



A Reliable Protocol for Data Aggregation and Optimized Routing in IoT WSNs based on Machine Learning

Neeraj Chandnani^{1,2} · Chandrakant N. Khairnar³

Accepted: 21 March 2023 / Published online: 15 April 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Data Aggregation for IoT-WSN, based on Machine Learning (ML), allows the Internet of Things (IoT) and Wireless Sensor Networks (WSN) to send accurate data to the trusted nodes. The existing work handles the dropouts well but is vulnerable to different attacks. In the proposed research work, the Data Aggregation (DA) based on Machine Learning (ML) fails the untrusted aggregator nodes. In the attack scenario, this paper proposes a Machine Learning Based Data Aggregation and Routing Protocol (MLBDARP) that verifies the network nodes and DA functions based on ML. This work is to authenticate the nodes to support the MLBDARP, a novel secret shared authentication protocol, and then aggregate using a secure protocol. MLBDARP types of the ML algorithm, such as Decision Trees (DT) and Neural Networks (NN). ML helps determine the probability of a successful Packet Delivery Ratio (PDR). This proposed ML model uses predictability value, Energy Consumption (EC), mobility, and node position. Simulation results proved that the proposed protocol of MLBDARP outperforms Differentiated Data Aggregation Routing Protocol (DDARP) and Weighted Data Aggregation Routing Protocol (WDARP) with Quality of Service (QoS) parameters of Network Throughput (NT), Routing Overhead (RO), End-to-End Delay (EED), Packet Delivery Ratio (PDR) and Energy Consumption (EC).

Keywords Machine learning · Neural network · Data aggregation · Routing · IoT-WSN · Decision tree · Security and privacy

✉ Neeraj Chandnani
chandnani.neeraj@gmail.com

Chandrakant N. Khairnar
cnkhairnar@gmail.com

¹ Institute of Advance Computing, SAGE University, Indore, India

² Military College of Telecommunication Engineering, Mhow, Research Centre, Affiliated to Devi Ahilya Vishwavidyalaya, Indore, Madhya Pradesh, India

³ Faculty of Communication Engineering, Military College of Telecommunication Engineering, Mhow, Madhya Pradesh, India

1 Introduction

The most critical application of ML, for example, at Amazon, IBM Watson, Azure, Google Cloud, and Microsoft, is Data Aggregation (DA) in edge computing, which collects a maximum of users along with their data [1–3]. Moreover, they pose a risk to the privacy of the users [4–6]. As shown in Fig. 1, Machine Learning (ML) helps the AN collect the value updates instead of accurate data [7–9]. For example, the online user sends the data of the local ML to the firms, which trains the system to predict the user’s interest in the future. Hospitals offer updates of ML on healthcare records to the World Health Organization (WHO) for new diagnostic systems, and financial corporations record the transaction log to improve the scam [10, 11]. Moreover, the users are disturbed about the information which the companies are taking to improve their expected results [12–14].

To increase security, companies like Google [15] and Apple [16, 17] have implemented a randomized response system which injects noise into users’ data with the help of a randomized method. Although, they provide weak privacy since the aggregator controls the users’ data when it receives a high volume of noisy data [17, 18]. Different secure algorithms have been proposed by researchers, which are based on secure computation [17, 19–21], Dining Cryptographer networks (DC-nets) [5, 22–24], Homomorphic Encryption (HE) [25–31] and Differential Privacy (DP) [5, 32–34].

Moreover, the existing secure algorithms, such as DC-nets and HE, accept high Computation Overhead (CO). Hence, this work cannot apply these algorithms to real-time challenges where nodes’ communication and CO are essential. On the other hand, the DP protocols are lightweight compared to others, but they are not secure as the malicious node fails during execution. A recent researcher [5] considers node failure vulnerable to attacks where Malicious Nodes (MN) communicate updates of the parameter such that they are considered abnormal by the Aggregator Node (AN).

To remove the ambiguity in the current work, this work emphasizes performing DA using ML in a more real-world system, as shown in Fig. 1. The MNs transmit the parameters’ incorrect updates to ANs to perturb the complete updates of parameters by AN. Although disobeying such a method usually gains nodes [17, 35], A selfish MN, for example, avoids the AN by sending an incorrect update of local ML on network traffic data, preventing other nodes from following the same routing path [35–37]. In the other example,

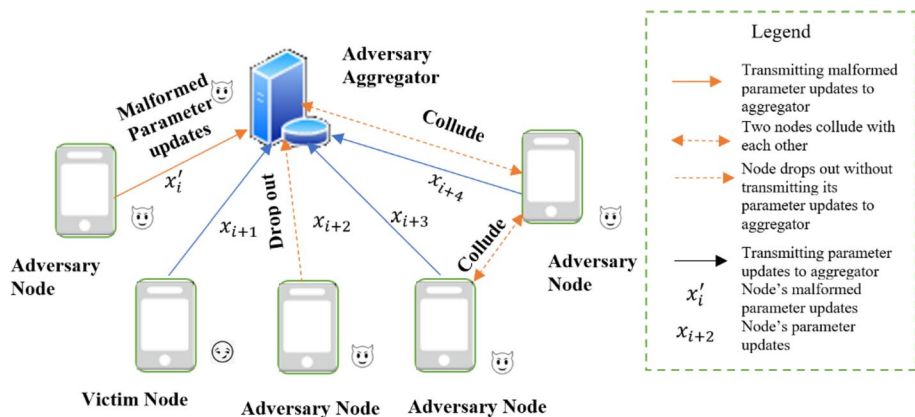


Fig. 1 ML-based data aggregation

an application developer uses the data to increase the common application's rank in the stores [17]. Also, to reveal other nodes' data privacy, the MN terminates and gets involved secretly with the nodes, and the aggregator becomes malicious [17, 35, 62].

For the practical scenario of IoT-WSN, this work proposes a protocol called Machine Learning Based Data Aggregation and Routing Protocol (MLBDARP), which verifies the node's DA based on ML and prevents the nodes and network from malicious attacks. It also defends against colluding nodes and AN, resulting in the discontinuation of MNs. The main objective of the proposed protocol is to authenticate the node's data using a protocol based on secured sharing and aggregate updates of nodes' parameters in a secure manner with the help of a secure DA protocol. It is not the direct method; the following difficulties have been overcome: First, verifying nodes' data without revealing it to the malicious aggregator and other nodes is challenging. Secondly, it is challenging to implement a protocol for DA that takes care of both agreement attacks and MN failures. To address the tests stated above, the following vital contributions are completed:

1. In this paper, a Secret Shared Authentication Protocol (SSAP) is made to check the validity of the node's data in a safe mode. When a node updates the parameter for the aggregator, it shares the proof with the aggregator and the neighbouring nodes. With the help of transmitting a few bytes of data to the neighbouring nodes, the AN authenticates the node's data without fully knowing the data.
2. This work proposes a protocol that securely performs the DA method and ANs parameter updates and prevents them from colluding with other nodes and MN failures. The proposed protocol neither implements computational encryption nor depends on secure communication paths. In other words, this research says that each node locally generates a secret code, which this work call a key, for encrypting its parameter updates, and the AN can know about the parameter updates without having the critical information.
3. In this article, the IoT-WSN was implemented, which protects the node data from malicious attacks in DA based on ML. The simulation results proved that the proposed protocol preserves node data security and allows for acceptable overhead.

IoT-WSNs are those networks in which the performance of the link is not secure, and the resources available to the nodes are limited. So, the routing in IoT-WSN is a challenging task since nodes carry the data packets until they find the trusted forward nodes that carry them to the desired destination (sink node) with a minimum End-to-End Delay (EED).

These protocols are classified as infrastructure-based and infrastructure-less-based routing protocols. In the former case, the infrastructure must help the nodes pass the data packets to the sink. In the latter case, there is no infrastructure, and the communication time is used to forward the network data. This article proposes a novel infrastructure-less protocol for IoT-WSN, the improved version of the protocol discussed [38]. The proposed protocol uses the ML methodology to train itself on factors like buffer size, node acceptance and speed, hop count, EC, and Packet Delivery Ration (PDR). The method is accomplished based on network routing, and a mathematical model is formed which computes the probability that a node will successfully PDR to the sink node. The computed value is used for the decision on the next hop.

This article is structured as follows: Sect. 2 discusses the related works. In Sect. 3, the proposed MLBDARP is explained. Protocol implementation, simulation results, and their comparison with the existing protocols are described in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Literature Review

The multi-node computation [39] developed a security protocol based on the Basal Metabolic Rate (BMR) algorithm, which prevents malicious and corrupted nodes. Research by [20] discusses secure third-party computations where users and servers are malicious or untrusted. Although the researchers in [19] and [21] measured the invalid data of the MNs in the computation algorithm, researchers in [40] and [41] implemented protocols in insecure networks. Moreover, these protocols have high CO because the nodes share the secured data with other nodes, which confirms their robustness. IoT-WSN has a resource limitation. When the server is reliable and functions well, even when the other nodes are MN, the Prior protocol [17] prevents the node's data security. Although, work considers the situation with passive servers.

DC-network-based research [23, 24, 42] allows the nodes to privately share their data with the help of pairwise cloaking inputs without one-to-one communication. Moreover, it is susceptible to untrusted nodes, which corrupt the data and block communication [5]. Also, the research [42] implemented a novel DC-net that recognizes dishonest nodes with a very high probability with the help of a further round. Similarly, [22] discusses the implementation of public-key encryption and data proofs to detect the misbehaviour of the nodes. It requires a high concentration of CO. On the other hand, our proposed work is lightweight in that the failures are set up without iterations.

HE performs low when a node fails, but it has a high CO [28, 29], and [31]. The authors in [25–27] implemented an encryption method and differential confidentiality for computing statistics and operational failures. Although, these researchers accept trusted agents, which are not present in the IoT-WSN scenario. The research work in [43] discusses secure computation, in which each node communicates with a server but cannot control failures. Research-based DP [44] implements an intelligent system that gathers the nodes' data and provides DP. The authors [32] proposed the PrivEx system, which aggregates the statistics from anonymous networks. The research work in [33] uses an Enhanced File Transfer (EFT) encryption algorithm to enhance DP for the aggregated results. The research work in [45] presents PrivCount, which aggregates across relays and private results differentially.

Moreover, these algorithms are completed when nodes discontinue the recovery process. The proposed work in [5, 34] has considered failures, but they are vulnerable to malicious attacks. On the other hand, this proposed protocol considers the failures, authenticates the nodes, and prevents the IoT-WSN from malicious attacks.

Various routing processes have been suggested in the current research work. The most important ones are discussed. In [46], the author proposed an algorithm where the transmitter node sends many copies of the network that it plans to send to the sink. It is done by sending a data copy to each node with which it connects. This process is repeated until a copy of the data is delivered to the sink. It has a high PDR rate and high resource consumption. In [47], the Hop protocol is proposed, in which the context of the nodes is stored in the identity table, and the history table saves characteristics from the identity table of neighbouring nodes. The idea is that the transmitted node passes the copies of data, and the PDR is computed. In [48], the author proposes a routing protocol that uses contextual information to select the next hop. The Markov predictor determines the next best hop based on the node's behaviour information. In [49], the routing protocol depends on computing the delivery predictability table, which keeps track of the successful PDR from the transmitter to the sink node. The packet is forwarded to the nodes with high predictability values. In [38], an improvement is presented using a weighted function to compute the

node's delivery probability. In [50], a Distance Routing Protocol (DRP) is proposed that relies on encounters and the node's distance from the sink node to select the next hop. The ratio of these two variables decides the selection of the next hop.

In the research work mentioned in [63], the authors have applied Fuzzy Logic (FL) to WSN and shown that it improves energy performance. In [64], authors have discussed a method in which the training time is reduced, resulting in improved presentation and network overhead. The proposed research in [65] is a practical route assessment to minimize the communication between the Sensor Nodes (SN), thereby reducing EED and EC and increasing the WSN's lifetime. In [66], problems related to routing are addressed; hence, the network lifetime improvement is shown when two problems are addressed. The authors in [67] have proposed adaptive Routing for In-Network Aggregation (RINA) for WSNs. Q-learning forms a routing tree using residual energy, increasing the distance between nodes and link strength and, thus, the network lifetime. The proposed research in [68] discusses improved network performance, secure routing services, and reduced EC. In [69], the detection and removal of the MN from the network are discussed with secure routing and less EC. The research based on the researchers' particulars of data collection is summarised in Table 1.

This proposed work ML is useful to DA and routing, and it has shown improvement not only in terms of the five major QoS parameters (*i.e.*) NT, EED, PDR, EC, and RO) but also in terms of security, computation, and RO, and this proposed work has outperformed other existing research works.

3 Proposed Machine Learning Based Data Aggregation and Routing Protocol (MLBDARP)

The MLBDARP consists of DA and routing algorithms based on ML, and the same has been discussed in Sects. 3.1 and 3.2.

3.1 Proposed MLBDARP

Let us take the example of a company that aggregates the user's activities for training the Recommender System (RS), which can predict the user's future interests. Each user has a record of their actions saved on their device and has personal information, like, their job, health, and lifestyle. Despite sending the exact data in DA based on ML, the user only requires updated parameters that contain minimum data compared with their exact data. The aggregator receives the user's variable updates and calculates the variable portion's weighted averages for training the recommender using Stochastic Gradient Descent (SGD). Even though each user parameter update contains less information than the activity data, some studies [12, 13] have revealed that variable updates allow attackers to obtain precise, private data.

3.1.1 Internet of Things-Wireless Sensor Network Attack Model

ML-based DA has the following challenges:

- a. *Malicious Nodes*: Nodes that are supposed to be malicious achieve the protocol honestly, as shown in the current work [5, 20, 33]. This work says that nodes honestly per-

Table 1 Comparative analysis of IoT-WSN with DA

| Authors | Objectives | Research design and drawbacks |
|--|---|--|
| Marcolla et al. [70] | Accepting node failures is the experiment's primary focus. For WSNs to handle node failures, the study investigated network topology evaluation methods and classified fault-tolerance methods as either proactive or reactive | Fault-tolerant DA algorithms were not addressed in the research. The investigation is restricted to node failures. Process faults are unnoticed |
| Al Badawi et al. [71] | Distributed WSN-DA methods are the priority of the research study. Communication classification: hierarchy, unstructured, and hybrid; computation classification: hierarchic, averaging, hash function and counting; and DA protocols: Distributed computing DA function concerns and solutions are surveyed in the paper | The receivers address intrusion, fault-tolerance, security and node speed challenges in DA. Process failures were not addressed in the study |
| Aloufi et al. [72] | Topology- and computational intelligence-based terrestrial WSN-DA methods and protocols have discussed in the article. WUSNs, UWSNs, and WBANs are deliberated in the research study | DA methods for terrestrial and underwater WSNs are the recent review of the primary goal. DA protocols unique to other software are excluded from the research study |
| Mono et al. [73], Ara Begum et al. [74] | In WSNs, anomaly-based, encryption-based, privacy, slicing, and other secure DA methods are explored. Secure DA issues are highlighted | The discourse and overview of secure DA methods encompass the article's primary objective. In the paper, DA methods for other challenges are not mentioned |

form calculations and communication, as mentioned in the proposed SSAP and Secure Aggregation of Data Protocol (SADP). Also, MN is transmitting abnormal parameter updates to the AN to perturb the aggregated results (*i.e.*,) the complete update of the parameter. For example, a researcher is transmitting anomalous data to the online store of the application to improve the application's rank [17]. Conversely, MN is a protocol failure that prevents AN from receiving complete parameter updates [5]. Also, MNs collude with other nodes and the aggregator to disclose the node's data privacy [20, 32].

- b. *Untrusted Aggregator*: Like research [5, 33], it firmly performs the algorithms if we suppose the AN is malicious. The MN's false aggregator methods reveal a specific node's data security [52].

Hence, based on ML, developing a security-aware method for protected DA is much necessary. As a result, in this paper, a DA algorithm based on ML is designed in a real-time case where the MNs that perform trust and untrustworthy aggregators are identified earlier.

3.1.2 Aim of Protocol Design

The main goal of the protocol design is to keep the node's updated parameters from being shown to other nodes and the untrusted aggregator during the DA process and to let the AN figure out how many updates there have been.

So, our proposed methods provide the following features:

- a. The protocol is to identify the validity of the node's parameter updates and defend against malicious attacks. As a result, no one interferes with the complete updates of parameters.
- b. The protocol is to aggregate the node's parameter update securely even if MN discontinues or plans to continue with the malicious aggregator and other nodes. A node's parameter updates are not shown to other nodes, and the aggregator computes the full update of the parameter even though MN discards or attacks secretly.

Generally, a node's parameter update is valid only if it transmits accurate updates. Also, the updates of a node's parameters are not revealed to other nodes, even if MN discontinues or plans to work with a malicious aggregator and other nodes; their variable updates are collected securely.

3.1.3 Secret Sharing

Suppose that the data represented as D is classified into ' n ' data parts $D_1, D_2, D_3, \dots, D_n$. The (k, n) secret sharing allows the attackers to attack the data only when the attacker knows k or more data segments. With the help of $(k-1)$ data sets, it is not possible to determine D [5, 53]. A secret sharing method contains a sharing process, for example, $SS.share(D, k, n) \rightarrow \{D_j \in\}_{j \in [1, 2, \dots, n]}$, which has input data D and produces a group of shares D_j . With the help of secret sharing, this work implements a novel SSAP (Sect. 3.1.6).

3.1.4 Computational Diffie-Hellman (CDH) Problem

Assume a group G with a generator g , and $a, b \in \mathbb{Z}$. The CDH problem in G is calculating g^{ab} Without the help of the knowledge of a or b [54, 55]. The researchers [54–56] have shown that the CDH problem is computationally uncompromising in polynomial time. Apart from these researches, this proposed algorithm is CDH protected when revealing node’s information in the proposed algorithm is difficult compared to the CDH problem (Sect. 3.1.7).

3.1.5 Design of Proposed MLBDARP

The basic idea of the proposed protocol is that each node (u_i) transmits the encrypted value updates and SSAP evidence to the AN. The AN authenticates the data and then knows the updates to values. Therefore, the proposed protocol has two essential points, as shown in Fig. 2.

- A. *SSAP*: Nodes first produce the corresponding SSAP indication of the updates to the ML parameter and then send the proof along with the parameter updates to the AN. On receiving node data, the AN authenticates it with the help of SSAP proofs. Only the AN accepts it if they find it to be correct. Otherwise, it rejects.
- B. *SADP*: Nodes locally produce private keys without any complex encoding algorithm and apply keys to encrypt updates to variables. The AN receives the encoded data and estimates the complete updates.

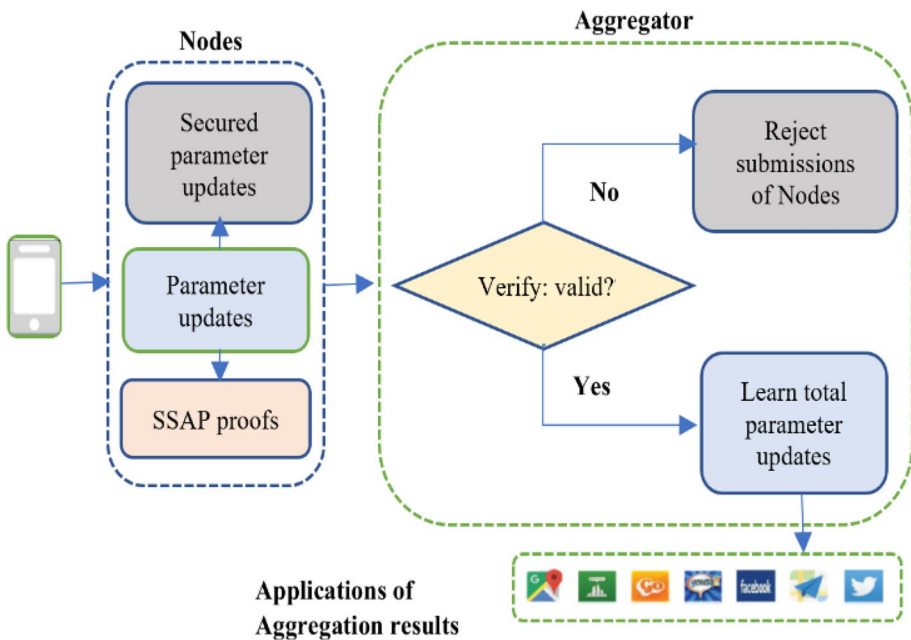


Fig. 2 ML-based DA model for SSAP and SAD

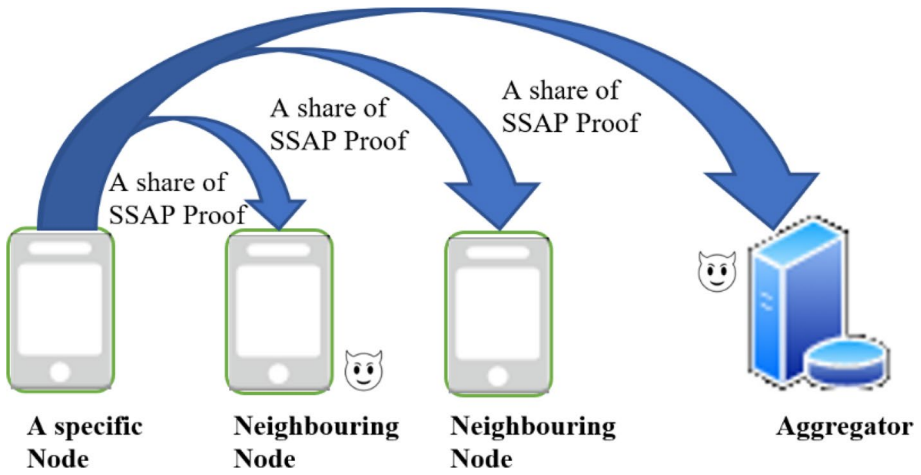


Fig. 3 Proof: A specific node transmits proofs to; neighbouring nodes and AN, even though MN failure

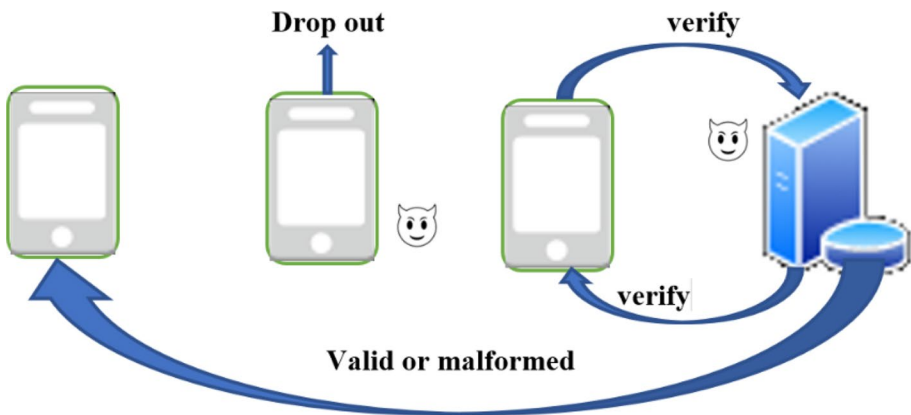


Fig. 4 Authentication: The aggregator authenticates the validity of the parameter updates with the help of proofs

3.1.6 Secret Shared Authentication Protocol

This work’s primary task is to authenticate that a node u_i does not send malformed parameter updates. To achieve this purpose, this research proposes an SSAP. The recommended protocol is stimulated by the research work [57], but for IoT-WSN, this article considers the different scenarios, as shown in Figs. 3 and 4. This paper detects that only an AN ‘A’ exists, and the “MN” is a failure.

The nodes and the DA is A have a mathematics check $Valid(.)$ SSAP can show if $Valid(x_i)=1$ without leaking u_i ’s variable updates x_i to the ANS ‘A’ and other nodes. The participation of the mathematics check $Valid(.)$ over the field ‘F’ are $x = \{x(1), x(2), \dots \dots , x(L)\} \in F^L$, and has one output. Each vertex in arithmetic check $Valid(.)$ is either a gate (or) input/output vertex. The input vertexes are denoted by $\{x(1), x(2), \dots \dots , x(L)\}$ (or) constant in ‘F’. Every gate vertex consists of 2 inputs and 1

output, denoted by $+/ \times$. Let us consider that N multiplication gates in mathematics check assure condition $2N \ll |F|$. Mathematics check $Valid(\cdot)$ conducts a mapping: $F^L \rightarrow F$.

This research proposes an SSAP algorithm which has the following steps of execution:

- Step 1* Initialize the protocol.
- Step 2* Node u_i divides updates of the parameter x_i into S shares with the help of the improved secret-sharing method.
- Step 3* SSAP, SS share $(x_i, n, u) \rightarrow \{[x_i]_j \in \mathbb{F}\}_{j \in [1, 2, \dots, s]}$
- Step 4* Input private data x_i , threshold ' n ' and set ' u '.
- Step 5* $(s - 1)$ nodes and aggregator $A(n \leq s)$ produce a set of shares $[x_i]_j$.
- Step 6* u_i Evaluates arithmetic check $Valid(\cdot)$ on input x_i .
- Step 7* The input and output of the t^{th} multiplication gate are IN_t and OUT_t .
- Step 8* ($t \in (1, 2, \dots, N)$). u_i defines polynomials $p_1 = IN_t$ and $p_1 = OUT_t$ with $N-1$ degree.
- Step 9* The Polynomial $p_3(t) = p_1(t)p_2(t)$ with $2N-2$ degrees.
- Step 10* The u_i generates the shares $[p_3(t)]_j (j = \{1, 2, \dots, s\})$.
- Step 11* Shares $\{[a_1]_j, [a_2]_j, [a_3]_j\}$ that meets the limits $a_1 a_2 = a_3$ using the improved secret-sharing method.
- Step 12* u_i transmits shares $[p_3(t)]_j, [x_i]_j$, and $\{[a_1]_j, [a_2]_j, [a_3]_j\}$ to the AN 'A' and neighbour nodes $u_{i-1}, u_{i+1}, u_{i+2}, \dots, u_{i+s-2}$.
- Step 3* If u_i is the MN, then it sends abnormal data $[p_3(t)]_j, \{[\hat{a}_1]_j, [\hat{a}_2]_j, [\hat{a}_3]_j\}$.
- Step 14* It is satisfied $\hat{a}_1 \hat{a}_2 \neq \hat{a}_3$ and $[x_i]_j$.
- Step 15* AN 'A' identifies abnormal data and discards it.

3.1.7 Secure Aggregation of Data Protocol (SSDP)

We introduce the SSDP algorithm, whose main aim is that each node encodes the data of the parameter updates locally, transmits the encoded data to neighbouring nodes, and calculates the missing value when the AN generates a list of discontinued nodes. The concept is encouraged by the research [54], but here we propose a new thing where MN is terminated, and therefore the missing values have to be calculated. The proposed SADP algorithm has the following steps of execution:

- Step 1* Nodes $u_1, u_2, \dots, u_i, \dots, u_s$ Join the DA based on ML.
- Step 2* Each node $u_i (i = 1, 2, 3, \dots, s)$ has private updates of parameter x_i .
- Step 3* Two prime numbers are assumed, represented as (p, q) and have the same length.
- Step 4* ' q ' divides $p-1$.
- Step 5* A q -order multiplication group $g = \langle g \rangle$, where, g is generator and represented as $g = g_1^p \text{ mod } p^2, g_1 = h^{\frac{(p-1)}{q}} \text{ mod } p (g_1 \neq 1)$, and $h \in \mathbb{Z}_p$ denotes a random value.
- Step 6* Execution of four rounds.
- Step 7* In round 0, each node u_i arbitrarily chooses a private number $r_i \in \mathbb{Z}_g$.
- Step 8* Each node shares a number $g^{r_i} \in G$ with nodes u_{i-1} and u_{i+1} .
- Step 9* The node u_s shares g^{r_s} with nodes u_1 and u_{s-1} .
- Step 10* Node u_1 shares with node u_s, u_2 .

- Step 11* Node u_i calculates private key $g^{r_i r_{i+1} - r_i r_{i-1}} \text{MOD } p^2$ and uses the private key for encryption value updates x_i with $\hat{x}_i = (1 + x_i p) g^{r_i r_{i+1} - r_i r_{i-1}} \text{MOD } p^2$.
- Step 12* In round 1, u_i send \hat{x}_i to AN 'A'.
- Step 13* In round 2, as the AN 'A' receives \hat{x}_i , then checks the MN who failed.
- Step 14* In such case, 'A' generate data about terminated nodes.
- Step 15* In round 3, the remaining nodes calculate the absent number and transmit it to 'A'.
- Step 16* Finally, 'A' calculates the updates of parameters without decrypting the data.

The above procedure of SADP has been summarized in Fig. 5.

3.1.8 Security Analysis

In this section, this article theoretically analyses the security of SSAP and SADP. SSAP prevents data confidentiality against abnormal malicious attacks. In SSAP, to fool the AN into accepting the wrong updates of parameter, an MN is transmitting the malformed $\hat{p}_3(t)$, ill-formed $\{[\hat{a}_1]_j, [\hat{a}_2]_j, [\hat{a}_3]_j\}$ and even wrong \hat{x}_i . The probability of noticing such malicious behaviour is at least $(1 - \frac{2N-2}{|F|})$. We can say that the most considerable probability is $\frac{2N-2}{|F|}$. So, the probability is decreased by increasing $|F|$, for example, by setting $|F| = 2^{265}$. Also, according to property (s-1), MN and malicious aggregator methods with each other cannot get accurate variable updates. In short, SSAP authenticates nodes without getting the information about the data. The AN is authenticated nodes in less than (s-n) MN failures (Sect. 3.1.3).

SADP provides CDH secured from failure nodes and colluding with other nodes. When an MN wants to disclose a specific node, for example, u_i 's updates x_i then it has to calculate $g^{(r_{i+1} - r_{i-1})r_i}$. Moreover, the MN gets the access to $g^{r_{i+1}}$, g^{r_i} , and $g^{r_{i-1}}$ with the help

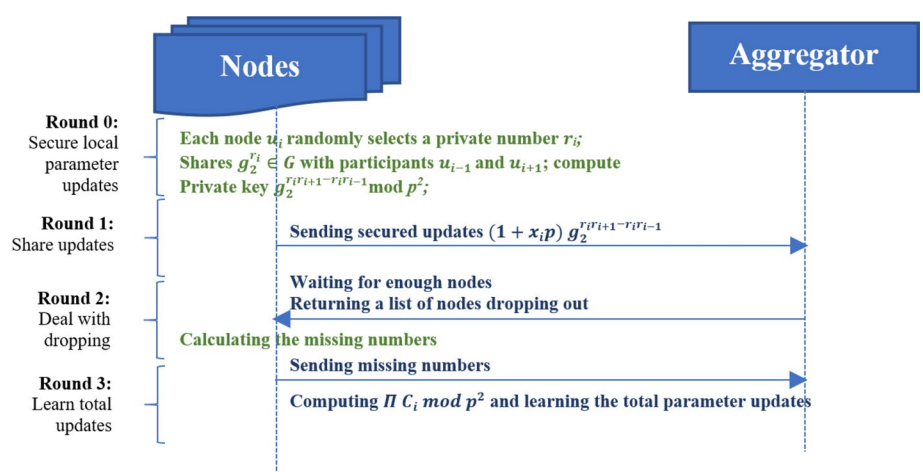


Fig. 5 Outline of SADP: Nodes may cause failure in each round, and SAD protects against the collusion attacks

of colluding with others and failure, but it cannot compute $g^{(r_{i+1}-r_{i-1})r_i}$ without the help of $r_{i+1} - r_{i-1}$ and r_i , since it is a CDH problem to calculate $g^{(r_{i+1}-r_{i-1})r_i}$ without having the information of $r_{i+1} - r_{i-1}$ and r_i (Sect. 3.1.4).

3.1.9 Complexity Analysis

The communication and computation complexity in the proposed protocol is $O(sN \log N)$ and $O(\max\{sN, s|p|\})$ at most, where ‘ s ’ is many nodes. The total complexity of the proposed protocol is specified in Table 2.

Every node has to complete $O(N \log N)$ non-cryptographic computations for $p_3(x)$, where ‘ N ’ represents good circuit size. As every node has to share the numbers, which consist of the output of wires in the valid circuit, the communication complexity is $O(N)$. The AN computes the validity of the circuit, which considers the ideal value $O(N \log N)$ of each node. In other words, this work exposes that the communication complexity is $O(1)$. The overall complexity of the computation and communication in SSAP protocol is as $O(sN \log N)$ and $O(sN)$, where ‘ s ’ represents the number of nodes.

In SAD protocol, in Round 0, each node computes keys and parameter updates, acquiring $O(1)$ as computation complexity. However, nodes exchange number $g_2^{r_i}$, and the communication overhead corresponding to each node is $O(|p|)$. In Round 1, $O(|p|)$ is the communication overhead because each node transmits the secured updates to the AN. In Round 2, communication complexity is $O(s|p|)$ at most because the AN returns a list of nodes that drop out during communication. In Round 3, $O(|p|)$ is the communication complexity of each node since nodes transmit the AN, the information about missing values. To calculate total updates, the AN performs increases, and hence the computation complexity is represented as $O(s)$. In conclusion, we can say that the computation and communication complexity in the SAD protocol is $O(s)$ and $O(s|p|)$, respectively.

3.2 Proposed Routing Protocol

The proposed protocol uses an ML method to perform the next-hop selection. When the connection is linked between the nodes, the buffer of one node has the data to be sent; the decision to send the message to another node is called next-hop selection. The data will only be transmitted from the transmitter node to the neighbouring receiver node if it has a high possibility of being transmitted to the sink either directly or indirectly. Sending the data too often may lead to high packet loss and RO. Less frequent data transmission results in fewer delivered messages. A successful PDR is determined by the numerous aspects representing the past and the node’s ability to send data effectively. The probability of a successful PDR at the next hop is considered by a model based on ML and trained, which

Table 2 Computation and communication complexity in proposed SSAP and SADP

| DA using ML | | Computation | Communication |
|-------------|---------|----------------|---------------|
| SSAP | | $O(sN \log N)$ | $O(sN)$ |
| SADP | Round 0 | $O(s)$ | $O(s p)$ |
| | Round 1 | None | $O(s p)$ |
| | Round 2 | $O(s)$ | $O(s p)$ |
| | Round 3 | $O(s)$ | $O(s p)$ |

Table 3 Input features for the proposed algorithm

| Features | Symbol | Description |
|----------------------------|----------|---|
| Probability | x_1 | Probability for the selection of the next hop in the proposed routing algorithm (Sect. 3.2.1 – (3)) |
| Buffer size | x_2 | The size remained in a buffer for saving more data packets (Sect. 3.2.1 – (3)) |
| Effective PDR | x_3 | Number of effective data transmissions from the beginning of the simulation to the present instance among any pair of nodes |
| Successful PDR | x_4 | The ratio of practical data moved to total transmissions initiated among any two nodes |
| Start node Speed | x_5 | Node mobility of transmitter node |
| End node speed | x_6 | Node mobility of receiver node |
| Travel time of data source | x_7 | Distance of the position of interaction among any two nodes from the beginning point of data |
| Distance to data limit | x_8 | Distance of position of interaction to the closing terminus of the data |
| Data process Time | x_9 | Time from the formation of the data to the present time |
| Initial node Energy | x_{10} | Transmitter node's energy |
| End node Energy | x_{11} | Destination node's energy |
| Current hop count | x_{12} | Number of hops, data has travelled before reaching the present transmitter node |

has the following features: speed, EC, PDR, distance from the data source, distance to a data sink, data processing time, and hop count. The time from data generation to the current time is signified by data processing time. The data is sent from the transmitter node if $P_m > k \times P_r$, where P_m is the final delivery probability, also known as the ML probability considered with ML models, P_r represents the probability of the transmitter node delivering the data to the sink, $k \in [0, 1]$ is a normalization factor.

3.2.1 Computation of the ML Probability P_m

For the calculation of P_m and assessing the performance analysis of the proposed protocol, we used two models based on the concept of ML, one NN and the other DT.

1. *Neural Network Model*: This work considered an IoT-WSN model, which is based on NN with many unseen layers where $(x_1, x_2, x_3, \dots, x_{12})$ are input features, as discussed in Table 3, creating an input layer. p_1 and p_2 are outputs that denote the probability of successful and unsuccessful delivery. This work says that p_1 is the ML probability P_m , which denotes the successful PDR probability of given inputs (*i.e.*,) $(x_1, x_2, x_3, \dots, x_{12})$. This value in IoT WSN functions as a linear clustering of the node's value in the preceding layer at each node. Value at node h_i as Eq. (1)

$$h_i = F\left(\sum_{j=1}^n x_j w_{ji}\right) \tag{1}$$

where, x_j denotes j th a node of the preceding layer, F represents the activation function, and ‘ w ’ denotes the weight matrix. For example, the value of the node h_1 , which is the linear combination of input parameters $(x_1, x_2, x_3, \dots, x_{12})$ is crossed with F , Eq. (2)

$$h_1 = F(w_{11}x_1 + w_{21}x_2 + w_{31}x_3 + \dots + w_{121}x_{12}) \tag{2}$$

Hence, during the forwarding feed operation, as a move from the input to the output layer, a value is measured at all nodes; therefore, the output layer values are computed. For the execution of this step, the weight matrix w_{ij} for each linear clustering is computed. It is done with the help of training, and computation weights make NN.

Training: Here, the use of the Backpropagation algorithm for training data for NN [58]. The training data has the input values $(x_1, x_2, x_3, \dots, x_{12})$ for selecting the next hop, the output rules whether the data reach the sink (or) not. If the output is correct, then $p_1 = 1$ and $p_2 = 0$, else, $p_1 = 0$ and $p_2 = 1$. With the help of training, a sample NN formation considers place using random values. To decide whether the PDR is successful, the NN is learned through the training set. For the resultant value of P_m , the ML model is trained based on data stored in the training situation. The data is in the training phase, and data access is an example. Following actions are presumed in training data for each training example.

- (i) At each layer, the input is distributed to produce the value of the activation functions. Then call it $NN_{Prediction}$. Suppose NN_{actual} be actual values from the training dataset.
- (ii) Error during the training is considered with the sum of the squared difference between desired and predicted value represented as given in the least Mean Squares (LMS) algorithm [59] (Eq. (3))

$$J(w) = \frac{1}{2} \sum (NN_{actual} - NN_{Prediction})^2 \tag{3}$$

- (iii) To reduce the error due to training, the weights are corrected $J(w)\delta(w)$, Eq. (4):

$$\delta(w) = -n\left(\frac{dJ}{dw}\right) \tag{4}$$

Where ‘ n ’ represents the learning rate which denotes the virtual transformation in weights because of error during training.

- (i)) The updating the value of w , Eq. (5):

$$w(new) = w(old) + \delta(w) \tag{5}$$

The steps mentioned above are for all the training test cases, which results in a professional NN with learned weights that enable the best prediction with minimum error.

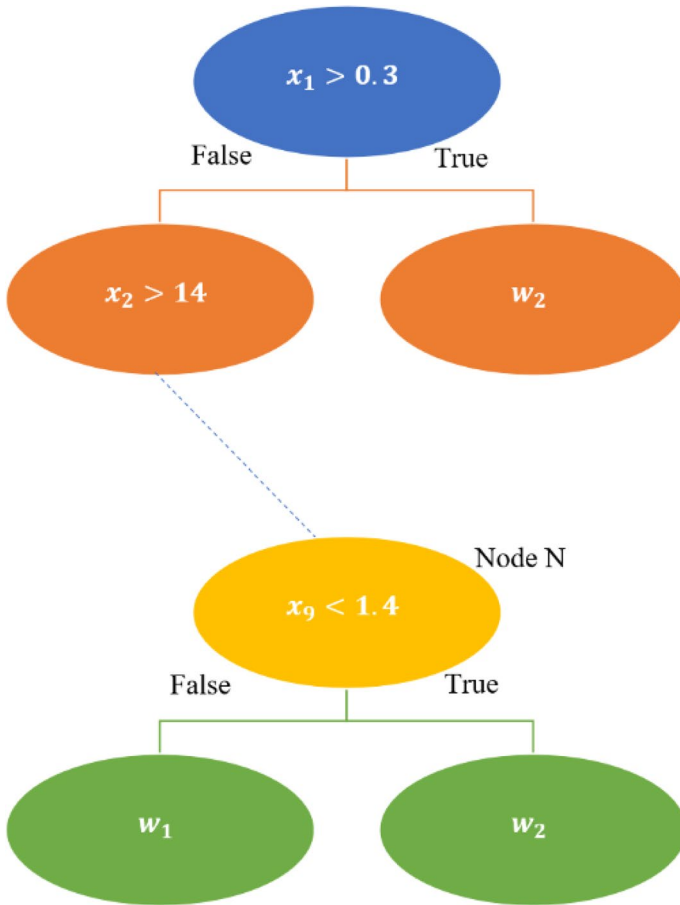


Fig. 6 Decision Tree

Calculation of P_m (Feed Forward Process (FFP)): For calculation of P_m , the trained NN is used based on the input parameter $(x_1, x_2, x_3, \dots, x_{12})$ found in real-time for the selection of the next hop. The value is computed at each node as we Move from the input to the output layer. The linear clustering of nodes of a preceding layer with the \mathbf{F} is applied to this clustering. With the help of Eq. (2), the FFP is performed where w is calculated from the training. As we proceed from the input to the output layer by crossing all unseen layers, the value of the neuron is calculated using Eq. (2), which is also used for calculating node value for the subsequent layers. The closing output is p_1 and p_2 , where p_1 represents P_m .

2. *Decision Tree Model:* Fig. 6 shows a DT model, where w_1 and w_2 denotes output represents the successful PDR and unsuccessful deliveries respectively, $(x_1, x_2, x_3, \dots, x_{12})$ are input, and nodes denote the decision based on input. For example, starting at the tree's root for a group of inputs, we proceed to leaf nodes. For example, if $x_1 = 0.8$, then the prediction of input belongs to a class w_2 . Let us suppose that we reach node N with the value $x_9 = 1.8$, then the prediction class becomes w_1 and P_m the node's probability of falling into w_1 from the training set when node N has moved.

Formation of DT: Formation of DT is recursively taking place with the help of training data as given below:

BuildTree(S): The most suitable characteristic and the equivalent value for forming the root of the DT is initiated first. x_1 is the first attribute selected in the first recursive call and $x_1 > 0.3$ is the decision. The most common method for calculating the attribute for splitting the data set is entropy impurity [59], Eq. (6)

$$i(N) = - \sum P(w_j) \log P(w_j) \tag{6}$$

where $P(w_j)$ represents the part of outlines at a specific node which lies in the w_j Group. For the splitting function, the query which decreases the impurity is selected [Eq. (7)]

$$\delta(i) = i(N) - P(x)i(N_x) - (1 - P(x))i(N_y) \tag{7}$$

where, N_x is left nodes and N_y denotes the correct nodes and $i(N_x)$ and $i(N_y)$ denotes their respective impurities. Let us suppose, if $x_1, x_2, x_3, \dots, x_{12}$ are features, then the features which have a maximum value of $\delta(i)$ is considered the root of S.

1. Based on the features estimated above in Step 1, dataset S split into S_x and S_y in which S_x resembles the left subtree of the root of S and S_y resembles the correct subtree.
2. The functions $BuildTree(S_x)$ and (S_y) are recursively for the creation of the root.

This method mentioned above is continuous until the maximum value of $\delta(i)$ falls below a predetermined threshold value. In this work, the C4.5 service of DT [60] and gain ratio impurity equations are used for the calculation of $\delta(i)$ where a modification in impurity is reduced by dividing it with entropy, the enhanced J48 DT by the WEKA tool [60] is used.

Computation of P_m : Because the DT is developed entirely based on the input value, the conditions for DT are concerned until a leaf node is moved. Class calculated at the leaf node is used for the determination of P_m with the help of the probability distribution from the training set. For example, if a class w_1 is predicted, P_m is calculated by the values in w_1 set from the predecessor node divided by the number of iterations of the predecessor noted. Input features $(x_1, x_2, x_3, \dots, x_{12})$ are used by the proposed protocol for the calculation of P_m . The selection of the next hop in the DT and NN.

3. *Computation of Normalization Factor Represented as K and Probability:* Though the router is not used directly to select the next hop, the PDR probabilities are updated and used as variables for the proposed protocol based on ML. The protocol is implemented by updating probabilities, so nodes communicate regularly with high PDR success rates. Equation (8)

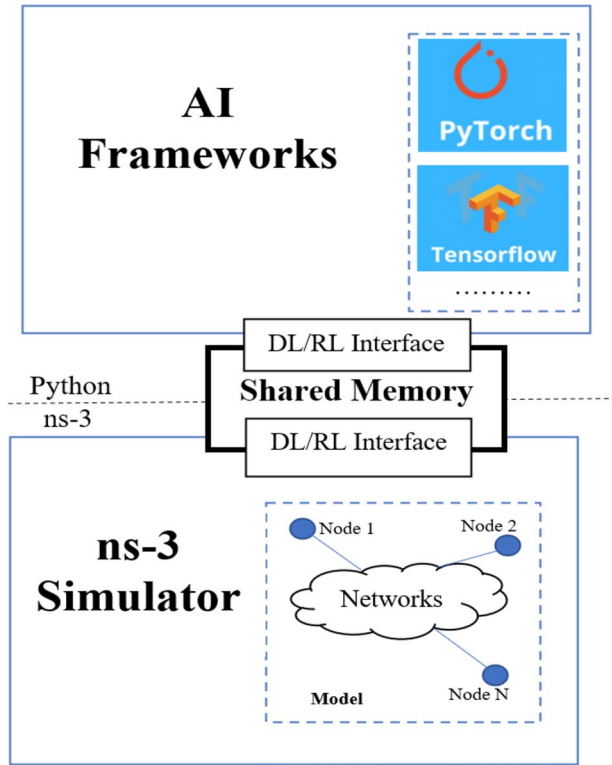
$$P(x, y) = P(x, y)_{old} + (1 - P(x, y)_{old}) \times P_{init} \tag{8}$$

If specific nodes are not connecting, their PDR probabilities must also become old, Eq. (9)

$$P(x, y) = P(x, y)_{old} \times (\gamma)^k \tag{9}$$

where γ represents the factor for aging and k denotes the period since the previous aging occurred.

Fig. 7 Proposed framework



The protocol also observes the tentative behaviour between the nodes. For example, let us consider that there are three nodes (n_1, n_2, n_3) , (n_1, n_2) frequently connect, (n_2, n_3) . In this situation, a message sent to n_3 , if forwarded from n_1 , has a high PDR probability, Eq. (10)

$$a(b_c)P_{(n_1, n_3)} = a + b + P_{(n_1, n_3)_{old}} + \left(1 - P_{(n_1, n_3)_{old}}\right) \times P_{(n_1, n_2)} \times P_{(n_2, n_3)} \times \beta \quad (10)$$

This PDR rate is an integral part of the proposed ML algorithm because it proposes the best method to send node and link logs, which is necessary to select the next hop.

The value of the K lies in the range of $(0, 1)$. The closing value of the normalization factor is decided by multiple values of K in the range of $(0, 1)$ and by noticing the number of data messages delivered. The final PDR probability vs. normalization factor K value is a Gaussian curve, and the value of K decided by the point which denotes the highest PDR probability. This value is assumed into consideration in this proposed work.

Buffer size denotes the level of the receiver node in terms of storage for forwarding the data packet (Eq. (11)).

$$Buffer\ Occupancy = buffer\ Size_{available} - message\ Size_{to\ Be\ Forwarded} \quad (11)$$

Table 4 Parameters of the Simulation

| Test variables | Description | Values | Description |
|-------------------------|---|-----------------------|-----------------------------------|
| Network simulator | NS-3, Version 3.33 | PHY /MAC | IEEE 802.11 |
| Network Area | 2500×2500 m | Propagation | Two-ray ground |
| Connection | UDP/CBR | Mobility/channel type | Random way point/wireless channel |
| Mobility (<i>ms</i>) | 2, 6, 10, 14, 18, 22, 26, 30, 34, 38 | Data (<i>kbps</i>) | 2, 4, 8, 16, 32, 64, 128, 256 |
| Source and destination | Random | Antenna | Omnidirectional |
| Packet size | 2048 bytes | Time | 1000 S |
| Protocols | MLDAR, DDAR, WDARS | Language | C++ and Python |
| Scenario (mobile nodes) | 25, 50, 75, 100, 125, 150, 175, 200, 225, 250 | MN | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 |
| Node energy | 20 dBm | NS3-AI | OpenAI and NS-3 |
| Numerology | 1 | Operating Frequency | 100 MHz |
| Bandwidth | 100 MHz | Ambient noise level | -73 dBm |
| Mobility module | Constant speed | Node energy | 100 J (%) |
| Algorithm | Random Robin | Threshold energy | 40 J (%) |

3.2.2 ML Process in Proposed MLBDARP

The proposed protocol has solved the following hop selection problem with the help of PDR probability when selecting the next hop. It depends on variables like probability, buffer size, history of the communication, and node speed. The ML process develops a model with input features at execution time and output PDR rate. The ML model training is based on input and output values in real-time scenarios. The process of ML is effective for assessing values for the selection of the next hop, which results in the prediction of PDR rates is P_m .

3.2.2.1 Proposed Algorithm The proposed algorithm for the selection of the next hop is explained below:

- (a) *Training*: The data is generated for the next hop by executing the training scenario. The sender node sends the message to the receiver node, and the entry in the data determines if the PDR was successful. To confirm the successful PDR in all cases, use the standard routing protocol [46].

After the DA, learning models based on networks and decisions are formed. Whenever the data is sent, the next-hop selection choice is made, and the following computations are done:

- (i) Compute the probability P_r .
- (ii) Calculate the ML probability, P_m which is based on a trained ML model.
- (iii) Send data from the transmitter to the desired node if $P_m > K \times P_r$.

4 Protocol Implementation and Simulation Results

The implementation of the proposed MLBDARP is discussed in Sect. 4.1, and the corresponding simulation results and their comparison with the existing research work [34, 61] have been explained in Sect. 4.2.

Table 5 Summary of the percent increase in NT of MBLDARP vs. other Protocols

| Parameters | DDARP (%) | WDARSP (%) |
|---------------|-----------|------------|
| Nodes | 43.17 | 10.02 |
| Data rate | 51.72 | 22.01 |
| Node mobility | 25.31 | 19.51 |

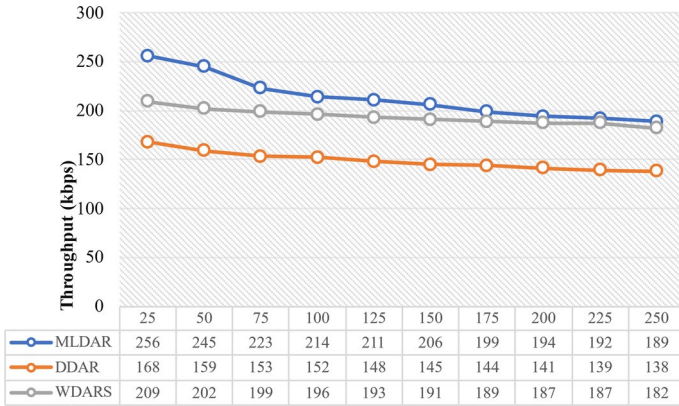


Fig. 8 No. of nodes versus NT

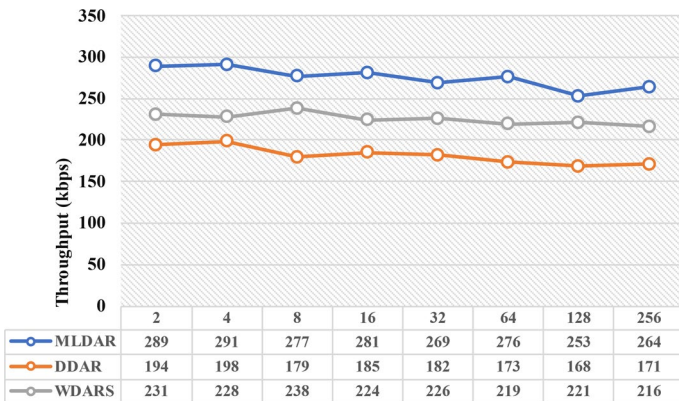


Fig. 9 Data rate versus NT

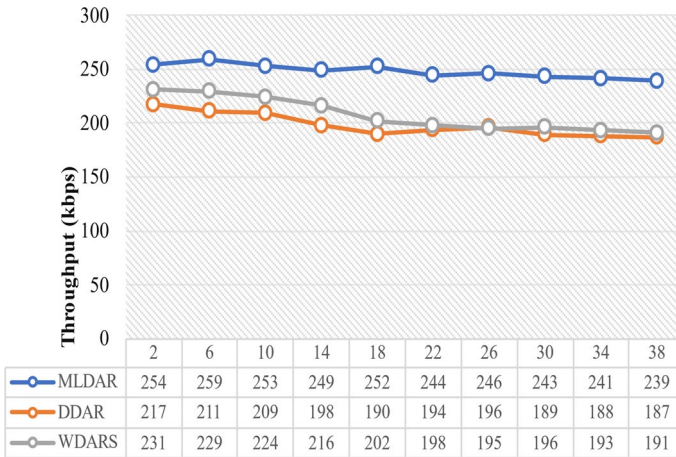


Fig. 10 Node mobility versus NT

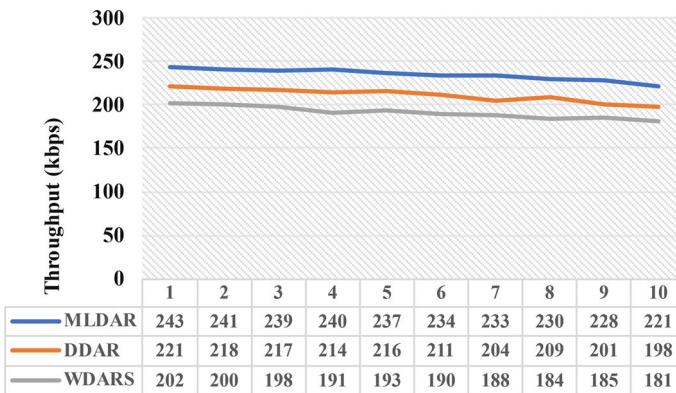


Fig. 11 Malicious nodes versus NT

4.1 Proposed Protocol Implementation

The proposed implementation model for running Artificial Intelligence (AI) algorithms in Network Simulator-3 (NS-3) is represented in Fig. 7. Different processes are involved in running NS-3 and AI. There are mainly two cases in which data transmission is required: transmitting data to train and test the AI in NS-3. For setting up the network and topology, NS-3 is used, which creates information for training in AI.

The NS-3 provides settings for validating AI algorithms across all network layers by generating simulated scenarios. It appears to include a function in NS-3 for analyzing a well-trained model. Because the NS-3 codes are all open-source, readily available, and documented, the analysis might occur in any inner layers (or) modules. In addition, the NS-3 may act as a data generator. Consider a mobile network where users join or disconnect

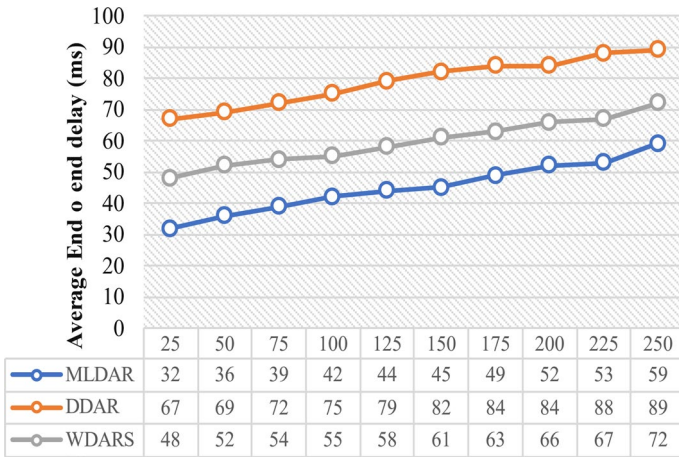


Fig. 12 Nodes versus EED

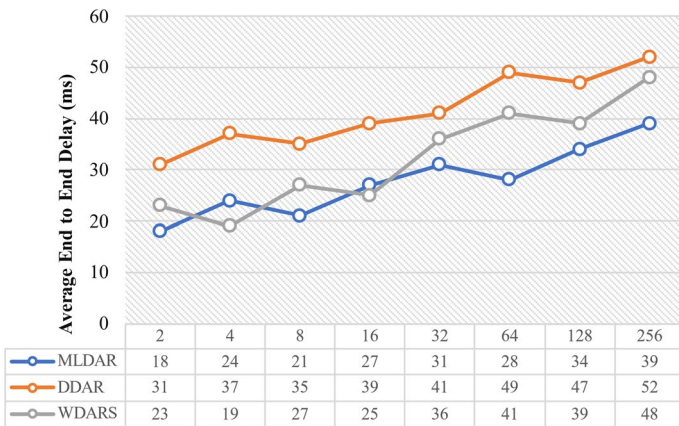


Fig. 13 Data rate versus EED

from the base station (or) move from one side to the other. In general, this work is frequently about the user’s node position and speed and classifying its channel state and user practices. A scheduling model might be trained using this data.

NS-3 efficiently handles the learning model’s simulation creation and lower-layer data collection. As a result, NS-3 is regarded as a data generation and testing tool for different needs. This module supports the integration of many AI systems that use Python scripts. We may retrieve data from shared memory using the interfaces on the Python side, then continue training the model or return the output for testing. It does not affect the AI’s core functioning, data processing, or set-point technique. As a result, rerunning the existing

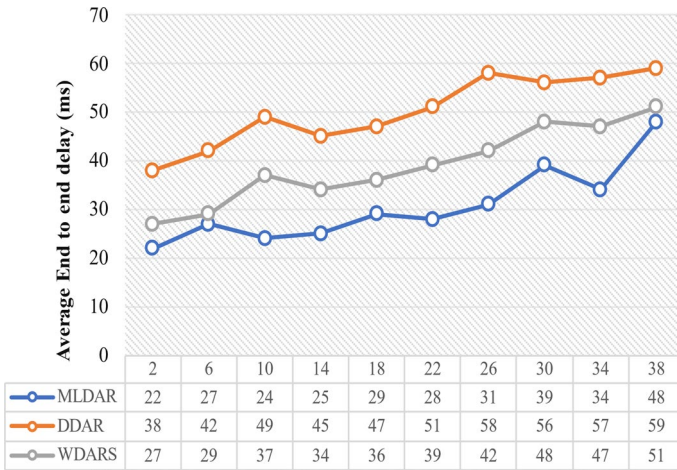


Fig. 14 Node mobility versus EED

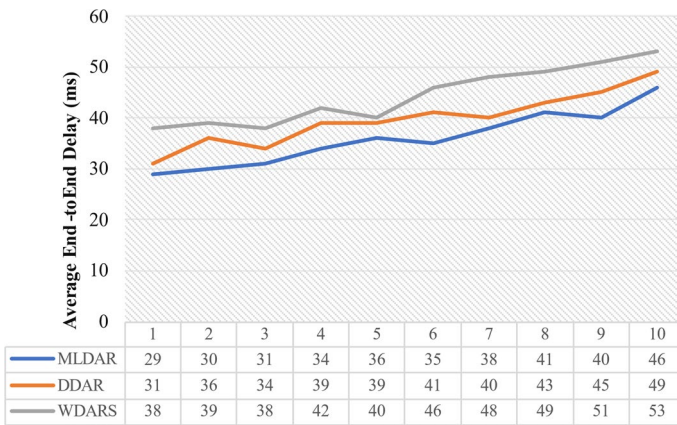


Fig. 15 Malicious nodes versus EED

Table 6 Summary of the percent decrease in EED of MBLDARP vs. other protocols

| Parameter | DDARP (%) | WDARSP (%) |
|---------------|-----------|------------|
| Nodes | 42.83 | 24.32 |
| Data rate | 32.93 | 13.95 |
| Node mobility | 38.84 | 21.28 |

methods in NS-3 is easy. The shared memory pool integrates the NS-3 and AI via the NS-3-Ai module. Both sides access and control the memory, which NS-3 primarily controls. The NS-3 module may shift the data sent through the two sections from training and testing data to control signals. As all types of communications are broadcast through the NS-3-AI module, it represents distinct memories for complex contexts in this method.

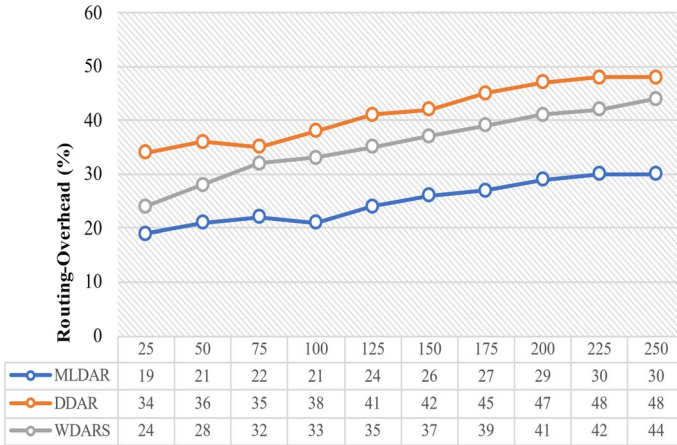


Fig. 16 Nodes versus RO

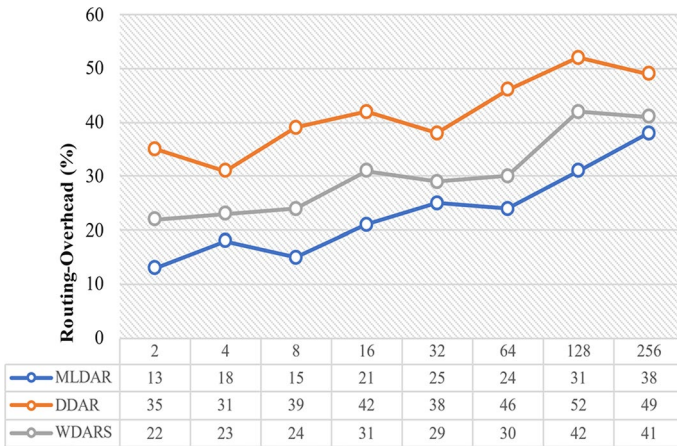


Fig. 17 Data rate versus RO

- a. *Hardware Configurations:* Intel (R) Core (TM) i7-8550U CPU, 1.80 GHz, 1992 MHz, 4 Core(s), 8 Logical Processor(s); RAM: 16 GB; HDD: 2 TB; Network: Gigabit Ethernet and Wi-Fi
- b. *Software Configurations:* OS: Ubuntu 20.04.2.0 LTS; Compiler: GCC 9.3; Parallel Computing: OpenMPI 4.0.3; Environments: PyTorch, Keras and TensorFlow dependences
- c. *Hyperparameter Configurations:* Learning rate method: exponential decay; mini-batch size: 64; training epochs: 100.

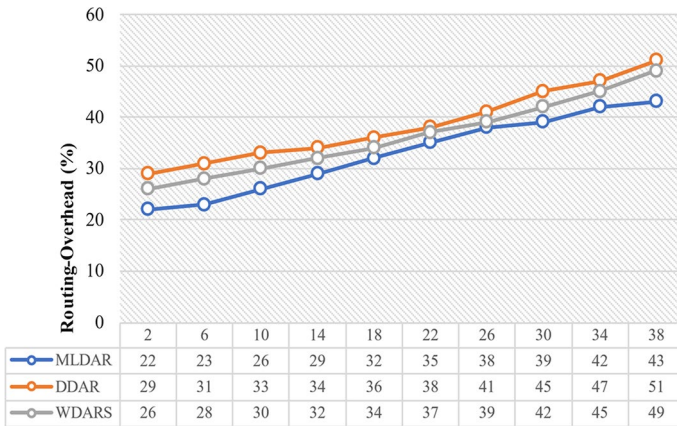


Fig. 18 Node mobility versus RO

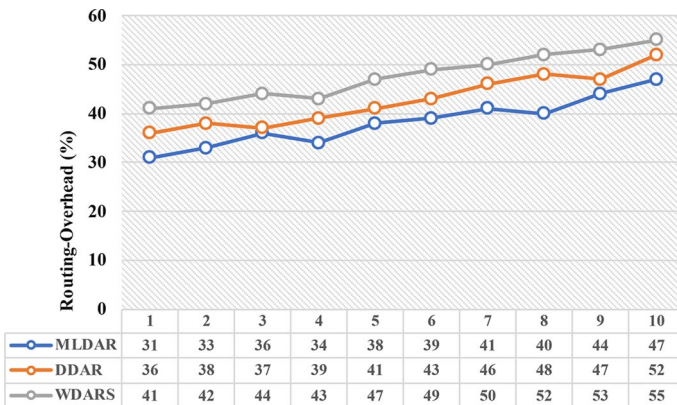


Fig. 19 Malicious nodes versus RO

Table 7 Summary of the percent decrease in RO of MBLDARP vs. other protocols

| Parameter | DDARP (%) | WDARSP (%) |
|---------------|-----------|------------|
| Nodes | 39.85 | 29.85 |
| Data rate | 44.27 | 23.55 |
| Node mobility | 14.54 | 9.11 |

Table 8 Summary of the percent increase in PDR of MBLDARP vs. other protocols

| Parameter | DDARP (%) | WDARSP (%) |
|---------------|-----------|------------|
| Nodes | 5.30 | 7.61 |
| Data rate | 10.17 | 4.62 |
| Node mobility | 4.27 | 2.26 |

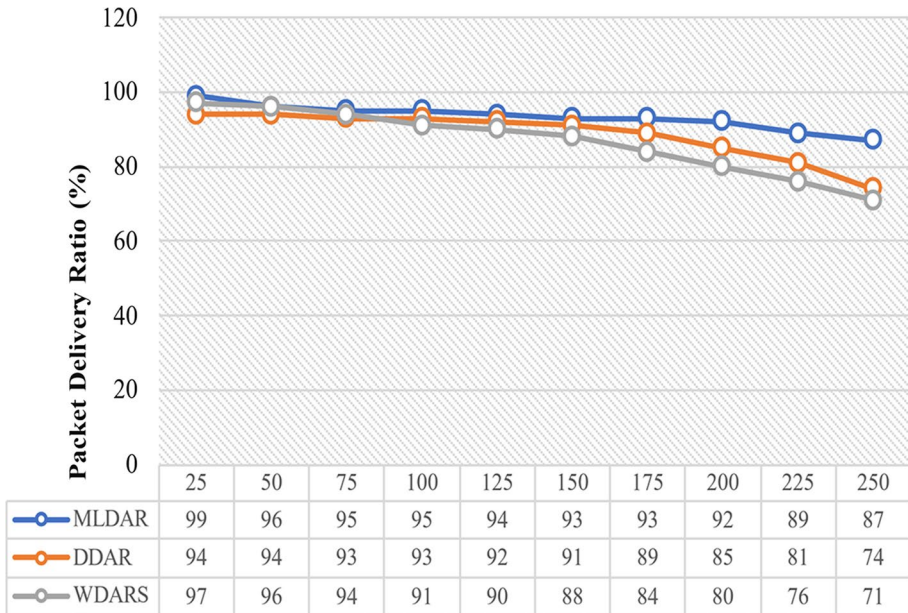


Fig. 20 Nodes versus PDR

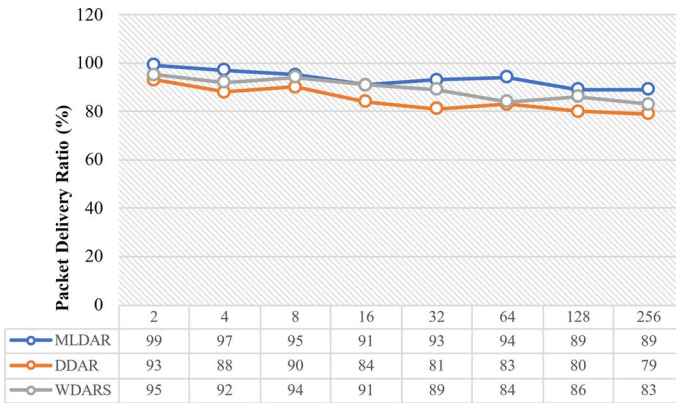


Fig. 21 Data rate versus PDR

5 Simulation Results

The benchmark tests the effectiveness of the recommended protocol MBLDARP, in which secure DA and a routing method based on ML are used, as discussed in Sect. 3, against DDAR [34] and WDARS [61] using NS-3. For that purpose, 25, 50, 75, 100, 125, 150, 175, 200, 225, and 250 SN of standardized features with variations in MN from 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 are installed in the IoT-WSN. While routing, MN fails to receive data packets. The NT, RO, EED, PDR, and EC QoS metrics of the MBLDARP simulation study are assessed by comparing them to those of other protocols [34, 61]. Table 4 summarises

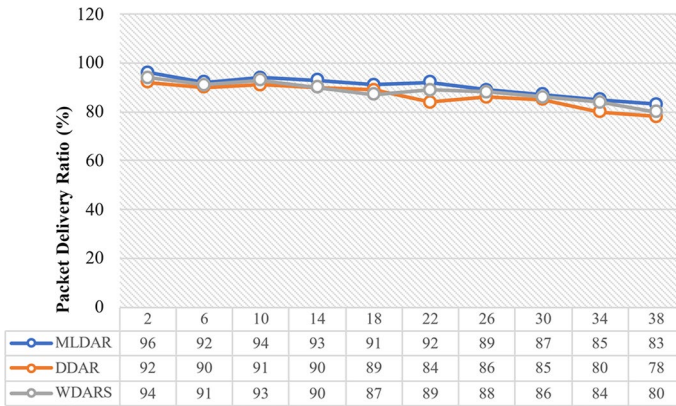


Fig. 22 Node mobility versus PDR

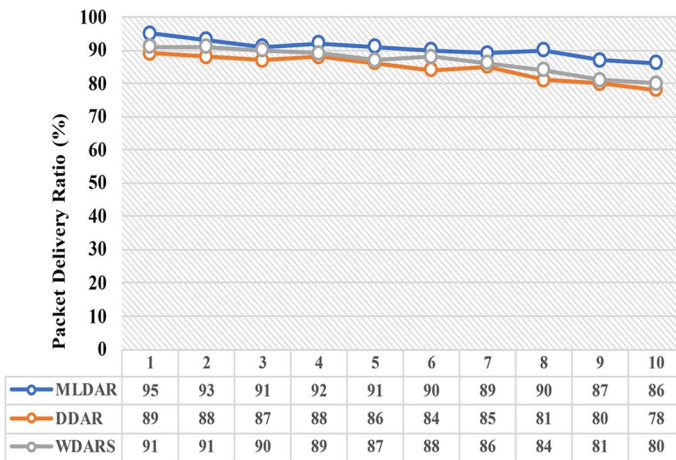


Fig. 23 Malicious nodes versus PDR

Table 9 Summary of the percent decrease in EC of MBLDARP over other protocols

| Parameter | DDARP (%) | WDARSP (%) |
|---------------|-----------|------------|
| Nodes | 10.96 | 9.27 |
| Data rate | 15.48 | 8.01 |
| Node mobility | 3.19 | 1.44 |

the simulation variables. This paper has added the idea of the proposed protocol, which is discussed in Sect. 3, and split NS3-GYM, a toolkit, into two communicating processes (*i.e.*) NS-3 and OpenAIGym (Python). This section compares the MBLDARP, DDARP, and WDARSP recommended protocols. The test considered data rates of 2, 4, 8, 16, 32,

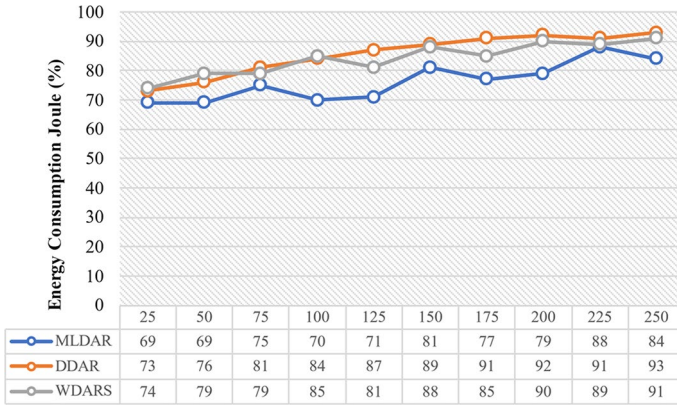


Fig. 24 Nodes versus EC

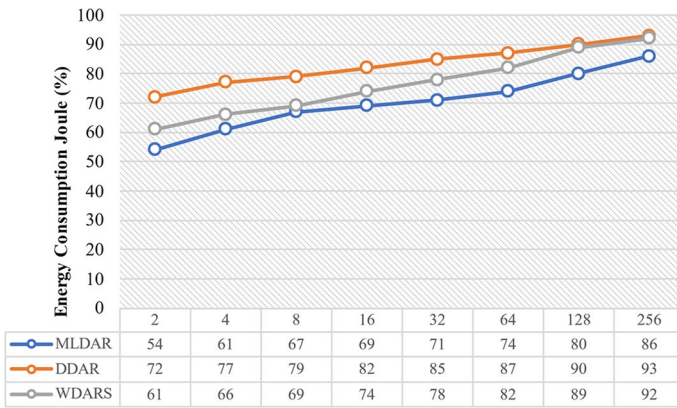


Fig. 25 Data Rate versus EC

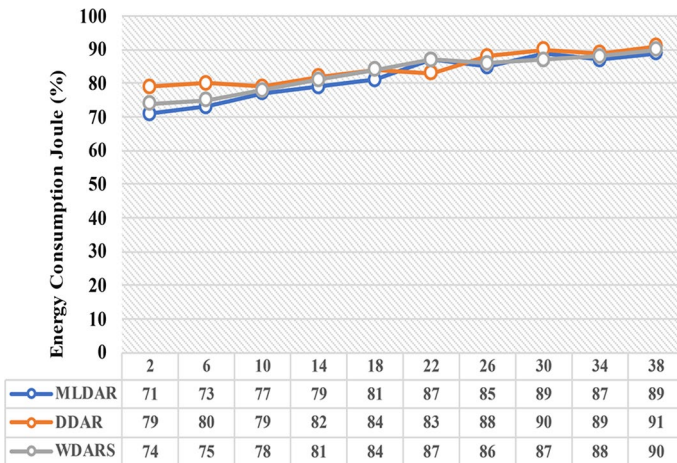


Fig. 26 Node mobility versus EC

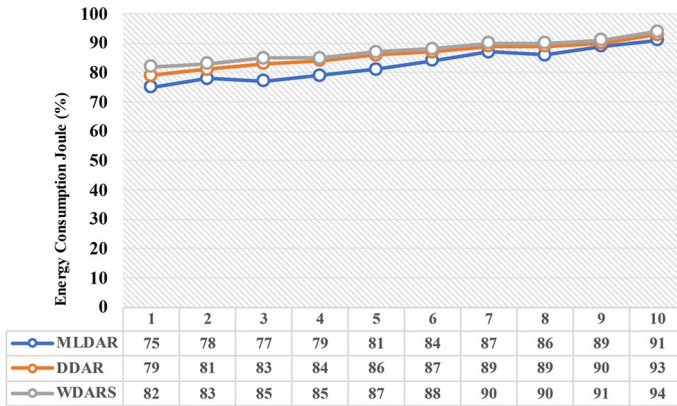


Fig. 27 No. of MN versus EC

64, 128 and 256 kbps, a group of 25, 50, 75, 100, 125, 150, 175, 200, 225, and 250 mobile nodes, and this paper statement roughly 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 are the MN that enter the network and are responsible for malicious attacks (Table 5).

Network Throughput: It equals the total PDR to the base station, split by the simulation time. It is high for the best IoT-WSN performance (Figs. 8, 9, 10, 11).

From the above graph, it is evident that in cases of malicious settings, this proposed protocol, MBLDARP, outperforms other protocols with an 11.23% increase in NT over the DDARP protocol and a 22.69% increase over WDARSP (Figs. 12, 13, 14, 15).

End-to-End Delay: EED is an acronym for "average time". It refers to the time required for data segments to travel from their corresponding source nodes to their destination base station. It is recommended to maintain its standard configuration for enhanced performance of the IoT-WSN (Table 6).

From the above graph, it is evident that in cases of malicious settings, this proposed protocol, MBLDARP, outperforms other protocols with a 9.31% decrease in EED over the DDARP protocol and an 18.91% decrease over WDARSP (Figs. 16, 17, 18, 19).

Routing Overhead: RO refers to the total data packets implemented for route discovery and maintenance processes. It is recommended to maintain its low setting for the greatest performance of the IoT-WSN (Table 7).

From the above graph, it is evident that in the case of malicious settings, this proposed protocol, MBLDARP, outperforms other protocols with a 10.30% decrease in RO over the DDARP protocol and a 19.53% decrease over WDARSP.

Packet Delivery Ratio: PDR is the ratio of data packets at the receiving node to those that were sent (Table 8). It is proposed to work to achieve the highest possible results from the IoT-WSN (Figs. 20,21, 22, 23).

From the above graph, it is evident that in the case of malicious settings, this proposed protocol, MBLDARP, outperforms other protocols, with a 6.85% increase in PDR over the DDARP protocol and a 4.26% increase over WDARSP.

Energy Consumption: EC refers to the energy level implemented by the network nodes to perform computation and communication (Table 9). It is suggested to keep it low for the greatest performance of the IoT-WSN (Figs. 24, 25, 26, 27).

Table 10 MBLDARP recommends QoS comparative evaluation with other protocols in malicious settings (i.e., when some IoT-WSN nodes are MN)

| QoS (%) | DDARP (%) | WDARSA (%) | Nodes (MN) |
|----------------|-----------|------------|---|
| NT (increase) | 11.23 | 22.69 | 25(1), 50(2), 75(3), 100(4), 125(5), 150(6), 175(7), 200(8), 225(9), 250(10) |
| EED (decrease) | 9.31 | 18.91 | |
| RO (decrease) | 10.30 | 19.53 | |
| PDR (increase) | 6.85 | 4.26 | |
| EC (decrease) | 3.94 | 5.48 | |

From the above graph, it is evident that in the case of malicious settings, this proposed protocol, MBLDARP, outperforms other protocols with a 3.94% decrease in EC over the DDARP protocol and a 5.48% decrease over WDARSP (Table 10).

6 Conclusion

This research paper has proposed a Machine Learning Based Data Aggregation and Routing Protocol (MBLDARP), Data Aggregation (DA) and a routing method for the Internet of Things (IoT)- Wireless Sensor Network (WSN). This practical method is a novel work that authenticates nodes' information in DA based on Machine Learning (ML) and ensures data privacy. It also prevents colluding nodes and aggregators and the failure of nodes. The proposed DA protocol maintains nodes' data privacy within the acceptable overhead. This paper also proposes a novel routing protocol based on ML for IoT-WSN. Simulation proved that the proposed protocol is best compared to the existing protocols. These results highlight that the proposed MBLDARP performs significantly better than the currently available methods. The proposed protocol MBLDARP is compared with other protocols, Differentiated Data Aggregation Routing Protocol (DDARP) and Weighted Data Aggregation Routing Protocol (WDARP), as shown in Table 10 of Sect. 4. It is experimentally proven that, in a malicious setting, the proposed protocol has proven improvement in all the considered QoS metrics (*i.e.*,) NT is an 11.23% increase, RO is a 10.30% decrease, EC is a 3.94% decrease, EED is a 9.31% decrease, and PDR is 6.85% increase over DDARP, and NT is a 22.69% increase, EC is a 5.48% decrease, EED is a 18.91% decrease, PDR is a 4.26% increase, and RO is 19.53% decrease over WDARSP.

Funding None.

Availability of Data and Material Not applicable.

Code Availability Not Applicable.

Declarations

Conflict of interest Not applicable.

References

- Hunt, T., Song, C., Shokri, R., Shmatikov, V., & Witchel, E. (2018). "Chiron: Privacy-preserving Machine Learning as a Service," *arXiv*, Mar. 2018, Accessed: May 11, 2021. [Online]. Available: <http://arxiv.org/abs/1803.05961>.
- Nie, J., Luo, J., Xiong, Z., Niyato, D., & Wang, P. (2019). A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing. *IEEE Transactions on Wireless Communications*, 18(1), 724–738. <https://doi.org/10.1109/TWC.2018.2885747>
- Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q. & Qi, H. (2018). "Beyond inferring class representatives: user-level privacy leakage from federated learning," in *Proceedings - IEEE INFOCOM*, vol. 2019-April, pp. 2512–2520, Dec. 2018, Accessed: May 11, 2021. [Online]. Available: <http://arxiv.org/abs/1812.00535>.
- Atapattu, S., Ross, N., Jing, Y., He, Y., & Evans, J. S. (2019). Physical-layer security in full-duplex multi-hop multi-user wireless network with relay selection. *IEEE Transactions on Wireless Communications*, 18(2), 1216–1232. <https://doi.org/10.1109/TWC.2018.2890609>
- Liu, Z., Guo, J., Lam, K.-Y., & Zhao, J. (2022). Efficient dropout-resilient aggregation for privacy-preserving machine learning. *IEEE Transactions on Information Forensics and Security*. <https://doi.org/10.1109/TIFS.2022.3163592>
- Liao, X., Zhang, Y., Wu, Z., Shen, Y., Jiang, X., & Inamura, H. (2018). On security-delay trade-off in two-hop wireless networks with buffer-aided relay selection. *IEEE Transactions on Wireless Communications*, 17(3), 1893–1906. <https://doi.org/10.1109/TWC.2017.2786258>
- Wang, Q., Zhang, Y., Lu, X., Wang, Z., Qin, Z., & Ren, K. (2018). Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 15(4), 591–606. <https://doi.org/10.1109/TDSC.2016.2599873>
- Wang, Z., et al. (2019). Personalized privacy-preserving task allocation for mobile crowdsensing. *IEEE Transactions on Mobile Computing*, 18(6), 1330–1341. <https://doi.org/10.1109/TMC.2018.2861393>
- Wang, Z., et al. (2019). Privacy-preserving crowd-sourced statistical data publishing with an untrusted server. *IEEE Transactions on Mobile Computing*, 18(6), 1356–1367. <https://doi.org/10.1109/TMC.2018.2861765>
- Niu, C., Wu, F., Tang, S., Ma, S., & Chen, G. (2022). Toward verifiable and privacy-preserving machine learning prediction. *IEEE Transactions on Dependable and Secure Computing*, 19(3), 1703–1721. <https://doi.org/10.1109/TDSC.2020.3035591>
- Yuan, D., Li, Q., Li, G., Wang, Q., & Ren, K. (2020). PriRadar: A privacy-preserving framework for spatial crowdsourcing. *IEEE Transactions on Information Forensics and Security*, 15, 299–314. <https://doi.org/10.1109/TIFS.2019.2913232>
- Kittur, L. J., & Pais, A. R. (2023). Combinatorial design based key pre-distribution scheme with high scalability and minimal storage for wireless sensor networks. *Wireless Personal Communications*, 128, 855–873. <https://doi.org/10.1007/s11277-022-09979-2>
- Elangovan, G. R., & Kumanan, T. (2023). Energy efficient and delay aware optimization reverse routing strategy for forecasting link quality in wireless sensor networks. *Wireless Personal Communications*, 128, 923–942. <https://doi.org/10.1007/s11277-022-09982-7>
- Wang, Z., et al. (2019). When mobile crowdsensing meets privacy. *IEEE Communications Magazine*, 57(9), 72–78. <https://doi.org/10.1109/MCOM.001.1800674>
- Butt, U. A., Amin, R., Mehmood, M., et al. (2023). Cloud security threats and solutions: A survey. *Wireless Personal Communications*, 128, 387–413. <https://doi.org/10.1007/s11277-022-09960-z>
- "Apple's 'Differential Privacy' Is About Collecting Your Data--But Not Your Data | WIRED." <https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/> (accessed May 12, 2021).
- Kaliyaperumal, K., Sammy, F. (2022). An efficient key generation scheme for secure sharing of patients health records using attribute-based encryption, in *2022 International Conference on communication, computing and internet of things (IC3IoT)*, Chennai, India, pp. 1–6, <https://doi.org/10.1109/IC3IOTS53935.2022.9767726>.
- Dwork, C., Lei, J. (2009). Differential privacy and robust statistics, in *Proceedings of the annual ACM symposium on theory of computing*, pp. 371–380, <https://doi.org/10.1145/1536414.1536466>.
- Alghamdi, W., Rezvani, M., Wu, H., & Kanhere, S. S. (2019). Routing-aware and malicious node detection in a concealed data aggregation for WSNs. *ACM Transactions on Sensor Networks*. <https://doi.org/10.1145/3293537>
- Araki, T., Furukawa, J., Lindell, Y., Nof, A. & Ohara, K. (2016). High-throughput semi-honest secure three-party computation with an honest majority, in *Proceedings of the ACM conference on computer and communications security*, vol. 24–28, pp. 805–817, <https://doi.org/10.1145/2976749.2978331>.

21. Nandakumar, K., Vinod, V., Batcha, S. M. A., Sharma, D. K., Elangovan, M., Poonia, A., Basavaraju, S. M., Dogiwal, S. R., Dadheech, P., & Sengan, S. (2021). Securing data in transit using data-in-transit defender architecture for cloud communication. *Soft Computing*. <https://doi.org/10.1007/s00500-021-05928-6>
22. Corrigan-Gibbs, H., Wolinsky, D.I. & Ford, B. (2012). Proactively accountable anonymous messaging in verdict. in *Proceedings of the 22nd USENIX security symposium*, pp. 147–162, Accessed: May 12, 2021. [Online]. Available: <http://arxiv.org/abs/1209.4819>.
23. Li, X., Liu, S., Wu, F., Kumari, S., & Rodrigues, J. J. P. C. (2019). Privacy-preserving data aggregation scheme for mobile edge computing assisted IoT applications. *IEEE Internet of Things Journal*, 6(3), 4755–4763. <https://doi.org/10.1109/JIOT.2018.2874473>
24. Liu, Y. N., Wang, Y. P., Wang, X. F., Xia, Z., & Xu, J. F. (2019). Privacy-preserving raw data collection without a trusted authority for IoT. *Computer Networks*, 148, 340–348. <https://doi.org/10.1016/j.comnet.2018.11.028>
25. Abdallah, A., & Shen, X. (2018). A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Smart Grid*, 9(1), 396–405. <https://doi.org/10.1109/TSG.2016.2553647>
26. Chan, T.H.H., Shi, E. & Song, D. (2012). Privacy-preserving stream aggregation with fault tolerance, in *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, vol. 7397 LNCS, pp. 200–214, https://doi.org/10.1007/978-3-642-32946-3_15.
27. Chen, Y., Martinez-Ortega, J. F., Castillejo, P., & Lopez, L. (2019). A homomorphic-based multiple data aggregation scheme for smart grid. *IEEE Sensors Journal*, 19(10), 3921–3929. <https://doi.org/10.1109/JSEN.2019.2895769>
28. Li, S., Xue, K., Yang, Q., & Hong, P. (2018). PPMA: Privacy-preserving multisubset data aggregation in smart grid. *IEEE Transactions on Industrial Informatics*, 14(2), 462–471. <https://doi.org/10.1109/TII.2017.2721542>
29. Liu, Y., Guo, W., Fan, C. I., Chang, L., & Cheng, C. (2019). A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. *IEEE Transactions on Industrial Informatics*, 15(3), 1767–1774. <https://doi.org/10.1109/TII.2018.2809672>
30. Kserawi, F., Al-Marri, S., & Malluhi, Q. (2022). Privacy-Preserving fog aggregation of smart grid data using dynamic differentially-private data perturbation. *IEEE Access*, 10, 43159–43174. <https://doi.org/10.1109/ACCESS.2022.3167015>
31. Wu, H., Wang, L., & Xue, G. (2020). Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing. *IEEE Transactions on Network Science and Engineering*, 7(1), 589–602. <https://doi.org/10.1109/TNSE.2019.2892583>
32. Zhang, X., Wang, W., Mu, L., et al. (2021). Efficient privacy-preserving anonymous authentication protocol for vehicular ad-hoc networks. *Wireless Personal Communications*, 120, 3171–3187. <https://doi.org/10.1007/s11277-021-08605-x>
33. Jegadeesan, S., Obaidat, M. S., Vijayakumar, P., et al. (2022). Efficient privacy-preserving anonymous authentication scheme for human predictive online education system. *Cluster Comput*, 25, 2557–2571. <https://doi.org/10.1007/s10586-021-03390-5>
34. Li, X., et al. (2018). Differentiated data aggregation routing scheme for energy conserving and delay-sensitive wireless sensor networks. *Sensors (Switzerland)*. <https://doi.org/10.3390/s18072349>
35. Zhang, R., Shi, J., Zhang, Y., & Zhang, C. (2013). Verifiable privacy-preserving aggregation in people-centric urban sensing systems. *IEEE Journal on Selected Areas in Communications*, 31(9), 268–278. <https://doi.org/10.1109/JSAC.2013.SUP.0513024>
36. Nanthini, S., Kalyani, S. N., & Sengan, S. (2021). Energy-efficient clustering protocol to enhance network lifetime in wireless sensor networks. *Computers, Materials and Continua*, 68(3), 3595–3614. <https://doi.org/10.32604/CMC.2021.015038>
37. Thiagarajan, A. et al. (2009) VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones, in *Proceedings of the 7th ACM conference on embedded networked sensor systems, SenSys 2009*, pp. 85–98, <https://doi.org/10.1145/1644038.1644048>.
38. Huang, T.K., Lee, C.K. & Chen, L.J. (2010). PRoPHET+: An adaptive PRoPHET-based routing protocol for opportunistic network, in *Proceedings - international conference on advanced information networking and applications, AINA*, pp. 112–119, DOI: <https://doi.org/10.1109/AINA.2010.162>.
39. Lindell, Y., Pinkas, B., Smart, N. P., & Yanai, A. (2019). Efficient constant-round multi-party computation combining BMR and SPDZ. *Journal of Cryptology*, 32(3), 1026–1069. <https://doi.org/10.1007/s00145-019-09322-2>

40. Sheikh, R. & Mishra, D.K. (2019). Secure sum computation using homomorphic encryption, in *Lecture notes on data engineering and communications technologies*, vol. 16, Springer Science and Business Media Deutschland GmbH, pp. 357–363.
41. von Maltitz, M., Bitzer, D. & Carle, G. (2019). Data querying and access control for secure multiparty computation, in *2019 IFIP/IEEE symposium on integrated network and service management, IM 2019*, pp. 171–179, Accessed: May 12, 2021. [Online]. Available: <http://arxiv.org/abs/1901.02651>.
42. Mödinger, D., Hauck, F.J. (2020). 3P3: Strong Flexible privacy for broadcasts, in *2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom)*, Guangzhou, China, pp. 1630–1637, <https://doi.org/10.1109/TrustCom50675.2020.00225>.
43. Narayanasami, S., Sengan, S., Khurram, S., Arslan, F., Murugaiyan, S. K., Rajan, R., Peroumal, V., Dubey, A. K., Srinivasan, S., & Sharma, D. K. (2021). Biological feature selection and classification techniques for intrusion detection on BAT. *Wireless Personal Communications*. <https://doi.org/10.1007/s11277-021-08721-8>
44. Rayati, M., Bozorg, M. (2022). Pricing differentially private smart meter data in distribution networks, in *18th International conference on the European Energy Market (EEM)*, Ljubljana, Slovenia, pp. 1–6, <https://doi.org/10.1109/EEM54602.2022.9921095>.
45. Jansen, R., Johnson, A. (2021). Safely Measuring Tor, Accessed: May 12, 2021. [Online]. Available: <https://doi.org/10.1145/2976749.2978310>.
46. Vahdat, A., Vahdat, A. & Becker, D. (2021). “Epidemic Routing for Partially-Connected Ad Hoc Networks,” 2000, Accessed: May 12, 2021. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.6151>.
47. Boldrini, C., Conti, M., Iacopini, I. & Passarella, A. (2007). HiBOP: A history-based routing protocol for opportunistic networks, <https://doi.org/10.1109/WOWMOM.2007.4351716>.
48. Dhurandher, S.K., Sharma, D.K., Woungang, I. & Bhati, S. (2013). “HBPR: History-based prediction for routing in infrastructure-less opportunistic networks, in *Proceedings - international conference on advanced information networking and applications, AINA*, pp. 931–936, <https://doi.org/10.1109/AINA.2013.105>.
49. Lindgren, A., Doria, A., & Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3), 19–20. <https://doi.org/10.1145/961268.961272>
50. Dhurandher, S.K., Borah, S., Woungang, I., Sharma, D.K., Arora, K. & Agarwal, D. (2016). EDR: An encounter and distance-based routing protocol for opportunistic networks, in *Proceedings - International conference on advanced information networking and applications, AINA*, vol. 2016, pp. 297–302, DOI: <https://doi.org/10.1109/AINA.2016.15>.
51. Khazaei, J., & Amini, M. H. (2021). Protection of large-scale smart grids against false data injection cyberattacks leading to blackouts. *International Journal of Critical Infrastructure Protection*, 35, 100457.
52. Zhao, P., et al. (2018). P3-LOC: A privacy-preserving paradigm-driven framework for indoor localization. *IEEE/ACM Transactions on Networking*, 26(6), 2856–2869. <https://doi.org/10.1109/TNET.2018.2879967>
53. Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11), 612–613. <https://doi.org/10.1145/359168.359176>
54. Jung, T., Li, X. Y., & Wan, M. (2015). Collusion-tolerable privacy-preserving sum and product calculation without secure channel. *IEEE Transactions on Dependable and Secure Computing*, 12(1), 45–57. <https://doi.org/10.1109/TDSC.2014.2309134>
55. Zhang, L., Li, X.Y., & Liu, Y. (2013) Message in a sealed bottle: Privacy-preserving friending in social networks, in *Proceedings - international conference on distributed computing systems*, pp. 327–336, <https://doi.org/10.1109/ICDCS.2013.38>.
56. Jung, T., Li, X.Y., Wan, Z. & Wan, M. (2013). Privacy-preserving cloud data access with multi-authorities, in *Proceedings - IEEE INFOCOM*, pp. 2625–2633, DOI: <https://doi.org/10.1109/INF-COM.2013.6567070>.
57. Goldwasser, S., Micali, S., & Rackoff, C. (1989). Knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1), 186–208. <https://doi.org/10.1137/0218012>
58. Kserawi, F., Malluhi, Q.M. (2020). Privacy preservation of aggregated data using virtual battery in the smart grid, in *Proceedings of the IEEE 6th international conference on dependability sensor cloud big data syst. Appl. (DependSys)*, pp. 106–111.
59. Duda, R.O., Hart, P.E., Stork, D.G. (2021). *Pattern classification*, 2nd edn, Wiley.” <https://www.wiley.com/en-sg/Pattern+Classification%2C+2nd+Edition-p-9780471056690> (accessed May 12, 2021).

60. Khan, Z. M. A., Saeidlou, S., & Saadat, M. (2019). Ontology-based decision tree model for prediction in a manufacturing network. *Production and Manufacturing Research*, 7(1), 335–349. <https://doi.org/10.1080/21693277.2019.1621228>
61. Adil Mahdi, O., Abdul Wahab, A. W., Idris, M. Y. I., Abu Znaid, A., Al-Mayouf, Y. R. B., & Khan, S. (2016). WDARS: A weighted data aggregation routing strategy with minimum link cost in event-driven WSNs. *Journal of Sensors*. <https://doi.org/10.1155/2016/3428730>
62. Liu, J. N., Weng, J., Yang, A., Chen, Y., & Lin, X. (2020). Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid. *IEEE Trans. Smart Grid*, 11(1), 247–257.
63. Bhushan, S., Kumar, M., Kumar, P., Stephan, T., Shankar, A., & Liu, P. (2021). FAJIT: A fuzzy-based data aggregation technique for energy efficiency in wireless sensor network. *Complex and Intelligent Systems*, 7(2), 997–1007. <https://doi.org/10.1007/S40747-020-00258-W>
64. Chen, Z., Long, X., Wu, Y., Chen, L., Wu, J. & Liu, S. (2020). Data aggregation aware routing for distributed training, in *Lecture Notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, vol. 12606 LNCS, pp. 241–250, https://doi.org/10.1007/978-3-030-69244-5_21.
65. Visu, P., Praba, T. S., Sivakumar, N., Srinivasan, R., & Sethukarasi, T. (2020). Bio-inspired dual cluster heads optimized routing algorithm for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 12(3), 3753–3761. <https://doi.org/10.1007/S12652-019-01657-9>
66. Alharbi, M. A., Kolberg, M., & Zeeshan, M. (2021). Towards improved clustering and routing protocol for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2021(1), 1–31. <https://doi.org/10.1186/S13638-021-01911-9>
67. Maivizhi, R., & Yogesh, P. (2021). Q-learning based routing for in-network aggregation in wireless sensor networks. *Wireless Networks*, 27(3), 2231–2250. <https://doi.org/10.1007/S11276-021-02564-8>
68. Feroz Khan, A. B., & Anandharaj, G. (2021). A cognitive energy efficient and trusted routing model for the security of wireless sensor networks: CEMT. *Wireless Personal Communications*, 119(4), 3149–3159. <https://doi.org/10.1007/S11277-021-08391-6/METRICS>
69. Feroz Khan, A. B., Kalpana Devi, H. L. R. S., & Rajalakshmi, C. N. (2022). A multi-attribute based trusted routing for embedded devices in MANET-IoT. *Microprocessors and Microsystems*, 89, 104446. <https://doi.org/10.1016/J.MICPRO.2022.104446>
70. Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H. P., & Aaraj, N. (2022). Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10), 1572–1609. <https://doi.org/10.1109/JPROC.2022.3205665>
71. Al Badawi, A., Polyakov, Y., Aung, K. M. M., Veeravalli, B., & Rohloff, K. (2021). Implementation and performance evaluation of RNS variants of the BFV homomorphic encryption scheme. *IEEE Transactions on Emerging Topics in Computing*, 9(2), 941–956.
72. Aloufi, A., Hu, P., Song, Y. & Lauter, K. (2020). Computing blindfolded on data homomorphically encrypted under multiple keys: An extended survey, [arXiv:2007.09270](https://arxiv.org/abs/2007.09270).
73. Mono, J., Marcolla, C., Land, G., Güneysu, T. & Aaraj, N. (2022). "Finding and evaluating parameters for BGV", *Cryptol. ePrint Arch.*.
74. Ara Begum, B., & Nandury, S. V. (2023). Data aggregation protocols for WSN and IoT applications – A comprehensive survey. *Journal of King Saud University - Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2023.01.008>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Neeraj Chandnani received his Bachelor of Engineering (Hons) degree (2013) in Electronics and Communication Engineering from Rajiv Gandhi Pradyogiki Vishwavidyalaya, Bhopal, Madhya Pradesh, India, and Master of Engineering (Hons) degree (2016) in Electronics and Communication Engineering with specialization in Embedded System and VLSI Design from the Rajiv Gandhi Pradyogiki Vishwavidyalaya, Bhopal, Madhya Pradesh, India. At present, he is research scholar at Military College of Telecommunication Engineering, Mhow, research centre, affiliated to Devi Ahilya Vishwavidyalaya, Indore, and pursuing his Ph.D. degree in the field of Electronics and Communication Engineering with specialization in Internet of Things (IoT). His current research interests are Internet of Things (IoT), Wireless Sensor Networks (WSNs), and Microcontroller-based System Design. He has to his credit highly influential research papers at International Science Citation Indexed Journals and Scopus Indexed Conferences of repute, including IEEE and Springer.



Chandrakant N. Khairnar is currently serving in the Faculty of Communication Engineering, Military College of Telecommunication Engineering (MCTE), Mhow (M.P). He received his Bachelor of Engineering in Electronics, from Pune University. Master of Technology in Electronics from Visvesvaraya Regional Engineering College, Nagpur (Now VNIT, Nagpur). He is Ph.D. in the field of Electronics and Telecommunication discipline from Faculty of Engineering, Devi Ahilya Vishwa Vidhyalaya (DAVV), Indore. He is supervising Ph.D. scholars in the field of Internet of Things, Software Defined Radios and Cognitive Radios. He has to his credit large number of research papers at National and International platforms. He is recipient of Chief of Army Staff (COAS) Certificate of Merit for outstanding contribution in developing an innovation for enhancing the operational/logistics capability of Indian Army. Also, recipient of General Officer Commanding-in-Chief, Army Training Command Commendation Card in recognition of devotion to duty and distinguished service. He has about 26 years' experience of academic institutions and industry. He is life

member of Indian Society of Technical Education (ISTE) and Fellow Life member of Institute of Electronic and Telecommunication Electronic (FIETE).