 5 (c), assume there are exactly ζ egress arcs from node v_1 with this specification $a_1, a_2, \ldots, a_\zeta$ and $\zeta > 1$. In that case, there are accurately ζ ingress arcs into v_2 in e_i with this specification $b_1, b_2, \ldots, b_\zeta$. As shown in Figure 5 (d), for each ξ member $\{1, 2, \ldots, \zeta\}$, if assume v_ξ subset v_i and e_ξ subset e_i, then all nodes (exclude v_2) are in accessible paths from node a_1. By definition, a_1 is the only source node of the DAG $G_\xi = (v_\xi, e_\xi)$. Note that b_1 the only sink node of G_ξ [29].

As given in the Algorithm 3, due to the efficiency, we solve the problem in the worst-case scenario, so we demand to find the longest path. The longest path LP_i of a C-DAG node n_i is any source-sink path of route that achieves the longest length. E_{LP} also represents the energy required to accomplish this path when the number of nodes is large enough. Hence, After determining the longest path, E_{LP} is the sum of the WAEs of all its nodes:

$$E_{LP} = \sum_{n=1}^{N} WAE(n) \tag{24}$$

Here, n is the number of nodes in the path. Computing Equation 24 is not trivial under normal status. Finding the longest path, unlike finding the shortest path, is an NP-hard problem. In this paper, we can be reduced to linear time with the following steps:

1) Compute the topological sorting algorithm for C-DAG. So that if there is an arc from u to v, u will appear in the order before v.
2) For each node v_i of the C-DAG, in topological order, calculate the length of LP_i ending in v_i by looking at its ingress neighbors, and add WAE to the maximum length listed for those neighbors. If v_i has no ingress neighbor, fixed the length of LP_i ending at v_i to WAE.
3) Eventually, LP_i may be gaining starting from v_i with the maximum WAE marked, then frequently stepping backward to its egress neighbor with the largest WAE and reversing the order obtained in this process.

All these steps can be done in linear time complexity $O(n)$. Now, we have defined the mapping graph and know how to measure the longest path, we can describe the level of the fault

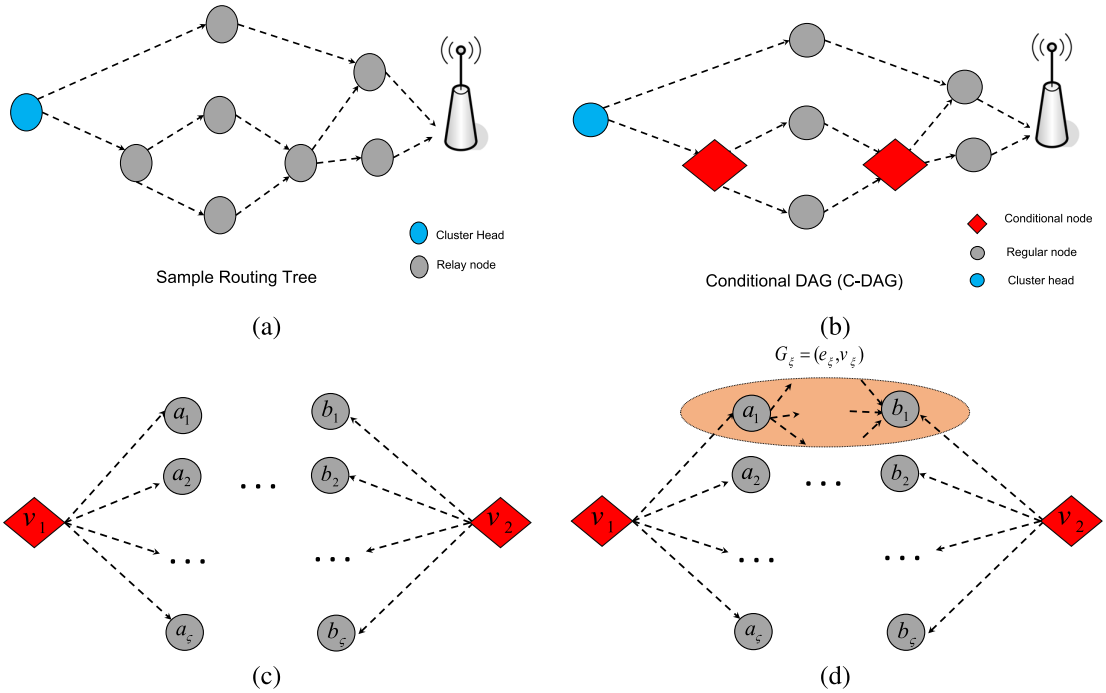

FIGURE 5. a) Tree sample generated in the set-up stage; b) C-DAG sample modeled in the steady-state phase; c) Formal definition of conditional pairs; d) A subset definition of conditional pairs.

tolerance mechanism with the following formulas.

$$feasible\_restriction = \begin{cases} if\ E_{LP} \geq E_{Total} \rightarrow feasible \\ o.w \rightarrow not\ feasible \end{cases} \quad (25)$$

According to the set-up state, paths were determined from the CHs to the sink. If the condition of Equation 25 is met, it means that in the worst-case scenario, there will be an alternative path. Otherwise, Due to the application, the next-hop of transmitting will be changed. If the next hop is busy, the current vertex will be isolated. In other words, if we miss the *feasible_restriction*, then at this round, the algorithm will stop running and enter to next round based on execution priority for CHs. Suppose that the condition of Equation 25 is satisfied, where it is necessary to check whether the data reliably reaches the sink despite the interference in the paths or not. The interference in the paths means that in order for packets to reach the sink from different branches, some branches may interfere [29]. If the maximum number of branches is m parameter, then the *reliable_restriction* in Equation 26 can be checked as follow:

$$reliable\_restriction = \begin{cases} if\ R_i^{ub} \geq D_i \rightarrow reliable \\ o.w \rightarrow not\ reliable \end{cases} \quad (26)$$

Here, D_i is the sum of minimum energy required for a node to handle packets and R_i^{ub} is the upper-bound response energy of path that reach to the sink as scheduling status is computed as follow:

$$R_i = E_{LP} + Interference(i) = E_{LP} + \frac{1}{m}(I_{i,i}) + \frac{\sum_{j \neq i} I_{i,j}}{m} \quad (27)$$

Here, $\frac{1}{m}(I_{i,i})$ is the intra-interference of path and $\frac{\sum_{j \neq i} I_{i,j}}{m}$ is the inter-interference of path. We must somehow eliminate these interference or minimize their effect [29]. Based on this, we have to compute the upper-bound of R_i as follow:

$$R_i \leq R_i^{ub} = E_{LP} + \frac{1}{m}(W_i - E_{LP}) \quad (28)$$

Here, W_i is defined as the workload of the graph so that nodes that are never selected in the longest path are eliminated. Thus, the branches leading to them are not counted. In that case, by putting $I_{i,k}(L) \leq W_k(L)$ in Equation 27 and if the type of scheduler is global fixed-priority (FP) with deadline energy D, then the value can be obtained.

Assume Figure 6 is a sample of a routing tree produced for microgrid application, where for CH1, new RN sets are generated in Table 1 to deal with possible cascading failure in Maximum Rounds. Suppose that in the set-up phase, set RN [2, 3, 5, 7, *sink*] was selected for CH1. Further, feasible and reliable constraints for CH1 were satisfied in the steady-state. As shown in Table 1, based on the overloaded nodes, REFIT generates possible effective paths as an alternative set of RNs. Route branching, which occurs at conditional nodes 2 and 5, indicates that sender node isolation is the near-optimal process when there is no other path to reach the sink. However, if the cascading failure does not affect our predetermined path (before the RN set), we consider the same path as the new RNs as the set-up phase generated the best solution. The table 1 shows effective and intelligent monitoring for all clusters so that in the event of failure, the best approach with minimum cost and energy replaces the current methods. All generated solutions for all CHs are

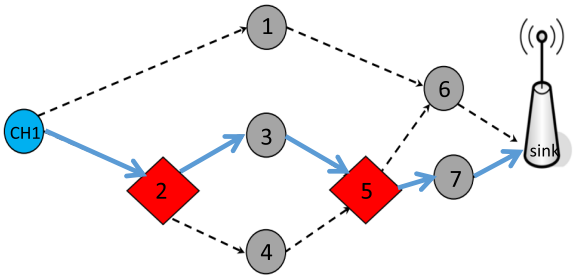

FIGURE 6. Sample routing tree to show the steady state processes.

TABLE 1. The output table generated for Figure 6.

CHs	Before RNs	Overloaded node	New RNs
CH1	[2, 3, 5, 7, sink]	Level 1: Node 2 or Node 5 or both of them	[1,6, sink] or isolated CH1
CH1	[2, 3, 5, 7, sink]	Level 2: Node 3 or node 7 or both of them	[2, 4, 5, 7, sink] or [2, 3, 5, 6, sink] or [1,6, sink] or isolated CH1, respectively.
CH1	[2, 3, 5, 7, sink]	None of them	[2, 3, 5, 7, sink]
CH2	...	...	...

stored in the sink, and the sink broadcasts the best solutions (alternative paths) based on the maximum residual energy of all solutions for each CH.

V. EXPERIMENTAL SIMULATION

All simulations are performed in MATLAB. To evaluate the proposed method, we use the randomly generated Figure 7 (a) topology in a two-dimensional area with a size 100m × 100m. We have presented this topology with a 2-D triangular plot to make the dimensions of the environment covered by its nodes more tangible. The sink node is located at the center of this topology.

The network is homogeneous and the nodes have a communication range of 20m and the same capacity. We repeated the experiment for 20 times and in each experiment, we executed the algorithm for 1000 iterations and reported the average result. Table 2 shows the list of parameters used in the simulation. Figure 7 (b) shows the load distribution based on topology of Figure 7 (a). It is clearly observed that in some areas, the traffic load is not uniformly distributed. The unbalanced coverage seen in Figure 7 (b) pushes the traffic flow, so there is a possibility of cascading failure at red circular points where the degree of nodes increases.

In the following, we will first analyze the proposed CH selection method and then evaluate the performance of the proposed fault tolerance mechanism in the face of cascading failure.

A. CH SELECTION ANALYSIS

In this section, we analyze the clustering approach used in this paper. Figure 7 (c) shows the number of CHs selected for our method based on different communication ranges (radius) of

TABLE 2. Simulation parameters.

Parameters	Values
Area	100m × 100m
scheduler	Global Fixed-Priority
Number of nodes (N)	100
Communication Range	20m
Initial energy	10 j
Maximum Iterations	1000
Maximum Rounds	75
Particle position	[0,100]
Particle velocity	[0,100]
Data packet size	512 B
Location of Base Station	(50,50)
Eamp	130 (pJ/bit/m2)
Eelec	50 (nJ/bit)
Efs	10 (pJ/bit/m2)

nodes. If the communication radius of the nodes decreases, then the number of CHs increases, which means the percentage of the number of nodes grows in each cluster. REFIT, unlike other methods [7], [26], [27], does not consider the number of CHs constant, which increases scalability for network size. For REFIT, if the total transmission range is between 96 and 100 meters, there will be about 4 to 5 CHs for IoT applications.

To show the advantage of the proposed method, we use the classical Leach method [36], Genetic algorithm (GA) [41], and PSO [26] to compare the clustering section. GA and PSO are known as representative of evolutionary algorithms and optimize energy-efficiency and network topology. Figure 7 (d) shows the distribution pattern of the clusters for different approaches. As shown in Figure 7 (d), the overall distribution of nodes in each cluster for different methods states that REFIT experiences a uniform scattering pattern because it has a lower average distribution of CMs than the percentage increase in the number of CHs. However, this uniform pattern prolongs the network lifetime.

Figure 8 indicates how the REFIT algorithm affects the load distribution of each node in the network. Before applying REFIT (Figure 8 (a)), the network load is unbalanced, in which the possibility of cascading failure is high, while after applying REFIT (Figure 8 (b)), the network load for each node is reduced and the load distribution is more balanced. The set-up state helps to balance the network load distribution and to optimize network topology by selecting the best set CHs by the functions 12–14 and then selecting the effective RNs using the modified PSO with the function 21.

B. NETWORK PERFORMANCE EVALUATION

In this section, after evaluating REFIT, we compare it with MA-TOSCA [18], GA [41], and PSO [26] methods. MA-TOSCA mentioned in the related works benefits the memetic algorithm (from the "Selfish Gene" concept) [18]. This novel method utilizes a population-based search approach to discover proper areas in the search space

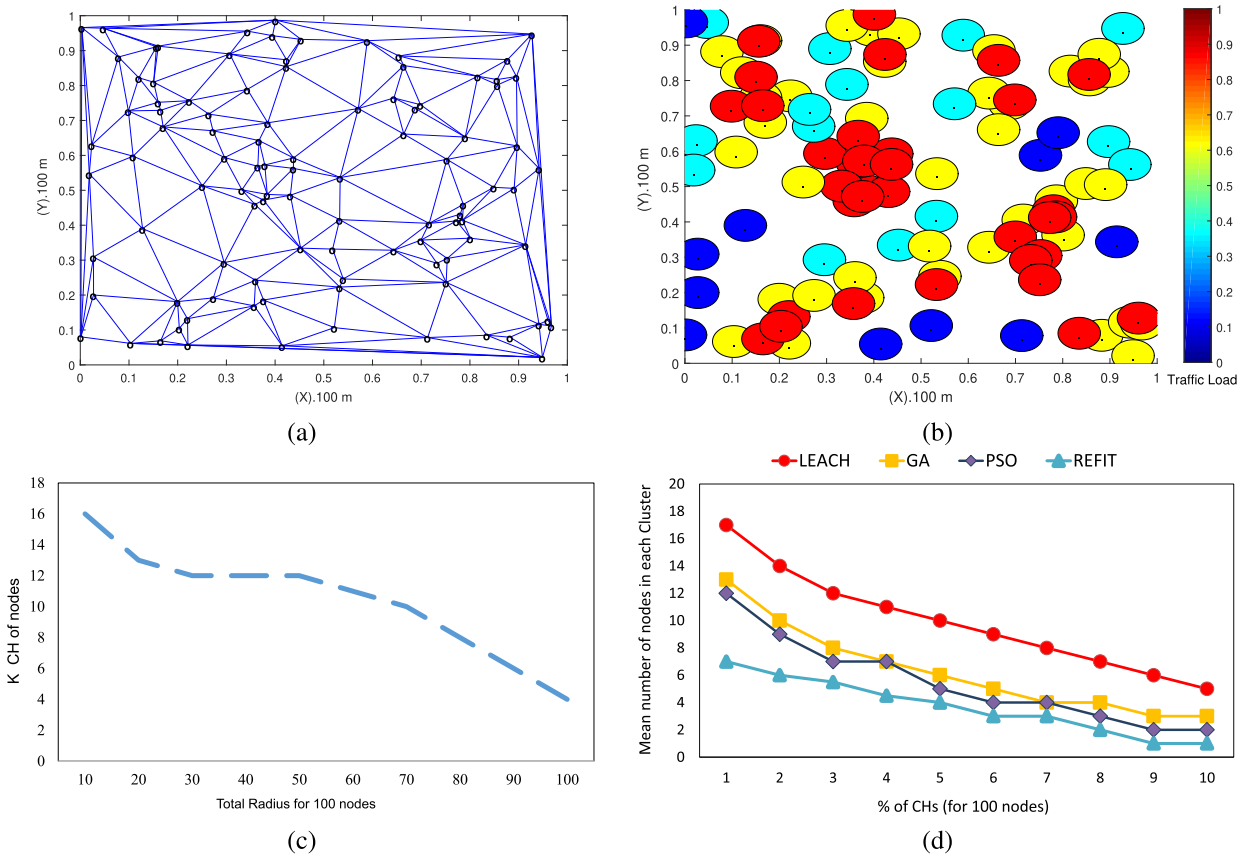

FIGURE 7. a) Network topology of our simulation experiments; b) Network load distributions based network topology; c) Number of selected CHs for 100 nodes based on the radius; d) Distribution pattern of the clusters for different approaches.

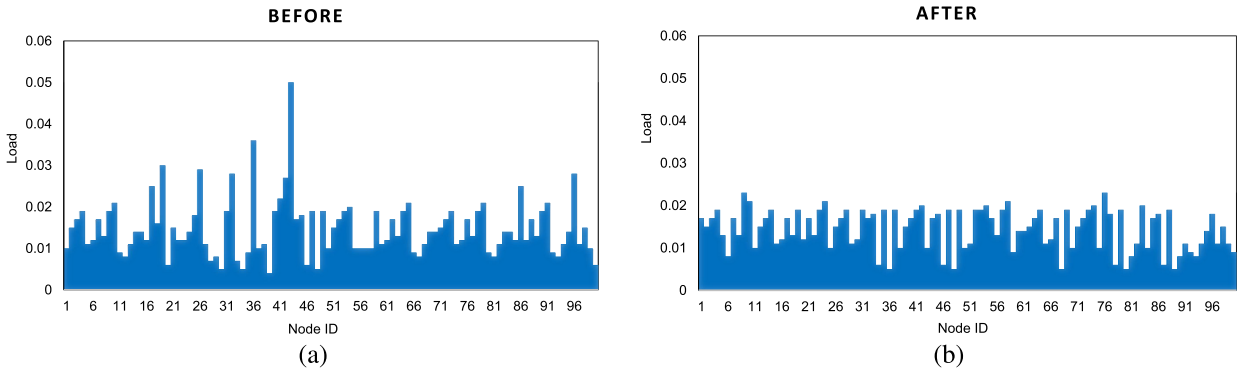

FIGURE 8. Load distribution of 100 nodes before (a) and after (b) applying REFIT.

and specify the optimal local area through exploratory. MA-TOSCA combines global and local optimal points, which results in excellent performance [18]. REFIT, based on the vacant energy of the nodes and the topology formed by the clustering, will be more efficient than these methods.

1) ROBUSTNESS METRICS

We utilized the robustness metrics employed in [17]. In this criterion, first, node i is randomly removed from the topology, and then the number of sensor nodes that can have at least one path to the sink is calculated. In fact, the number of endurance nodes is recorded. Suppose the size of endurance nodes Φ_i and N is the total number of sensor nodes in the network, we can easily get that $0 \leq \Phi_i \leq N-1$.

$$\mathbb{R} = \frac{\sum_{i \in \Psi} \Phi_i}{N(N-1)} \tag{29}$$

Normalized this metric is given between 0 and 1 in Equation 29.

Figure 9 (a) shows REFIT robustness with varying load-coefficient (σ) and overload-coefficient (δ). Obviously, as the capacity of the nodes increases (δ), network robustness improves. In contrast, as σ coefficient increases, Equation 29 ($\mathbb{R}$) decreases exponentially. The concurrent effect of these

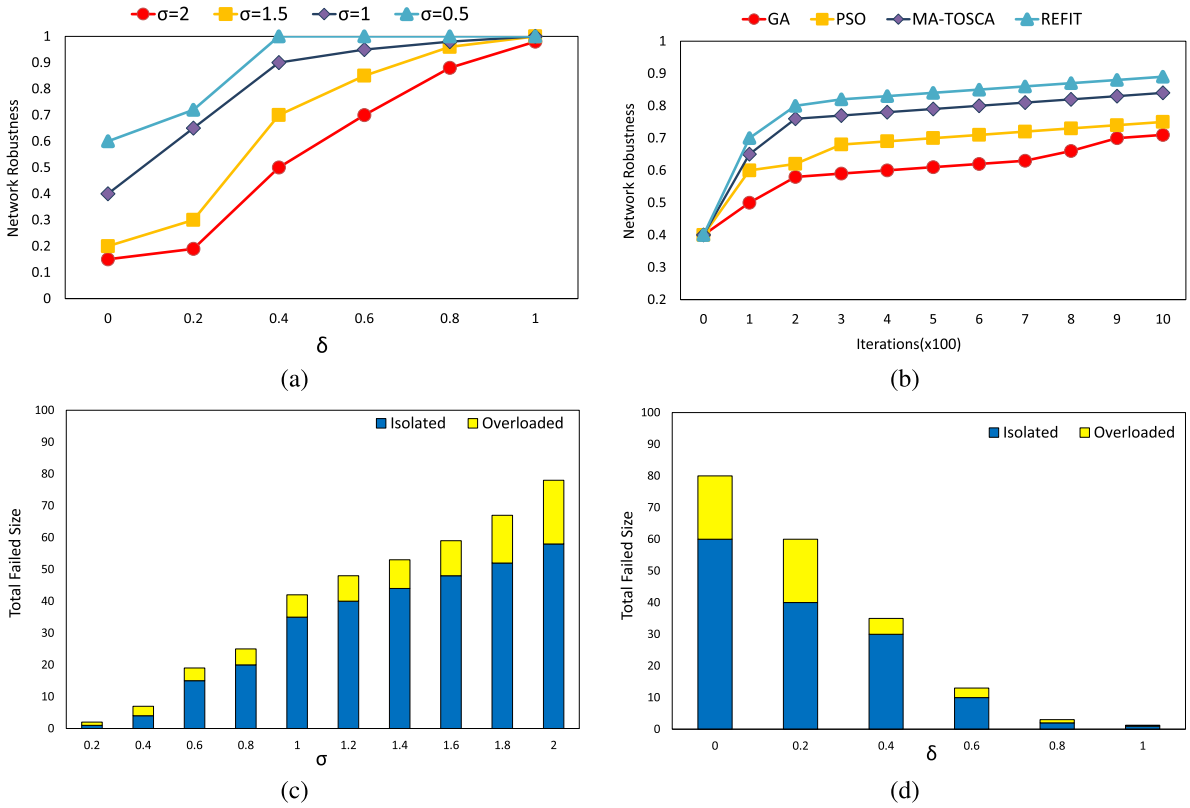

FIGURE 9. Robustness Metrics: a) Network robustness of REFIT with varying δ and σ coefficients; b) Network robustness of REFIT versus three reference algorithms; c) Comparison of failed nodes against diversity σ; d) Comparison of failed nodes against diversity δ.

two coefficients on $\mathbb{R}$ is shown in Figure 9 (a). The capacity threshold for sensor nodes that protect network topology against cascading failure is overload-coefficient = [0.4, 0.6] value. As shown in Figure 9 (a), if the two coefficients δ and σ are [0.4, 0.6] and 0.5 respectively, $\mathbb{R}$ will reach its maximum value, which means that any failure in the nodes will not cause a cascading failure. However, when δ becomes larger than capacity threshold ([0.4, 0.6]), σ changes ([0.5, 2]) do not have a tangible effect on the $\mathbb{R}$, thus the robust network requires more resource capacity and cost.

Figure 9 (b) depicts network robustness versus three reference algorithms. From Figure 9 (b), we can clearly observe that REFIT can obtain a more robust topology structure than other methods. The routing tree constructed with PSO and then modeled with C-DAG makes the REFIT method about 8% better than MA-TOSCA per 1000 iterations of network robustness metric. However, famous evolutionary algorithms have lowered robust of the network topology. Figures 9 (c) and (d) compare the total number of failed nodes against variations in σ and δ, respectively. Here, the sum of the number of isolated and overloaded nodes is assumed to be failed nodes. In Figure 9 (c), suppose that when σ changes, we consider overload-coefficient to be a constant value ($\delta = 1$). Clearly, as σ increases, the network load becomes unbalanced and the ratio of the number of failed nodes increases. According to Figure 9 (c), REFIT grows the ratio of isolated to overloaded nodes by rising σ.

Conversely for Figures 9 (d), suppose that when δ changes, we consider load-coefficient to be a constant value ($\sigma = 1$). Figure 9 (d) depicts that for values of δ below the capacity threshold, the ratio of the number of isolated nodes to overloaded is higher. Furthermore, when we cross the capacity threshold and reach $\delta = 1$, the ratio of overloaded nodes becomes little, and even at $\delta = (0.8,1]$, no overloaded nodes are reported on the REFIT network.

2) PERFORMANCE METRICS

In this section, REFIT is compared with three reference methods based on performance metrics: mean residual energy, clustering coefficient, and mean length of the shortest path. On the other hand, these three metrics can be considered as topological factors. As shown in Figure 10 (a), REFIT has a higher mean residual energy than the other methods. Hence, the network lifetime improves, and consequently, the duration of network resistance against failures also increases. According to Figure 10 (a), REFIT about 22% better than MA-TOSCA per 1000 iterations of mean residual energy metric. We can clearly observe that PSO algorithm alone consumes a lot of energy in the face of cascading failure, while its modified version helps to the energy efficiency of REFIT.

Figures 10 (b) and (c) depict clustering coefficient and mean length of the shortest path for different methods against cascading failure, respectively. Increasing gradually of

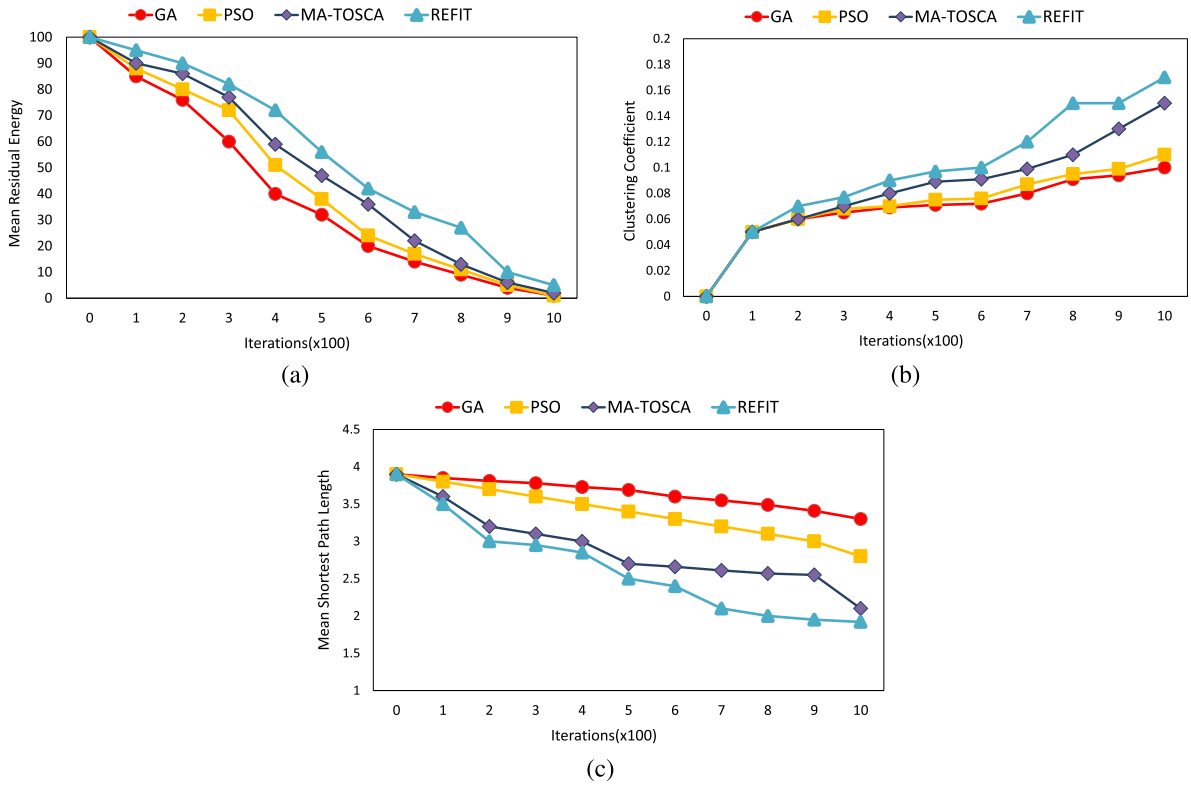

FIGURE 10. Performance Metrics: a) Mean residual energy of REFIT versus three reference algorithms; b) Clustering coefficient of REFIT versus three reference algorithms; c) Mean length of the shortest path of REFIT versus three reference algorithms.

clustering coefficient and decreasing the mean length of the shortest path proves the efficiency of REFIT. Statistically, the REFIT method has increased the clustering coefficient by about 36%, 51%, and 59% as compared to MA-TOSCA, PSO, and GA algorithms, respectively. Also, Our method decreased the mean length of the shortest path by about 23%, 34%, and 42% as compared to MA-TOSCA, PSO, and GA algorithms, respectively. This means that improving the topological parameters will enhance the network robustness because both the local connections increase and data reaches the sink with fewer hops. From Figure 10 (c), it can be inferred that despite the cascading failure, short paths can be utilized as crucial shortcuts, which our method effectively benefits this factor.

VI. CONCLUSION

Using the PSO algorithm for clustering in IoT networks leads to an increase in energy efficiency. But appearing cascading failure will lead to a severe decline in network lifetime. In this research, by a modified version of PSO, we were able to reduce the effect of cascading failure in the network topology. On this basis, the proposed method divides the network operations into two phases, set-up state and steady-state. In the set-up, the supreme set of CHs and their RNs are generated, and then in the steady-state, the fault tolerance mechanism promoted by modeling the routing tree to C-DAG which generates shortcut nodes as effective paths. Through extensive simulations, we prove that the proposed REFIT can obtain a more robust topology structure than existing algorithms. Moreover, REFIT provides a suitable infrastructure for small and large-scale IoT applications such as microgrids by not assuming a constant number of CHs and balanced distribution of network load. In the future, we will focus more on evaluating REFIT in multi-sink and mobile sink scenarios against cascading failure.

REFERENCES

[1] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-Things-based smart cities: Recent advances and challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, 2017.

[2] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of Things applications: A systematic review," *Comput. Netw.*, vol. 148, pp. 241–261, Jan. 2019.

[3] H. Fotouhi, M. Alves, M. Z. Zamalloa, and A. Koubaa, "Reliable and fast hand-offs in low-power wireless networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 11, pp. 2620–2633, Nov. 2014.

[4] M. A. Araghizadeh, P. Teymoori, N. Yazdani, and S. Safari, "An efficient medium access control protocol for WSN-UAV," *Ad Hoc Netw.*, vol. 52, pp. 146–159, Dec. 2016.

[5] M. Adnan and M. Tariq, "Cascading overload failure analysis in renewable integrated power grids," *Rel. Eng. Syst. Saf.*, vol. 198, Jun. 2020, Art. no. 106887.

[6] S. Chang, "An emergence alert broadcast based on cluster diversity for autonomous vehicles in indoor environments," *IEEE Access*, vol. 8, pp. 84385–84395, 2020.

[7] M. Biabani, H. Fotouhi, and N. Yazdani, "An energy-efficient evolutionary clustering technique for disaster management in IoT networks," *Sensors*, vol. 20, no. 9, p. 2647, May 2020.

[8] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[9] T. Han, L. Zhang, S. Pirbhulal, W. Wu, and V. H. C. de Albuquerque, "A novel cluster head selection technique for edge-computing based IoMT systems," *Comput. Netw.*, vol. 158, pp. 114–122, Jul. 2019.
[10] N. Yao, X. Hao, D. Liu, W. Liu, and B. Chen, "Research on channel allocation game algorithm for improving robustness in WSN," *Phys. Commun.*, vol. 43, Dec. 2020, Art. no. 101230.
[11] Y.-C. Wang and K.-C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1527–1540, Jul. 2019.
[12] W. Li, Y. Han, P. Wang, and H. Guan, "Invulnerability analysis of traffic network in tourist attraction under unexpected emergency events based on cascading failure," *IEEE Access*, vol. 7, pp. 147383–147398, 2019.
[13] J. Wu, Z. Chen, Y. Zhang, Y. Xia, and X. Chen, "Sequential recovery of complex networks suffering from cascading failure blackouts," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2997–3007, Oct. 2020.
[14] Z. J. Bao, Y. J. Cao, L. J. Ding, Z. X. Han, and G. Z. Wang, "Dynamics of load entropy during cascading failure propagation in scale-free networks," *Phys. Lett. A*, vol. 372, no. 36, pp. 5778–5782, Sep. 2008.
[15] X. Fu, H. Yao, and Y. Yang, "Exploring the invulnerability of wireless sensor networks against cascading failures," *Inf. Sci.*, vol. 491, pp. 289–305, Jul. 2019.
[16] J. Zhong, H. Sanhedrai, F. Zhang, Y. Yang, S. Guo, S. Yang, and D. Li, "Network endurance against cascading overload failure," *Rel. Eng. Syst. Saf.*, vol. 201, Sep. 2020, Art. no. 106916.
[17] X. Fu and Y. Yang, "Modeling and analysis of cascading node-link failures in multi-sink wireless sensor networks," *Rel. Eng. Syst. Saf.*, vol. 197, May 2020, Art. no. 106815.
[18] X. Fu, P. Pace, G. Aloi, L. Yang, and G. Fortino, "Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107327.
[19] C.-Y. Chen, Y. Zhao, J. Gao, and H. E. Stanley, "Nonlinear model of cascade failure in weighted complex networks considering overloaded edges," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020.
[20] G. Zhang, Z. Li, B. Zhang, and W. A. Halang, "Understanding the cascading failures in Indian power grids with complex networks theory," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 15, pp. 3273–3280, Aug. 2013.
[21] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
[22] P. Osterrieder, L. Budde, and T. Friedli, "The smart factory as a key construct of industry 4.0: A systematic literature review," *Int. J. Prod. Econ.*, vol. 221, Mar. 2020, Art. no. 107476.
[23] A. Kondoro, I. Ben Dhaou, H. Tenhunen, and N. Mvungi, "Real time performance analysis of secure IoT protocols for microgrid communication," *Future Gener. Comput. Syst.*, vol. 116, pp. 1–12, Mar. 2021.
[24] W. Liao, S. Salinas, M. Li, P. Li, and K. A. Loparo, "Cascading failure attacks in the power system: A stochastic game perspective," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2247–2259, Dec. 2017.
[25] R.-R. Yin, B. Liu, H.-R. Liu, and Y.-Q. Li, "The critical load of scale-free fault-tolerant topology in wireless sensor networks for cascading failures," *Phys. A, Stat. Mech. Appl.*, vol. 409, pp. 8–16, Sep. 2014.
[26] R. S. Y. Elhabyan and M. C. E. Yagoub, "Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network," *J. Netw. Comput. Appl.*, vol. 52, pp. 116–128, Jun. 2015.
[27] B. M. Sahoo, T. Amgoth, and H. M. Pandey, "Particle swarm optimization based energy efficient clustering and sink mobility in heterogeneous wireless sensor network," *Ad Hoc Netw.*, vol. 106, Sep. 2020, Art. no. 102237.
[28] P. Maheshwari, A. K. Sharma, and K. Verma, "Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization," *Ad Hoc Netw.*, vol. 110, Jan. 2021, Art. no. 102317.
[29] A. Melani, M. Bertogna, V. Bonifaci, A. Marchetti-Spaccamela, and G. C. Buttazzo, "Response-time analysis of conditional DAG tasks in multiprocessor systems," in *Proc. 27th Euromicro Conf. Real-Time Syst.*, Jul. 2015, pp. 211–221.
[30] R. A. Shuvro, P. Das, M. M. Hayat, and M. Talukder, "Predicting cascading failures in power grids using machine learning algorithms," in *Proc. North Amer. Power Symp. (NAPS)*, Oct. 2019, pp. 1–6.
[31] M. Adnan, M. Ali, A. Basalamah, and M. Tariq, "Preventing cascading failure through fuzzy co-operative control mechanism using V2G," *IEEE Access*, vol. 7, pp. 142607–142622, 2019.
[32] H. Dui, X. Meng, H. Xiao, and J. Guo, "Analysis of the cascading failure for scale-free networks based on a multi-strategy evolutionary game," *Rel. Eng. Syst. Saf.*, vol. 199, Jul. 2020, Art. no. 106919.
[33] C. Zhai, H. Zhang, G. Xiao, and T.-C. Pan, "An optimal control approach to identify the worst-case cascading failures in power systems," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 2, pp. 956–966, Jun. 2020.
[34] J. Sakhnini, H. Karimipour, A. Dehghantanha, R. M. Parizi, and G. Srivastava, "Security aspects of Internet of Things aided smart grids: A bibliometric survey," *Internet Things*, Sep. 2019, Art. no. 100111.
[35] N. M. Tabatabaei, E. Kabalci, and N. Bizon, *Microgrid Architectures, Control and Protection Methods*. Cham, Switzerland: Springer, 2019.
[36] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
[37] G. Armano and M. R. Farmani, "Multiobjective clustering analysis using particle swarm optimization," *Expert Syst. Appl.*, vol. 55, pp. 184–193, Aug. 2016.
[38] K. M. Lhaksmana, Y. Murakami, and T. Ishida, "Cascading failure tolerance in large-scale service networks," in *Proc. IEEE Int. Conf. Services Comput.*, Jun. 2015, pp. 1–8.
[39] J.-W. Lin, P. R. Chelliah, M.-C. Hsu, and J.-X. Hou, "Efficient fault-tolerant routing in IoT wireless sensor networks based on bipartite-flow graph modeling," *IEEE Access*, vol. 7, pp. 14022–14034, 2019.
[40] Z. Wang, H. Chen, Q. Cao, H. Qi, and Z. Wang, "Fault tolerant barrier coverage for wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2014, pp. 1869–1877.
[41] T. Wang, G. Zhang, X. Yang, and A. Vajdi, "Genetic algorithm for energy-efficient clustering and routing in wireless sensor networks," *J. Syst. Softw.*, vol. 146, pp. 196–214, Dec. 2018.

MORTEZA BIABANI received the B.S. and M.S. degrees in computer engineering from the University of Tabriz, Tabriz, Iran, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, University of Tehran. He is also working as a Researcher with the Router Laboratory, University of Tehran. He received several distinguished student awards from the University of Tabriz. His current research interests include the Internet of Things and wireless sensor networks.

NASSER YAZDANI received the B.S. degree in computer engineering from the Sharif University of Technology, Tehran, Iran, and the Ph.D. degree in computer science and engineering from Case Western Reserve University, Cleveland, OH, USA. For few years, he worked with the Iran Telecommunication Research Center (ITRC), as a Consultant, a Researcher, and a Developer. Then, he worked with different companies and research institutes in USA. In September 2000, he joined the ECE Department, University of Tehran, Tehran. He initiated different research projects and labs in high speed networking and systems. He is currently a Full Professor with the School of Electrical and Computer Engineering, University of Tehran. His research interests include networking, packet switching, access methods, operating systems, and database systems.

HOSSEIN FOTOUHI received the B.S. degree in electrical engineering in 2004, the M.S. degree in communication network engineering from University Putra Malaysia, in 2009, and the Ph.D. degree from the University of Porto, in April 2015. He is currently an Assistant Professor with the School of Innovation, Design, and Engineering, Mälardalen University. His research interests include the Internet of Things, sensor networks, wireless communication, and Fog computing.