A Comprehensive Review of Fault-Tolerant Routing Mechanisms for the Internet of Things

Zhengxin Lan*
College of Power Technology, Liuzhou Railway Vocational and Technical College, Liuzhou 545616, Guangxi, China

***Abstract*—The Internet of Things (IoT) facilitates intelligent communication and real-time data collection through dynamic networks. The IoT technology is ideally suited to meet intelligent city requirements and enable remote access. Several cloud-based approaches have been proposed for constrained IoT systems, including scalable data storage and effective routing. In real-world scenarios, the effectiveness of many methods for wireless networks and communication links can be challenged due to their unpredictable characteristics. These challenges can result in path failures and increased resource utilization. To enhance the reliability and resilience of IoT networks in the face of failures, fault tolerance mechanisms are crucial. Network failures can occur for various reasons, including the breakdown of the wireless nodes' communication module, node failures caused by battery drain, and changes in the network topology. Addressing these issues is essential to ensure the continuous and reliable operation of IoT networks. Fault-tolerant routing plays a critical role in IoT-based networks, but no systematic and comprehensive research has been conducted in this area. Therefore, this paper aims to fill this gap by reviewing state-of-the-art mechanisms. An analysis of the practical techniques leads to recommendations for further research.**

Keywords—Internet of things; routing; data transmission; fault-tolerant; review

I. Introduction

The Internet of Things (IoT) has witnessed rapid growth due to advancements in software and hardware platforms, the expansion of communication networks, and the progress in data analysis tools [1]. IoT refers to a network of interconnected devices that generate and collect data using technologies like RFID, sensors, actuators, and mobile phones [2, 3]. The International Telecommunication Union (ITU) defines IoT as a global infrastructure that connects physical and virtual objects through integrated information and communication technologies. Another definition in [4] describes IoT as a network that connects people and things, allowing them to be connected anytime, anywhere, with anyone and anything, utilizing various networks, paths, and services. Moreover, IoT technology enables communication-related services such as information exchange for collaborative IoT services, user connectivity for global networking, and data offloading to edge cloud servers for enhanced computation [5].

There has been a growing interest in the IoT in academia and industry over the last few years. As the IoT becomes increasingly ubiquitous, it provides comprehensive representations of physical environments and offers a high level of engagement with them [6]. Some examples of potential applications for this innovative paradigm include e-health, business management, Intelligent Transportation Systems (ITS), and logistics. The realization of IoT is greatly influenced by several factors, such as the system's architecture, the network and communication infrastructure, the processing of data, and ubiquitous computing technologies, which enable effective, reliable, physical, and cyber connectivity. As part of IoT, networking, and primarily routing in the network, is an integral component that facilitates the interconnection of devices. It entails the creation of traffic routes and routing packets from the source to the final destination in a network.

In the realm of the Internet of Things (IoT), the convergence of big data, cloud computing, artificial intelligence (AI), machine learning (ML), deep learning, feature and channel selection, meta-heuristic algorithms, game theory, and association rule mining holds paramount significance, shaping the potential and efficacy of IoT applications. The massive influx of data generated by interconnected devices in IoT necessitates the utilization of big data techniques to handle, store, and process this vast volume of information [7]. Cloud computing plays a pivotal role by providing the scalable infrastructure and computational power required for efficient data management and analysis in IoT environments [8]. The integration of AI and ML in IoT applications unlocks the capability of devices to learn from data patterns, enabling them to make intelligent decisions, adapt to changing circumstances, and optimize their performance [9-11]. Deep learning, a subset of ML, further empowers IoT devices to process complex and unstructured data like images, audio, and text, enabling advanced applications in computer vision, speech recognition, and natural language processing [12-15]. Feature and channel selection play a critical role in optimizing IoT data processing by identifying the most relevant data attributes and sources, streamlining data analysis and reducing computational overhead [16]. Meta-heuristic algorithms find applicability in IoT systems for optimization tasks, such as routing and resource allocation, ensuring efficient utilization of resources and enhancing IoT network performance [17, 18]. Moreover, game theory principles are instrumental in designing cooperative or competitive strategies among IoT devices, optimizing resource allocation and energy usage in dynamic IoT environments [19]. Association rule mining brings valuable insights by discovering patterns and correlations within IoT data, aiding in decision-making processes and providing recommendations for improving IoT system performance [20].

Despite the promise of IoT, ensuring network reliability remains a significant challenge. Broken links and faulty nodes can severely impact the reliability of IoT-enabled networks, hindering seamless communication and data exchange. To address this challenge and enhance the resilience of IoT networks, fault-tolerant routing mechanisms are indispensable. These mechanisms are designed to maintain network performance and data delivery even in the presence of failures, disruptions, or changes in the network topology. In light of the growing importance of fault tolerance in IoT networks, this review paper aims to explore state-of-the-art fault-tolerant routing mechanisms comprehensively. By analyzing critical factors such as packet delivery ratio, network lifetime, energy consumption, delay, scalability, availability, and reliability, we seek to identify effective techniques and recommend further research in this crucial area. Our goal is to contribute to the advancement of reliable and resilient IoT networks, unlocking their full potential in various applications and domains.

II. Background

A. IoT Definitions

The IoT has been defined as a communication network where all our daily devices can connect to other devices by identifying, sensing, and processing functions. The International Telecommunication Union (ITU) has defined these devices as innovative items of the information world (virtual objects) or the physical world (physical objects), which may connect and exchange information with each other [21]. As a dynamic global network infrastructure based on standard and adaptive communication protocols, the IoT enables smart and virtual objects to be autonomous, have a physical identity, and have virtual personalities [22]. The IoT is regarded as the most influential among emerging technologies, ranking higher than artificial intelligence and robotics. The IoT is currently the most significant technology trend in the world, as stated by Burrus [23]; it is expected to cause the most disruption and provide the most opportunities over the next five years. According to a study published in Forbes Insights by over 500 executives from countries with a minimum of 500 employees, IoT has acquired global traction [24]. Forecasts encompass all aspects of the IoT ecosystem, including professional services, analytics, security, infrastructure purpose-built IoT platforms, connectivity services, and intelligent and embedded systems. All predictions indicate that IoT adoption and usage will grow regardless of the number of forecasts [5].

The IoT is defined in some ways only from a physical perspective. Real-world or virtual objects can be regarded as things. For example, Al-Fuqaha, et al. [25] defined IoT as the ability of tangible objects to perceive, listen, reason, communicate, collaborate on decisions, and fulfill tasks. As explained in a special IEEE report, the IoT is a network of connected devices, each equipped with sensors. IoT is also described by the Organization for the Advancement of Structured Information Standards (OASIS) as a system where ubiquitous sensors are used to connect the Internet to the physical world. This perspective is often called Machine to Machine (M2M) in definitions. As defined by ETSI, M2M communications are communications between two or more entities without direct human involvement. IoT covers many application fields today, including transport, utilities, healthcare, smart cities, and monitoring. It can apply to a variety of situations effectively. For instance, IoT can collect valuable data through sensor devices. In addition, IoT devices can serve as an efficient means of transmitting data. However, IoT devices have some limitations regarding transmission, processing, battery life, and memory capabilities.

B. IoT Architecture

All nodes in the IoT environment can communicate and cooperate to accomplish predefined goals. Therefore, the IoT must have a layering architecture that is flexible enough to support a wide range of heterogeneous elements over the Internet. Despite the growing number of IoT architectures, none can be adopted as a reference model. A typical IoT architecture comprises three layers: application, network, and perception. In the meantime, researchers have proposed updated models by incorporating additional abstractions into the architecture of the IoT. Fig. 1 illustrates two common IoT architectures. Several research studies like [26, 27] used a 5-layer model similar to TCP/IP. A 5-layer architecture model starts with the perception or objects layer, which comprises physical sensors that collect, process, and analyze information. Several crucial technologies facilitate the transfer of the produced data from the perception layer to the next layer, including Wireless Fidelity (Wi-Fi), Third Generation (3G), Radio Frequency Identification (RFID), Global System for Mobile Communications (GSM), and infrared. The object abstraction layer securely transfers data from the preceding layer (objects) to the upper layer (management of services). This layer serves as a standard interface to handle a variety of things. Known as the pairing (middleware) layer, the service management layer pairs services with requesters based on their names and addresses. IoT programmers can work independently with heterogeneous objects platforms through this layer. The application layer provides the requested services to users. The business layer creates a business representation, flowcharts, diagrams, etc., according to the data provided by the application layer.

Connectivity is a necessary and sufficient condition for IoT, which integrates different technologies. It is, therefore, essential to enhance communication protocols as part of the technology. IoT communication protocols are generally classified into three categories, Server to Server (S2S), Device to Device (D2D), and Device to Server (D2S). S2S communications involve the exchange of data between servers, which is mainly used in cellular networks. Mobile phones can communicate with each other through D2D communication, referred to as the next generation of cellular networks. In D2S, all data is transmitted to servers, regardless of location. These communications require the processing and preparation of data. This challenge calls for various data processing methods, such as analytics at the edge, stream analysis, and IoT analysis at the database. Each process should be customized based on the specific application and its requirements. Cloud and fog processing are two analytics tools used to prepare and process data before transferring it to another application. In a nutshell, sensors and IoT devices collect environmental data. The next step is to extract knowledge from the unprocessed data. Afterward, data can be transferred to other elements, devices, or servers over the Internet [8].

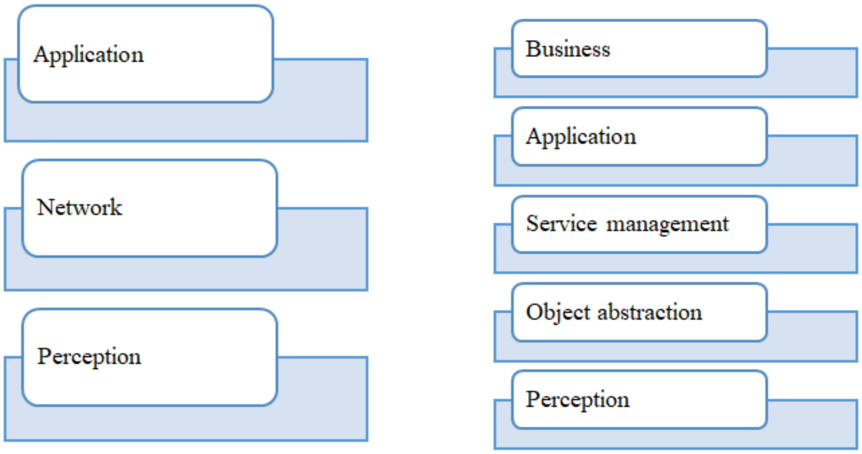

Fig. 1. Two common IoT architectures.

C. IoT Key Elements

As shown in Fig. 2, the IoT environment involves some critical elements defined below.

- Identification: The IoT requires identification to establish its services and match them with the demands of its users. Identifying objects paves the way for query, management, and control of object information. Currently, several identification methods exist, each with its characteristics in terms of coding schemes and analysis systems, such as Ubiquitous Codes (uCode) and Electronic Product Codes (EPC) [28].

- Sensing: IoT sensing involves capturing data from connected sensor nodes and delivering it to a database or data warehouse. Based on the recorded data, specific actions are taken according to the required services. IoT sensors can be intelligent actuators or wearable sensing devices [29].

- Communication: Connecting heterogeneous objects with IoT communication technologies enables customized intelligent services. Generally, IoT nodes should operate at low power when communication links are lossy and noisy. The IoT uses a variety of communication protocols, such as RFID, Near Field Communication (NFC), Wi-Fi, ultra-wide bandwidth (UWB), LTE-Advanced, IEEE 802.15.4, Z-wave, and Bluetooth [30].

- Computation: The IoT is powered by microcontrollers, microprocessors, SOCs, FPGAs, and software applications. IoT applications can be run on various hardware platforms, including T-Mote Sky, Z1, Cubieboard, Gadgeteer, Raspberry PI, and Arduino [31].

- Semantics: The concept of semantics focuses on the capability of machines to intelligently extract knowledge to facilitate the provision of services necessary for the IoT. Extraction of knowledge involves discovering and utilizing resources and incorporating information into models. In addition, it consists of the recognition and analysis of data to determine the most appropriate service. Semantics are integral to the IoT by sending requests to the appropriate resources. Semantic Web technologies, such as the Resource Description Framework (RDF) and Web Ontology Language (OWL), support this requirement [32].

- IoT service: Aiming to facilitate human life, the IoT provides a wide range of services typically delivered as physically isolated vertical solutions. Discovering suitable IoT services faces various challenges and requirements, such as the heterogeneity of accessible services, vast distribution of services, and a highly dynamic environment [33]. Some essential services that IoT provides include healthcare systems [34], remote control [35], transportation [36], education [37], environment monitoring [38], disaster recovery [39], and anomaly detection [40].

- IoT resource: As the IoT entails diverse heterogeneous components, it requires substantial storage and processing to meet users' requests and provide valuable services. Some applications may require complex processing, such as time series analysis, while others may be latency-sensitive. Since the resources of IoT objects are limited in terms of energy, network bandwidth, CPU, and memory, it isn't easy to obtain an ultra-scale and real-world IoT network without taking advantage of cloud platforms or some powerful devices, such as edge or fog nodes and smart gateways [41].

- IoT task: In an IoT-based network, users' requests are organized into two main types of tasks: independent tasks and dependent tasks. The separate tasks, also called atomic tasks, refer to the tasks in which no dependency exists among them. In contrast, the dependent tasks require a specific execution order due to their relationship [42].

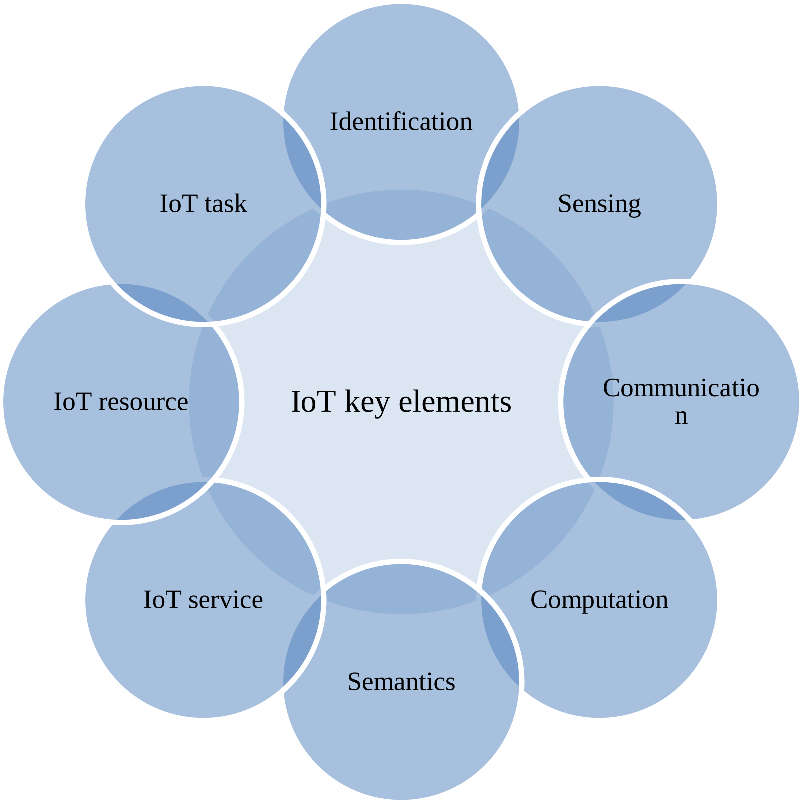

Fig. 2. IoT key elements.

D. Basic Optimization Parameters

- Optimization problem definition: Optimization in the computational domain refers to selecting the most optimal solution, among others, based on various factors [43].
- Search space: In a given problem, the search space refers to a collection of potential or candidate solutions. Each point within the search space represents a specific solution that can be evaluated based on its value or suitability for the problem. The search space encompasses all feasible solutions that can be explored and considered during problem-solving. Analyzing and assessing different points within the search space can identify the most optimal or desirable solution for the problem. [44].
- Objective function: An objective function quantifies the solution to an optimization problem. It refers to an optimization objective, such as minimizing or maximizing metrics (e.g., performance, energy consumption). Optimization variables are transformed into real numbers by the objective function, which can be a single-objective (minimizing response time) or a multi-objective (e.g., minimizing energy consumption and maximizing throughput) [45].
- Population and individual encoding: In meta-heuristic algorithms, individuals within a population represent potential solutions to a specific problem. These individuals are encoded using various data structures, such as Boolean values, strings, or trees. The choice of encoding depends on the nature of the problem being solved and the information required to represent the solutions accurately. In many cases, fixed-length and fixed-order bit strings encode candidate solutions in meta-heuristic approaches. This allows for a consistent and standardized representation of solutions across the population. Fig. 3 illustrates the common encoding methods employed in meta-heuristic algorithms. The encoding can involve binary, discrete, or real values, depending on the specific problem requirements and constraints [46].
- Initialization: Initialization is the process of assigning initial values to the search space in order to create the initial population for a meta-heuristic algorithm. The selection of initial solutions can be made using various methods, one of which is random initialization. In random initialization, individuals are randomly chosen from the search space to form the initial population. This means that the values of the individuals are selected without any specific pattern or bias, providing a diverse starting point for the algorithm. Random initialization helps to explore different regions of the search space and avoids getting stuck in local optima. By introducing randomness in the selection of initial solutions, the algorithm has the potential to discover better solutions throughout the optimization process [47].

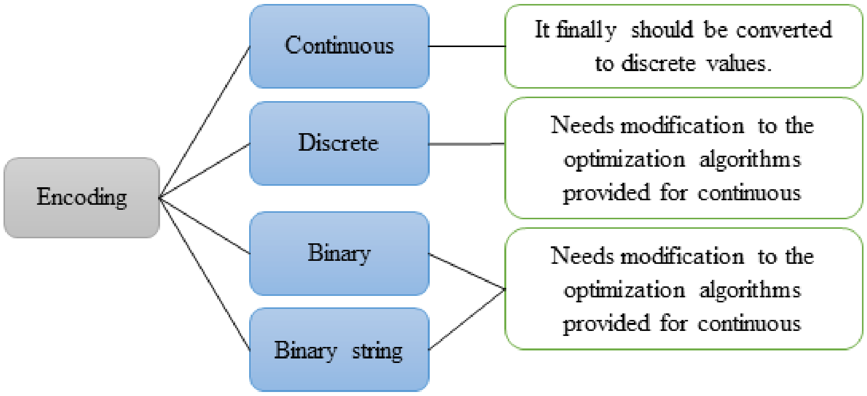
Encoding
Continuous
Discrete
Binary
Binary string
It finally should be converted to discrete values.
Needs modification to the optimization algorithms provided for continuous
Needs modification to the optimization algorithms provided for continuous

Fig. 3. Encoding techniques adopted in IoT routing.

- Termination criteria: Optimization algorithms are typically executed multiple times in order to obtain the best possible results. The way in which these algorithms are executed can be classified as either dynamic or static. In dynamic mode, the iterations of the algorithm continue until the fitness function, which measures the quality of the solutions, fails to improve after a certain number of repetitions. This approach allows the algorithm to adapt and continue searching for better solutions as long as progress is made. If the fitness function stops improving, it suggests that the algorithm has reached a point where further iterations are unlikely to yield significant improvements. In static mode, a fixed number of iterations is predetermined before the optimization algorithm begins. The algorithm will iterate for the specified number of iterations, regardless of whether the fitness function continues to improve or not. Once the predetermined number of iterations is completed, the algorithm terminates. Both dynamic and static termination criteria have their advantages and use cases. Dynamic termination criteria provide flexibility and allow the algorithm to adapt its stopping point based on progress. On the other hand, static termination criteria provide a predetermined stopping point, which can be useful in scenarios where a fixed number of computational resources or time is allocated for the optimization process. The choice of termination criteria depends on the specific problem, available resources, and desired optimization objectives [48].

III. Review of Fault-Tolerant Routing Protocols

This section reviews recent fault-tolerant IoT routing protocols based on important factors such as network lifetime, energy consumption, packet delivery ratio, delay, scalability, availability, and reliability. The strengths and weaknesses of existing works are briefly outlined. Table I provides a side-by-side comparison of the protocols using qualitative parameters for analysis and evaluation. This comparison helps identify efficient methods based on specific requirements.

Agarwal, et al. [49] introduced a multi-objective Deep Reinforcement Learning (DRL)-based approach to fault tolerance in IoT-enabled WSNs. They proposed a double-layer DRL-based approach that incorporates a low-level DRL agent to identify and isolate faulty nodes and a high-level DRL agent to select optimal fault-tolerant routing paths. Their approach was tested on a real-world WSN dataset and showed promising results. The proposed double-layer DRL-based approach enables the WSNs to identify and isolate faulty nodes in a timely manner and select optimal fault-tolerant routing paths with the optimal deployment of resources. The results of the study suggest that the proposed approach is able to detect faulty nodes with high accuracy and minimal overhead, which is essential for the robust and reliable operation of IoT-based WSNs. In addition, it emphasizes the need to transmit data in a reliable manner after fault detection. This is because faulty nodes can cause significant delays in data transmission and lead to data loss. The proposed approach is able to identify such faulty nodes quickly, allowing them to be removed from the network and data transmission to be resumed in a timely manner. The last step is to use a mobile sink to gather data in an energy-efficient manner, which in turn significantly increases the lifetime of the network. The proposed algorithm outperformed the state-of-the-art algorithms in terms of throughput, network lifetime, and fault detection accuracy.

Moreover, it demonstrated superior scalability, allowing it to be implemented in larger networks without compromising performance. However, there are some potential drawbacks to this algorithm as well. One such drawback is that it requires more computational resources than some of the other algorithms. Additionally, it is unclear how well this algorithm will perform in more complex networks.

Cluster-based routing is an effective way to reduce transmission overhead and conserve energy as well as improve transmission quality. It also allows nodes to form clusters which are then used to route data between nodes. By forming clusters, the number of messages that need to be transmitted across the network is reduced, which leads to improved energy efficiency. The Cluster Heads (CHs) are responsible for aggregating and filtering the sensed data before forwarding it to the base station, which further reduces the transmission overhead. Additionally, cluster-based routing allows for better scalability, as nodes can be added or removed from clusters without disrupting the overall network. When one or more CHs fail, the faulty CHs cannot forward data from their serving sensor nodes. This results in insufficient sensed data of the IoT application field being available to the sink node. The IoT applications will be adversely affected by this change. Lin, et al. [50] developed virtual CH formation and flow graph models to tolerate CH failures effectively. By using the virtual CH formation and flow graph models, the resources of all failure-free CHs can be effectively utilized. This allows the CHs to serve as a backup for any faulty CHs, ensuring that packets are still able to be routed properly even in the event of a failure. The experiments conducted show that the approach is successful in providing fault-tolerant routing for IoT WSNs. However, there are also potential drawbacks to this approach. For example, if there are too many CH failures, the system may become overloaded and unable to function properly. Additionally, if the CHs are not properly distributed, this could lead to areas of the network being underserved or not having any CHs at all.

WSNs are vulnerable to physical and environmental factors, such as node failure, interference, and signal attenuation, which can lead to significant delays and packet losses. This can cause disruption to applications that rely on the network for communication, such as smart home applications and automated industrial systems. Topology changes, battery drain, or failure of the wireless node communication module pose the greatest threat to network failure. These factors can cause a decrease in the Quality of Service (QoS) of the network, which can lead to increased latency, data loss, increased energy consumption, and, finally, failure of the application. WSNs can be configured with fault tolerance mechanisms to ensure reliable data acquisition and transmission in IoT applications. Jaiswal and Anand [51] proposed a fault-tolerant cluster-based routing scheme for WSNs combining Grey Wolf Optimization (GWO) and Firefly Optimization (FO) algorithms. An optimal clustering algorithm is implemented using FO, and a routing algorithm is implemented using GWO. To improve network performance and meet quality of service criteria, the algorithm takes into account the energy efficiency and fault tolerance of sensor nodes and CHs. The FO and GWO algorithms were simulated in WSN scenarios in order to compare the results to those from existing techniques. The results showed that the algorithms were able to achieve optimal clustering, improved network performance, and higher energy efficiency while also providing greater fault tolerance and meeting quality of service requirements. However, it should be noted that the FO and GWO algorithms have not been tested in large-scale or real-world scenarios. As such, it is unclear how well they would perform in these types of environments.

The routing strategy proposed by Muhammed, et al. [52] uses multiple clusters and a hierarchical routing structure, which allows it to detect faults and dynamically reroute messages quickly. This reduces the amount of energy consumed and increases the reliability of the network. The simulations show that the proposed method is more reliable and efficient than LEACH and DFTR protocols, as it can detect and reroute traffic more quickly and with less energy consumption. This makes it a better option for networks with large numbers of nodes and dynamic traffic patterns. According to the results, the proposed method performs better than LEACH and DFTR in terms of the network's total energy, the number of nodes left active after a given period, and the network bandwidth. The proposed method is able to balance the energy load more efficiently by selecting the appropriate nodes for activation. It also provides a more balanced traffic pattern across the entire network, which leads to improved energy efficiency and better performance overall. However, there are some potential drawbacks to the proposed method. First, it is more complex than LEACH and DFTR and thus requires more processing power and time to execute. Second, it may be less effective in very large networks with a large number of nodes due to the increased complexity.

WSN-based IoT networks are deployed to collect and transmit data from various sources. IoT sensor nodes within these networks possess diverse properties and characteristics. To efficiently manage data transmission, cluster-based routing is commonly employed. This approach involves dividing the network into clusters, with each cluster being managed by a CH node responsible for aggregating and forwarding the data. However, in the event of one or more CH failures, the sensor nodes within the affected clusters are unable to forward their sensed data to the CHs.

Consequently, the sink node or gateway, which receives and processes the collected data, may not be able to effectively receive the data from the IoT application due to the disrupted data transmission caused by the faulty CHs. This will have a significant impact on the processing of information in this field. To avoid this problem, a more reliable routing protocol is needed that can detect and recover from faults quickly. Such a protocol should be able to detect the failure of CHs and re-assign the sensed data of sensor nodes to other CHs so that the sink node (gateway) can still collect the sensed data of the IoT application in a timely manner. Sivakumar and Vivekanandan [53] present a paired cluster of fault-tolerant disjoint path routing in a path graph and a novel method for solving this dilemma in polynomial time. By providing disjoint paths, the proposed routing model increases the reliability of the system since even if one path fails, the data packets can be rerouted over the other path.

Additionally, the polynomial time solution ensures fast and efficient routing in terms of latency and throughput. The benchmark network simulators measure the latency, throughput, packet delivery ratio, and packet drop ratio in order to accurately evaluate the proposed routing model's performance. In addition to the reliability and efficiency of the system, the simulators also measure the scalability of the model in order to ensure that it can handle large numbers of nodes in a network. However, there are some potential drawbacks to this approach. For example, if the network topology changes, the routing table will need to be recalculated, which could cause delays in communication. Additionally, this approach may not be well suited for highly dynamic networks.

IoT routing problems can be most effectively addressed by swarm intelligence algorithms. Swarm intelligence algorithms are highly adaptive and flexible, allowing them to adjust to changes in the environment and traffic conditions. They also have the ability to make decisions based on real-time data and to quickly update routing paths in response to changes in the network. This makes them an ideal tool for addressing the ever-changing challenges of IoT routing. IoT devices should be fault free in order to improve their efficiency and reliability. To ensure that the devices are error free, many approaches and mathematical models are used. Sharma, et al. [54] proposed the Improved Efficient and Intelligent Fault-Tolerance Algorithm (IEIFTA). IEIFTA can fix any fault at a fast rate, improve efficiency, and prevent data loss if any fault occurs. IEIFTA is a highly reliable and efficient algorithm that can identify the source of any fault and fix it before it can affect the entire system. It also uses a predictive model to anticipate potential faults and take preventive measures to ensure that the system remains error free.

Additionally, it can identify any potential threats and take preventive measures to ensure that the system does not suffer from any data loss. However, there are some potential disadvantages to using IEIFTA. First, it is possible that the algorithm may identify a false positive, which would result in the unnecessary shutdown of a system. Additionally, IEIFTA requires a significant amount of data to be effective, which may not be available in all cases. Finally, the algorithm may be computationally intensive, which could impact its performance in real-time applications.

The proposed routing protocol Bounceur, et al. [55] addresses the need for a leader node in ad hoc networks, particularly in WSNs and IoT networks. The leader node serves various purposes, such as key generation for encryption or decryption and identifying nodes with minimum energy. In their protocol, the process of identifying boundary nodes begins by placing a leader node on the far left of the network. These nodes typically monitor sensitive, dangerous, or inaccessible areas. Since it can be challenging or impossible to intervene if a node fails, the algorithm must be robust and fault-tolerant. In case the leader node fails, it can have catastrophic consequences. To overcome these challenges, the authors propose a new algorithm called DoTRo, which is based on a tree routing protocol. The algorithm involves initiating a flooding process by the local leaders to determine a spanning tree. During this process, the values of the local leaders are routed. If two spanning trees meet, the tree with the best value continues while the other tree stops. The dominant tree that remains becomes the leading tree, and its root becomes the new leader. The DoTRo algorithm has demonstrated high energy efficiency, achieving reduction rates exceeding 85%. It operates effectively even in scenarios where any node can fail or when the network becomes disconnected. The algorithm is designed to be efficient and fault-tolerant in such situations.

Increasingly sophisticated applications, such as fire sprinkler systems, employ multiple sources and sinks, referred to as many-many IoT networks. The development of a fault-tolerant routing protocol is necessary for these critical applications to ensure that messages can be routed around failed nodes without causing significant overhead. Focusing on many-many IoT networks, Grosso and Jhumka [56] propose an efficient distributed fault-tolerance IoT routing scheme based on the Ant Colony Optimization (ACO) algorithm, capable of routing data from multiple sources to multiple sinks. Based on the simulation results, the protocol achieves a delivery rate of more than 80% with a failure rate of only 5%. In comparison with a number of approaches that require periodic maintenance of the topology, this approach is more scalable.

Hasan and Al-Turjman [57] introduce a biologically inspired particle swarm optimization (PMSO) routing approach for constructing, recovering, and selecting k-disjoint paths that can tolerate failure while maintaining the quality-of-service requirements. Using a multi-swarm strategy, the optimal direction for selecting the multipath routing can be determined while all segments of the network exchange message at the same time. Compared with canonical particle swarm optimization (CPSO), the proposed algorithm has demonstrated high-quality solutions. The results indicate that multi-swarm and full PMSO with constriction coefficients are superior to CPSO in terms of sensor count and 89.15% and 86.51% under the ring and mesh topologies, respectively.

Misra, et al. [58] suggest an integrated multi-layer and learning automata-based fault-tolerant routing approach for IoT networks, ensuring packet delivery despite failures affecting both source and destination nodes. As this work involves IoT, the algorithm designed should be highly scalable and should deliver high levels of performance in heterogeneous environments. The learning automata and multi-layer strategies incorporated into the proposed method provide a flexible structure to the algorithm so that a consensus can be achieved across the network using the same standard. As a result, it chooses the optimal action based on the changing environment. In order to conserve energy, all nodes located on unused paths are put to sleep. Simulated results show that the proposed strategy improves the overall energy efficiency of the network and reduces overhead compared to the existing protocols we have used as benchmarks.

In order to address the constraints of IoT systems, numerous cloud-based solutions with effective routing and scalable data storage have been presented. These solutions allow for the efficient handling of large amounts of data and reliable communication between connected devices. They also provide a secure and reliable platform for various applications and services, such as analytics and machine learning. However, as mobile networks and communication links are unpredictable, most of the solutions may not be suitable for realistic applications and will result in path failure and an increase in resource consumption. Thus, data forwarding can only be reliable and valuable when the algorithms proposed are trusted and aware, have low overheads, and consume a balanced amount of energy across the nodes. Haseeb, et al. [59] suggested a fault-tolerant supervised routing scheme in the context of IoT trust management in order to enhance trustworthiness and collaboration within smart cities. A reliable and optimized network structure is established by each node evaluating its neighbors' behavior.

Furthermore, a fault-tolerant relaying system is provided by employing supervised machine learning without imposing additional overhead. In addition, it eliminates the additional workload associated with determining the optimal decision and training the IoT system to balance network costs. Finally, a secure algorithm with secured keys is proposed to ensure the privacy and authentication of the relaying system. Compared with previous work, the proposed model has shown significant improvements in performance. However, the proposed model has not been proven to be secure against all possible attacks. Furthermore, the algorithm has not been tested on a large scale.

Chanak, et al. [60] present a fault-tolerant routing protocol for IoT-driven WSNs, which significantly enhances the QoS of these networks. They develop a new multi-population z-test-based fault detection method to identify faulty devices in the network. This method is based on the analysis of the data collected from the network and uses a combination of the z-test and the chi-square test to detect faulty nodes. The protocol also utilizes a novel routing algorithm to reroute the data around faulty nodes in order to ensure that the data is delivered to its destination in a timely manner with minimal disruption. The proposed routing protocol has been designed to provide fault tolerance and flexibility, allowing the reuse of faulty nodes in the network. The experiments conducted to test the protocol demonstrate its efficiency and effectiveness in various areas, including fault detection accuracy, energy consumption, and network lifetime. The results of these experiments are then compared with the state-of-the-art algorithms to show the effectiveness of the proposed scheme. However, there are some potential drawbacks to this scheme that should be considered. First, the reliance on faulty nodes could lead to increased network instability. Second, the additional overhead required to maintain the fault tolerance could lead to higher energy consumption and shorter network lifetimes.

In Industrial 4.0, safety is one of the main concerns, where various physical parameters are monitored to prevent uncertain events. A natural disaster, such as a fire or the leakage of harmful gases, can cause tremendous damage to both life and property in the industrial sector. Industrial IoT (IIoT) is employed to monitor such natural calamities and take appropriate action in a timely fashion. The IIoT, however, is susceptible to sensor failures as a result of energy depletion and hardware malfunctions. This results in a significant reduction in the network's reliability. Kaur and Chanak [61] propose a fault-tolerant framework in which faults in the WSN-assisted IIoT in the form of node failures and link failures are identified and handled efficiently. The proposed framework uses a distributed consensus-based approach to identify and detect faults in the WSN-assisted IIoT. It also uses a fault-tolerant routing protocol to route traffic around the faulty nodes, thus ensuring that the IIoT remains reliable even in the case of a node or link failure. The proposed scheme has been extensively simulated and has been found to outperform other schemes as measured by recovery speed, communication delay, network lifetime, throughput, and energy consumption. Although the proposed scheme has been found to have many benefits, there are also some drawbacks to consider. For example, the scheme requires more energy to operate than other schemes, which may not be feasible for some IIoT applications. Additionally, the scheme may not be able to handle all types of node and link failures, which could lead to network outages.

The implementation of various technologies for industrial information delivery and process control has been hindered by the challenges of increased complexity and associated faults. These factors have posed obstacles to achieving reliable and timely network activation. This is because industrial systems are usually comprised of multiple components that can be affected by different environmental conditions, as well as by the transmission of inaccurate or incomplete data. As a result, the transmission of reliable data to activate the timely network is challenging. In order to transfer this data from one node to another, there must be no faults or delays between the nodes. As a solution to this problem, Vishal Sharad, et al. [62] developed a new algorithm to communicate messages between different services without any delays. Their algorithm, based on the MoO4RPL objective, simulates IoT with mobile sink nodes in the network. The strategy consists of several phases, including topology generation, route discovery, communication, and route maintenance. In the multi-objective route discovery phase, the algorithm constructs the network topology and calculates a rank based on factors, such as energy, trust, delay, fault tolerance, and link quality. The proposed method with fitness function factors is used to select the optimal route during the communication phase. The fitness function factors are used to evaluate each route based on metrics, such as distance, energy, link quality, and trust. The route with the best metrics is determined to be the optimal route and is then maintained during the route maintenance phase. However, the proposed method may not be feasible in real-world scenarios due to the computational overhead required to calculate the fitness function factors for each route. In addition, the proposed method does not consider dynamic changes in the network, which can lead to suboptimal routing decisions.

Pankajavalli and Karthick [63] introduced a novel approach known as the Free Poisson Law method. This technique addresses the challenge of paired fault-tolerant cluster routing in a data flow graph by establishing disjoint routes. The technique utilizes the idea of assigning probabilistically independent Poisson-distributed weights to the edges of the graph. This allows for a simple and efficient algorithm that can find a pair of disjoint routes between two nodes using a single pass through the graph. The Free Poisson Law technique takes advantage of the fact that Poisson-distributed weights can be used to represent the probability of a given edge being selected. This allows the algorithm to select a pair of disjoint routes that are both likely to be successful. In addition, the algorithm only needs to take one pass through the graph, making it highly efficient. The primary objective of the algorithm is to minimize latency, energy consumption, dissipated energy, and functional complexity, thereby enhancing the packet delivery ratio, throughput, and fault detection rate. However, the proposed algorithm does not guarantee that the selected pair of routes will be successful. In addition, the algorithm may not be able to consider all the factors that can affect packet delivery, such as network congestion.

TABLE I. A SIDE-BY-SIDE COMPARISON OF FAULT-TOLERANT IOT ROUTING PROTOCOLS

References	Qualitative metrics						
	Packet delivery ratio	Network lifetime	Energy consumption	Delay	Scalability	Availability	Reliability
[49]		↑	↓	↓			↑
[50]					↑	↑	
[51]		↑	↓			↑	
[52]	↑		↓				↑
[53]	↑			↓		↑	
[54]					↑		↑
[55]			↓		↑	↑	
[56]	↑						↑
[57]			↓		↑	↑	
[58]			↓		↑		↑
[59]	↑						↑
[60]		↑	↓				
[61]	↑	↑	↓	↓			↑
[62]			↓				↑
[63]	↑		↓	↓		↑	

↑ = Increased and ↓= Decreased

IV. DISCUSSION

The reviewed protocols encompass diverse approaches, each offering unique advantages and addressing specific challenges. Throughout our analysis, we observed that fault tolerance plays a pivotal role in ensuring the reliability and resilience of IoT networks, particularly in the face of various failures, topology changes, and node malfunctions. One promising approach we encountered is the use of Deep Reinforcement Learning (DRL)-based methods for fault tolerance in IoT-enabled Wireless Sensor Networks (WSNs) [36]. These approaches leverage a double-layer DRL agent to identify and isolate faulty nodes and select optimal fault-tolerant routing paths. The results indicate high accuracy in fault detection with minimal overhead, enabling robust and reliable operation of IoT-based WSNs. However, some of these algorithms may require more computational resources and might not perform optimally in highly complex networks.

Cluster-based routing emerged as an effective method for conserving energy and improved transmission quality by forming clusters and aggregating data before forwarding it to the base station. To handle potential CH failures, virtual CH formation and flow graph models are proposed to effectively utilize resources from failure-free CHs, ensuring continuity in data routing. However, the proper distribution of CHs and excessive CH failures could still pose challenges to the overall system's stability and efficiency. Multi-cluster and hierarchical routing structures provide a rapid fault detection and dynamic rerouting mechanism, reducing energy consumption and increasing network reliability. While such protocols demonstrate superiority in terms of network performance, they may require more processing power and might be less effective in very large networks.

Swarm intelligence algorithms, known for their adaptability to changing environments, have shown the potential to address IoT routing problems effectively. Such algorithms can anticipate faults, fix them promptly, and make decisions based on real-time data, making them suitable for the dynamic challenges in IoT routing. However, their effectiveness in large-scale or real-world scenarios remains to be validated.

Cloud-based fault-tolerant routing solutions have been proposed to address the constraints of IoT systems. These solutions aim to handle large data volumes and offer reliable communication between devices. Supervised routing schemes based on trust management enhances trustworthiness and collaboration in smart cities. However, ensuring security against all possible attacks and real-world testing in large-scale environments are still areas of concern. In the Industrial IoT (IIoT) context, a distributed consensus-based approach coupled with fault-tolerant routing protocols has been introduced to handle sensor failures and link disruptions. While the proposed framework demonstrates improved efficiency and effectiveness, it may require additional energy, and its ability to handle all types of node and link failures requires further scrutiny.

Finally, the Free Poisson Law method presents a novel technique for establishing disjoint routes in paired fault-tolerant cluster routing in data flow graphs. This technique provides an efficient algorithm for finding probabilistically independent Poisson-distributed weights on graph edges, minimizing latency and enhancing packet delivery ratio. However, ensuring successful routing and considering all factors affecting packet delivery requires further investigation. In conclusion, our review of fault-tolerant routing mechanisms in IoT-based networks has revealed a rich landscape of innovative approaches that tackle the challenge of ensuring reliable and resilient communication. Each approach offers unique strengths and limitations, making it crucial to select the most appropriate method based on the specific application requirements and network characteristics. We recommend further research to explore hybrid approaches, combining the strengths of different techniques and conducting extensive real-world testing to validate the performance and scalability of these methods. Additionally, addressing security concerns and investigating the potential trade-offs between energy efficiency and fault tolerance is vital for optimizing network performance in IoT environments. By building upon these findings and pursuing further research in fault tolerance, we envision the development of more robust and efficient IoT networks, unlocking their full potential in diverse applications and domains.

V. CONCLUSION

The IoT is revolutionizing various facets of our lives and leading us towards enhanced societies in the future. WSNs play a crucial role in the IoT landscape. Energy conservation, resilience, and reliability are three essential requirements for WSNs. The ability of WSNs to tolerate faults ensures their reliability and resilience in the event of failure. The most frequent causes of network failure are typically attributed to changes in network topology, node failure caused by battery depletion, and the malfunctioning of wireless communication modules within nodes. These factors have been identified as the primary culprits responsible for disrupting the seamless operation of networks. It is essential to address these challenges to ensure the reliability and stability of network connectivity in various environments. This paper reviewed state-of-the-art fault-tolerant IoT routing protocols concerning critical factors such as packet delivery ratio, network lifetime, energy consumption, delay, scalability, availability, and reliability. Based on the analysis of the effective techniques, recommendations are made for further research. One promising direction lies in exploring machine learning-based approaches, leveraging real-time data to dynamically adapt network routing decisions. Additionally, investigating the integration of blockchain technology could enhance the security and reliability of fault-tolerant routing in IoT systems. Efforts towards developing lightweight fault-tolerant routing algorithms, mindful of the resource constraints and energy limitations of IoT devices, will greatly benefit the scalability and practicality of fault-tolerant IoT networks. Moreover, tailored fault-tolerant solutions catering to specific IoT applications, such as smart cities, industrial automation, or healthcare systems, can better address unique challenges and requirements in those domains.

REFERENCES

[1] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy-efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, p. e6959, 2022.

[2] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023.

[3] A. Peivandizadeh and B. Molavi, "Compatible authentication and key agreement protocol for low power and lossy network in IoT environment," Available at SSRN 4194715, 2022.

[4] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging internet of things marketplace from an industrial perspective: A survey," IEEE transactions on emerging topics in computing, vol. 3, no. 4, pp. 585-598, 2015.

[5] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic," Peer-to-Peer Networking and Applications, pp. 1-21, 2022.

[6] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[7] M. Ilbeigi, A. Morteza, and R. Ehsani, "Emergency Management in Smart Cities: Infrastructure-Less Communication Systems," in Construction Research Congress 2022, pp. 263-271.

[8] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[9] H. Kosarirad, M. Ghasempour Nejati, A. Saffari, M. Khishe, and M. Mohammadi, "Feature Selection and Training Multilayer Perceptron Neural Networks Using Grasshopper Optimization Algorithm for Design Optimal Classifier of Big Data Sonar," Journal of Sensors, vol. 2022, 2022.

[10] M. Bagheri et al., "Data conditioning and forecasting methodology using machine learning on production data for a well pad," in Offshore Technology Conference, 2020: OTC, p. D031S037R002.

[11] S. R. Abdul Samad et al., "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," Electronics, vol. 12, no. 7, p. 1642, 2023.

[12] B. M. Jafari, X. Luo, and A. Jafari, "Unsupervised Keyword Extraction for Hashtag Recommendation in Social Media," in The International FLAIRS Conference Proceedings, 2023, vol. 36.

[13] R. Soleimani and E. Lobaton, "Enhancing Inference on Physiological and Kinematic Periodic Signals via Phase-Based Interpretability and Multi-Task Learning," Information, vol. 13, no. 7, p. 326, 2022.

[14] W. Anupong et al., "Deep learning algorithms were used to generate photovoltaic renewable energy in saline water analysis via an oxidation process," Water Reuse, vol. 13, no. 1, pp. 68-81, 2023.

[15] C. Han and X. Fu, "Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model," Frontiers in Business, Economics and Management, vol. 8, no. 2, pp. 51-54, 2023.

[16] M. Javidan, M. Yazdchi, Z. Baharlouei, and A. Mahnam, "Feature and channel selection for designing a regression-based continuous-variable emotion recognition system with two EEG channels," Biomedical Signal Processing and Control, vol. 70, p. 102979, 2021.

[17] S. Mahmoudinazlou and C. Kwon, "A Hybrid Genetic Algorithm for the min-max Multiple Traveling Salesman Problem," arXiv preprint arXiv:2307.07120, 2023.

[18] S. Mahmoudinazlou and C. Kwon, "A Hybrid Genetic Algorithm with Type-Aware Chromosomes for Traveling Salesman Problems with Drone," arXiv preprint arXiv:2303.00614, 2023.

[19] V. Ashrafimoghari and J. W. Suchow, "A game-theoretic model of the consumer behavior under pay-what-you-want pricing strategy," arXiv preprint arXiv:2207.08923, 2022.

[20] M. Shahin et al., "Cluster-based association rule mining for an intersection accident dataset," in 2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2021: IEEE, pp. 1-6.

[21] A. J. Onumanyi, A. M. Abu-Mahfouz, and G. P. Hancke, "Low Power Wide Area Network, Cognitive Radio and the Internet of Things: Potentials for Integration," Sensors, vol. 20, no. 23, p. 6837, 2020.

[22] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9326-9337, 2019.

[23] D. Burrus, "The internet of things is far bigger than anyone realizes," Burrus Research via Wired, 2014.

[24] F. Insight, "Internet of Things-From theory to reality," Forbes Insight, Jersey City, 2017.

[25] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies,

protocols, and applications," IEEE communications surveys & tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[26] H. Herath, "Internet of Things (IoT) enable designs for identify and control the COVID-19 pandemic," in Artificial Intelligence for COVID-19: Springer, 2021, pp. 423-436.

[27] L. Zhu, K. Gai, and M. Li, "Blockchain and Internet of Things," in Blockchain Technology in Internet of Things: Springer, 2019, pp. 9-28.

[28] Y. Liu et al., "Zero-bias deep learning for accurate identification of Internet-of-Things (IoT) devices," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2627-2634, 2020.

[29] Z. Sisi and A. Souri, "Blockchain technology for energy-aware mobile crowd sensing approaches in Internet of Things," Transactions on Emerging Telecommunications Technologies, p. e4217, 2021.

[30] A. Khanna and S. Kaur, "Internet of things (IoT), applications and challenges: a comprehensive review," Wireless Personal Communications, vol. 114, no. 2, pp. 1687-1762, 2020.

[31] M. Mahbub, "IoT Ecosystem: Functioning Framework, Hierarchy of Knowledge, and Intelligence," in Artificial Intelligence-based Internet of Things Systems: Springer, 2022, pp. 47-76.

[32] I. H. Sarker, "Smart city data science: Towards data-driven smart cities with open research issues," Internet of Things, vol. 19, p. 100528, 2022.

[33] I. Mashal and O. Alsaryrah, "Fuzzy analytic hierarchy process model for multi-criteria analysis of internet of things," Kybernetes, 2019.

[34] V. Hayyolalam, M. Aloqaily, O. Ozkasap, and M. Guizani, "Edge Intelligence for Empowering IoT-based Healthcare Systems," arXiv preprint arXiv:2103.12144, 2021.

[35] S.-R. Yang, S.-C. Yuan, Y.-C. Lin, and I.-F. Yang, "DTMFTalk: a DTMF-Based Realization of IoT Remote Control for Smart-Home Elderly Care," Mobile Networks and Applications, pp. 1-12, 2020.

[36] E. B. Priyanka, C. Maheswari, and S. Thangavel, "A smart-integrated IoT module for intelligent transportation in oil industry," International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, p. e2731, 2020.

[37] P. Isaias, "Model for the enhancement of learning in higher education through the deployment of emerging technologies," Journal of Information, Communication and Ethics in Society, 2018.

[38] S. L. Ullo and G. Sinha, "Advances in smart environment monitoring systems using iot and sensors," Sensors, vol. 20, no. 11, p. 3113, 2020.

[39] E. Andrade and B. Nogueira, "Dependability evaluation of a disaster recovery solution for IoT infrastructures," The Journal of Supercomputing, vol. 76, no. 3, pp. 1828-1849, 2020.

[40] F. Cauteruccio et al., "A framework for anomaly detection and classification in Multiple IoT scenarios," Future Generation Computer Systems, vol. 114, pp. 322-335, 2021.

[41] F. C. Delicato, P. F. Pires, and T. Batista, "The resource management challenge in IoT," in Resource management for Internet of Things: Springer, 2017, pp. 7-18.

[42] A. Hussain, S. Manikanthan, T. Padmapriya, and M. Nagalingam, "Genetic algorithm based adaptive offloading for improving IoT device communication efficiency," Wireless Networks, vol. 26, no. 4, pp. 2329-2338, 2020.

[43] M. Kumar and S. C. Sharma, "PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing," Neural Computing and Applications, vol. 32, no. 16, pp. 12103-12126, 2020.

[44] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: theory, analysis, and application," Swarm and Evolutionary Computation, vol. 63, p. 100868, 2021.

[45] K. R. Wagiman, M. N. Abdullah, M. Y. Hassan, and N. H. M. Radzi, "A new metric for optimal visual comfort and energy efficiency of building lighting system considering daylight using multi-objective particle swarm optimization," Journal of Building Engineering, vol. 43, p. 102525, 2021.

[46] Y. Wu, "A survey on population-based meta-heuristic algorithms for motion planning of aircraft," Swarm and Evolutionary Computation, vol. 62, p. 100844, 2021.

[47] D. Oliva and M. A. Elaziz, "An improved brainstorm optimization using chaotic opposite-based learning with disruption operator for global optimization and feature selection," Soft Computing, vol. 24, no. 18, pp. 14051-14072, 2020.

[48] T. Bonny and M. Kashkash, "Highly optimized Q-learning-based bees approach for mobile robot path planning in static and dynamic environments," Journal of Field Robotics, vol. 39, no. 4, pp. 317-334, 2022.

[49] V. Agarwal, S. Tapaswi, and P. Chanak, "Intelligent fault-tolerance data routing scheme for IoT-enabled WSNs," IEEE Internet of Things Journal, vol. 9, no. 17, pp. 16332-16342, 2022.

[50] 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.

[51] K. Jaiswal and V. Anand, "FAGWO-H: A hybrid method towards fault-tolerant cluster-based routing in wireless sensor network for IoT applications," The Journal of Supercomputing, vol. 78, no. 8, pp. 11195-11227, 2022.

[52] T. Muhammed, R. Mehmood, A. Albeshri, and A. Alzahrani, "HCDSR: A hierarchical clustered fault tolerant routing technique for IoT-based smart societies," Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies, pp. 609-628, 2020.

[53] S. Sivakumar and P. Vivekanandan, "Efficient fault-tolerant routing in IoT wireless sensor networks based on path graph flow modeling with Marchenko–Pastur distribution (EFT-PMD)," Wireless networks, vol. 26, pp. 4543-4555, 2020.

[54] A. K. Sharma, K. Kanhaiya, and J. Talwar, "Effectiveness of Swarm Intelligence for Handling Fault-Tolerant Routing Problem in IoT," Swarm Intelligence Optimization: Algorithms and Applications, pp. 325-341, 2020.

[55] A. Bounceur, M. Bezoui, L. Lagadec, R. Euler, L. Abdelkader, and M. Hammoudeh, "Dotro: A new dominating tree routing algorithm for efficient and fault-tolerant leader election in wsns and iot networks," in Mobile, Secure, and Programmable Networking: 4th International Conference, MSPN 2018, Paris, France, June 18-20, 2018, Revised Selected Papers 4, 2019: Springer, pp. 42-53.

[56] J. Grosso and A. Jhumka, "Fault-Tolerant Ant Colony Based-Routing in Many-to-Many IoT Sensor Networks," in 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), 2021: IEEE, pp. 1-10.

[57] M. Z. Hasan and F. Al-Turjman, "Optimizing multipath routing with guaranteed fault tolerance in Internet of Things," IEEE Sensors Journal, vol. 17, no. 19, pp. 6463-6473, 2017.

[58] S. Misra, A. Gupta, P. V. Krishna, H. Agarwal, and M. S. Obaidat, "An adaptive learning approach for fault-tolerant routing in Internet of Things," in 2012 IEEE Wireless Communications and Networking Conference (WCNC), 2012: IEEE, pp. 815-819.

[59] K. Haseeb, T. Saba, A. Rehman, Z. Ahmed, H. H. Song, and H. H. Wang, "Trust management with fault-tolerant supervised routing for smart cities using internet of things," IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22608-22617, 2022.

[60] P. Chanak, I. Banerjee, and S. Bose, "An intelligent fault-tolerant routing scheme for Internet of Things-enabled wireless sensor networks," International Journal of Communication Systems, vol. 34, no. 17, p. e4970, 2021.

[61] G. Kaur and P. Chanak, "An Intelligent Fault Tolerant Data Routing Scheme for Wireless Sensor Network-assisted Industrial Internet of Things," IEEE Transactions on Industrial Informatics, 2022.

[62] H. Vishal Sharad, S. R. Desai, and K. Y. Krishnrao, "SAOA: Multi-Objective Fault-Tolerance Based Optimized RPL Routing Protocol in Internet of Things," Cybernetics and Systems, pp. 1-22, 2022.

[63] P. Pankajavalli and G. Karthick, "Efficient Data Flow Graph Modeling Using Free Poisson Law for Fault-Tolerant Routing in Internet of Things," in Computer Networks and Inventive Communication Technologies: Proceedings of Fifth ICCNCT 2022: Springer, 2022, pp. 475-487.