Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2437578, 12 pages
https://doi.org/10.1155/2022/2437578

Research Article

Network Lifetime Enhancement by Elimination of Spatially and Temporally Correlated RFID Surveillance Data in WSNs

Lucy Dash,[1] Binod Kumar Pattanayak,[1] Debabrata Singh,[2] Debabrata Samanta,[3] and Yagyanath Rimal[4]

[1]*Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar, India*
[2]*Department of Computer Application, Siksha 'O' Anusandhan University, Bhubaneswar, India*
[3]*Department of Computer Science, CHRIST (Deemed to Be) University, Bengaluru, Karnataka 560029, India*
[4]*Department of Computer Science, Pokhara University, Pokhara 30, Khudi, Kaski, Gandaki, Nepal*

Correspondence should be addressed to Debabrata Singh; debabratasingh@soa.ac.in,
Debabrata Samanta; debabrata.samanta369@gmail.com, and Yagyanath Rimal; rimal.yagya@gmail.com

Received 16 February 2022; Revised 22 July 2022; Accepted 30 July 2022; Published 7 September 2022

Academic Editor: Chuanwen Luo

In wireless sensor networks (WSNs), radio frequency identification (RFID) plays an important role due to its data characteristics which are data simplicity, low cost, simple deployment, and less energy consumption. It consists of a series of tags and readers which collect a huge number of redundant data. It increases system overhead and decreases overall network lifetime. Existing solutions like Time-Distance Bloom Filter (TDBF) algorithm are inapplicable to the large-scale environment. Received Signal Strength (RSS) used in this algorithm is highly dependent on quality of tag and application environment. In this paper, we propose an approach for data redundancy minimization for RFID surveillance data which is a modified version of TDBF. The proposed algorithm is formulated by using the observed time and calculated distance of RFID tags. To overcome these problems, we design our approach to relevantly reduce the spatiotemporal data redundancy in the source level by adding the Received Signal Strength Indicator (RSSI) concept for energy-efficient RFID data communication in wireless sensor network scenario. We introduce in this paper the new improved idea of an existing algorithm which efficiently reduces the rate of data redundancy spatially and temporally. The implemented results overcome the limitations of existing algorithm for data redundancy reduction. Nevertheless, the performance evaluation shows the efficiency of proposed algorithm in terms of time and data accuracy. Furthermore, this algorithm supports multidimensional and large-scale environment suitable for sensor network nowadays.

1. Introduction

In wireless sensor network, electromagnetic signals from sensor nodes are detected by RFID technology which consists of antenna, transceiver, and tags optimized with technical programs. Antennas build the emitted radio signal communication bridge between the tag and the transceiver emitted for tag activation and data encryption-decryption [1]. RFID has a wide range of application in agriculture, military, defence, health care, supply chain management, logistics, access control, IT access tracking, material management, race timing, tool tracking, video surveillance, product tracking, and payment. It is vastly used in passenger identification in airport and postal tracking [2]. A simple example of RFID is NFC (Near Field Communication) tag reader in smartphones. NFC supports reader modes which are having operations as writer or reader, tag identification, and peer-to-peer tracking. For simplified meaning, NFC reader tags used in iPhone can enable devices within the perimeter of few centimetres for wirelessly exchanging information among each other. Applications in iOS (iPhone operating system) mobile operating system can read the scanned data through NFC tag reader attached to the real-environment objects. For example, a working employee from retail sector can scan different products for tracking, or the mobile users can scan their devices attached to tag reader in order to equip their video game feature with any toy [3].

In access monitoring field, RFID technology is used for accurate tag data measurement. Risk assessment due to real-time performance of RFID tags is an influencing factor affecting the loss of tag information, advanced risk detection, and real-time event handling. According to the RFID reader architecture, it collects a huge number of relevant data within the detection range of the reader which eventually leads to the collection of redundant data. Massive gathering of redundant data can lead to system overhead and loss of important information. It leads to storage space scarcity, increased network latency, and decreased quality of service (QoS). So it is important to reduce the number of redundant data for loss-less data communication, efficiency increase of RFID systems, and enhanced overall network lifetime. RFID tags work densely in WSNs and due to high density in the sensor environment; observations are highly correlated in the space domain. The nature of network topology and physical phenomenon constitutes the temporal correlation among tags [4]. These spatial and temporal correlations bring huge number of redundant data for consecutive observations between RFID reader and RFID tag. Communication operation involves huge power consumption, as it is associated with activities like collision of data, overemission of sensing information, overhearing of sensed information, and idle channel listening in absence of communications. These spatially and temporally correlated redundant data need to be filter out for efficient overall system management of RFID technology. In RFID system, most of the data are static which can be filtered out easily as compared to the dynamic data. In this paper, we have proposed a modified version of Time-Distance Bloom Filter (TDBF) proposed by Wang et.al in [5] to eliminate the existing limitations of this algorithm for RFID redundant data elimination and filtering out the irrelevant data. We have renamed it as mTDBF (Modified Time-Distance Bloom Filter). Experimental results show that this approach can efficiently improve the performance of RFID system in real scenario as compared with the existing methods [6]. The rest of the paper is descripted as follows. In Section 2, the RFID system model is described, followed by RFID data and its characteristics in brief in Section 3, followed by data characteristics in Section 4. Section 5 presents the signal characteristics and real-time performance. Section 6 presents the related works. The proposed system model is discussed in Section 7. Section 8 discusses the experimental results, performance evaluation, and comparison. Section 9 presents the conclusion and future work.

2. RFID System Model

RFID technology is an emerging technology which is used for various applications like supply chain management, habitat monitoring, and wild fire detection [7]. It consists of RFID tags, RFID reader (active and passive), and software (middleware) as described in Figure 1.

2.1. RFID Tag. RFID tags associated with the objects are responsible for identification without using line of sight eliminating the limitations of barcode [8]. RFID tags are

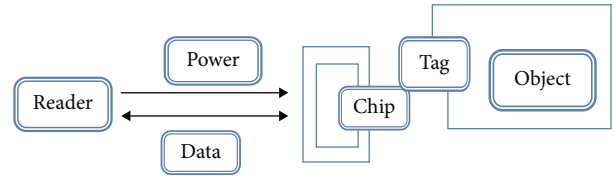

Figure 1: Architecture of RFID tag.

made of three different components: an RFID chip, which is an integrated circuit (IC), an antenna, and a substrate. A tag manufacturer typically does not make all three components in-house. The IC is typically designed and made by a semiconductor manufacturer, while an antenna is usually designed and made by a tag manufacturer. Tags are available in various sizes, designs, and form factors, and they can be customized for a particular application. An electronic circuit, microchip, or chip is designed and manufactured by a semiconductor manufacturer. The IC needs power to operate. This power may come from a battery on the tag (in an active tag), or it may be obtained from the radio energy radiated by the interrogator antenna. The processing logic implements the communication protocol. It also is used to modulate/demodulate signals during the communication between the interrogator and the tag. Making tag ICs more efficient in power usage and requiring less power to operate increases the read range of passive tags.

2.2. RFID Reader. RFID readers are responsible for collaborating relevant information from various tags within the read range. It can also alter the tag information according to need. EPC (Electronic Product Code) Class 1 Gen 2 RFID standard [9] is one of the suitable reader for various applications. Active readers are equipped with battery power supply whereas passive readers are free from a finite source of power supply as shown in Figure 2.

2.3. Middleware. RFID middleware is responsible for collecting and processing the data received from readers and converting it into relevant information as described in Figure 3.

2.4. Workflow of RFID. UHF (ultrahigh frequency) RFID tags are vastly used in many applications of WSNs. It ranges from 433 MHz (megahertz) to 956 MHz; our work is focused on UHF RFID tags as shown in Figure 4.

In RFID system, the reader collects the data from the tags adhered with the objects in the real environment. The middleware processes the data with some actions such as aggregation, filtering, and transforming raw data to information. Then, these processed and filtered data proceed to the application server for next execution of events such as sending theft alarms, fire detection, gas explosion, and earthquake detection [10].

3. RFID Data and Its Characteristics

3.1. Data Description. Basically, a RFID tag storehouse EPC (Electronic Product Code) [11] is a unique identification entity. Tags attached to the reader when comes under the

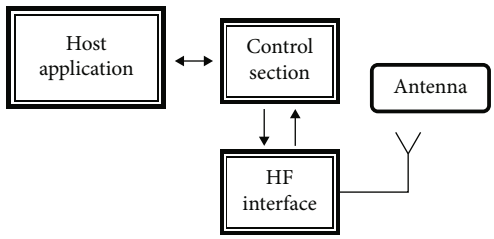

Figure 2: RFID reader architecture.

reading range can generate EPC. Air Interface Protocol is used for reading the tag code. An inbuilt clock synchronises the read tad data along with the time of observation. The time duration is known as epoch time [12]. A number of observations are recorded in epoch time consisting of various time stamps known as epoch duration. These readings are optimized to the middleware depending upon the requirements of the application. Middleware data server receives the information along with location, and time can be represented as

$$(\mathrm{EPC}_i, \mathrm{loc}_i, t_i), \tag{1}$$

where EPC_i is the code of the tag, loc_i is the location of the reader, t_i is the current time, and $t_1, t_2, \cdots, t_i$ is the epoch duration.

3.2. Spatiotemporal RFID Data. In RFID, the timestamp of a tag exhibits the spatial change of a tag in a timestamp due to its space-time functionality. Related events are being identified by tags are influenced by the time-space association of data. Due to multilocational data, a single RFID reader can exhibit multiple same data in the specified time interval [13]. This generates temporally and spatially correlated redundant data for a single reader. RFID readers collecting information of different tagged items form a group. A series of observations of multiple RFID tags exhibits temporal correlation among each read data for a specific timestamp. Data representation groups multiple EPC IDs for different timestamps in a time-interval yield spatial correlation [14]. Exploiting spatial and temporal correlation among different RFID readers with unique $\mathrm{tag}_{\mathrm{ID}}$ can reduce the redundant data by the following ways:

(1) By reducing the false negative rate by minimizing the percentage of epochs in a certain time interval (ρ value) of multiple observations

(2) By reducing the false positive rate by maximizing the ρ value

False negative rate arises when multiple tags are detected simultaneously or a tag is not detected due to radio frequency interference (noise). False positive rate arises when tags read captures the data outside the read range or readers get affected by environment. Getting the nonredundant data as a subset of codes which is a part of larger group can efficiently yield the redundant data [15].

4. Data Characteristics

RFID data has some unique characteristics which make it suitable for large-scale environment and multidimensional application scenario.

4.1. Large Volume of Redundant Data. RFID technology is a unique technology as it generates hundred to thousand times the data volume as compared to existing barcode technique. This large volume of data eventually yields a series of redundant data increasing the overall system overhead. In an ultrahigh frequency, the RFID devices frequently collect the data due to low-cost communication between the readers and the tags [16].

4.2. Unreliable Data. RFID data are very simple in nature. This simplicity attribute of data is represented as ID, loc, and t where ID is the $\mathrm{tag}_{\mathrm{ID}}$, loc is the location, and t is the time. Due to this simplicity nature, the data can be recorded and altered without any extra cost or communication in the data warehouse. Data stream produced by RFID readers is often inaccurate which marks the drawback of using RFID technology. In real-world scenario, the read rate of RFID readers is approximately 70%. Inaccuracy of read data often generates high error rate and important data loss in data warehouse [17].

4.3. Query Processing. Frequent queries are processed as RFID data are collected hugely due to their wide range of applications. In presence of active tags, the generated sensor data get refined by execution of complex queries to meet the emerging challenges. Depending upon the large intensity of data, the queried data are authenticated by frequently updating the $\mathrm{tag}_{\mathrm{id}}$ information. Physical movement of the data leads to its mobility and hence identifies the location of the RFID tag. For nonmobile or static tag, several authentications have been carried out for ensuring its presence in the prescribed location. Increase in volume of data may generate frequent RFID tag polling [18].

4.4. Privacy Concerns. In management systems like supply chain management, ticket countering system, and blockchain management system, the privacy issues of customers who are in association with the objects physically connected to the tag are of paramount significance. In connection to this, the RFID tags go through malicious attacks like Denial of Service (DoS) attack, Man-in-Middle attack, and Distributed Denial of Service (DDoS) attack. However, the attacked tags may replicate it and use necessary information violating the user's copyrights. The unintended user can manipulate the information as RFID generates rich source of information. Thus, data must be made secure generated by RFID tags [19].

5. Signal Characteristics and Real-Time Performance

RFID data are based on electromagnetic radio frequency waves which have different range from low to high. The range of frequencies can be categorized into 4 types which are described as follows in Table 1.

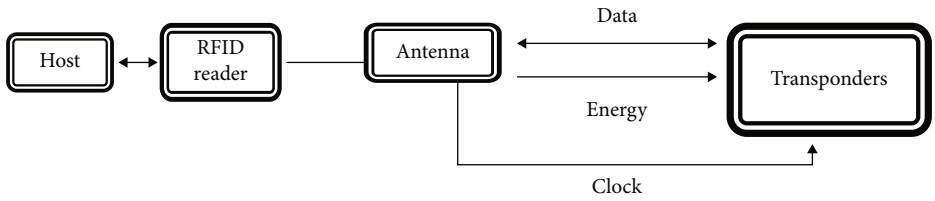

Figure 3: RFID middleware architecture.

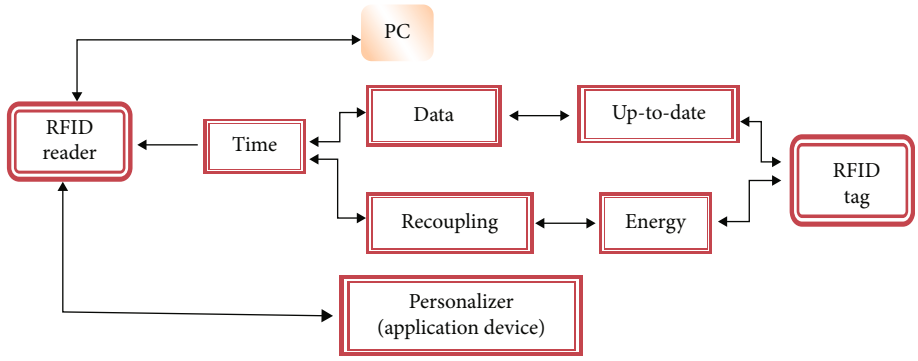

Figure 4: Overall system workflow of RFID.

Table 1: RFID frequency bands and its description.

Frequency band	Range (in hertz)	Data transfer rate (in kilobits)	Detection range (in meters)	Penetrating device	Applications
Low	125 KHz to 134 KHz	1	0.5	Water	Animal identification
High	13.56 MHz	25	1.5	Water	Access and security
Ultrahigh	433 MHz to 956 MHz	100	0.5 to 5	None	Logistics
Micro	2.45 GHz	100	10	None	Mobile vehicle toll

Electromagnetic radio signals read by readers from tags in RFID technology have important characteristics known as Received Signal Strength (RSS) [20]. It is a set of scalars associated with some linear equations. It is inversely proportional to the distance of transmission, i.e., RSS value decreases with the increase in transmission distance and vice versa. RSS gets affected by variation in latency. So it needs to be filtered to efficiently utilise the read rates in short distance [21].

5.1. Clustering Algorithms. In many situations, the data collected by many nodes will be the same. In such cases, redundant data transmission can be eliminated by forming group of nodes called clusters and by electing one node among the nodes in the cluster to be cluster head. All nodes can send data to the cluster head where the aggregation of data can take place. There are two types of clustering techniques. The clustering technique applied in homogeneous sensor networks is called homogeneous clustering schemes, and the clustering technique applied in the heterogeneous sensor networks is referred to as heterogeneous clustering schemes. If we have used fixed node as the cluster head, then it has to collect data from all of its child nodes and has to process the data for all the time period. This leads to faster battery drainage in the fixed cluster head. Even if one cluster head dies, it will affect the working of the network. By choosing dynamic cluster head, this problem can be eliminated. LEACH is an example of clustering protocol for wireless sensor network which considers homogeneous sensor networks where all sensor nodes are designed with the same battery energy. HEED and PEGASIS are some of the other examples of the clustering algorithm.

5.2. LEACH. It stands for low-energy adaptive clustering hierarchy which is the first protocol of hierarchical routing which proposed data fusion, and it is of milestone significance in clustering routing protocol. LEACH minimizes the communication energy that is dissipated by the cluster heads and the cluster members as much as 8 times when compared with direct transmission and minimum transmission energy routing. LEACH incorporates randomized rotation of the high-energy cluster-head position such that it rotates among the sensors in order to avoid draining the battery of any one sensor in the network. In this way, the energy load associated with being a cluster head is evenly distributed among the nodes. Since the cluster-head node knows

all the cluster members, it can create a TDMA schedule that tells each node exactly when to transmit its data. In addition, using a TDMA schedule for data transfer prevents intracluster collisions. The operation of LEACH is divided into rounds. Each round begins with a set-up phase when the clusters are organized, followed by a steady-state phase where several frames of data are transferred from the nodes to the cluster head and onto the base station.

5.3. HEED. It stands for hybrid energy-efficient distributed clustering. HEED is one of the most effective cluster-based routing protocols in WSN. HEED has four primary objectives:

(i) Prolonging network lifetime by distributing energy consumption

(ii) Terminating the clustering process within a constant number of iterations

(iii) Minimizing control overhead (to be linear in the number of nodes)

(iv) Producing well-distributed cluster heads

It is a distributed, energy efficient clustering approach which makes use of two parameters to cluster the network: the sensor residual energy as a primary parameter and intracommunication like node degree and node proximity as a secondary parameter. The HEED operation for clustering is divided into three phases: the initialization phase in which the sensors put their probabilities to become CHs, the main processing phase in which the sensors go through many steps to elect the CHs, and the finalization phase in which each sensor joins the least communication-cost CH or announces itself as a CH. The reclustering in HEED is triggered dynamically at the beginning of each round which is a predefined period of time; the round in HEED can be in the range of seconds, minutes, or even hours depending on the application at hand. The HEED clustering operation is invoked at each node in order to decide if the node will elect to become a cluster head or join a cluster. A cluster head is responsible for two important tasks:

(1) Intracluster coordination, i.e., coordinating among nodes within its cluster

(2) Intercluster communication, i.e., communicating with other cluster heads and/or external observers

The cluster range or radius is determined by the transmission power level used for intracluster announcements and during clustering. We refer to this as the cluster power level. The cluster power level should be set to one of the lower power levels of a node, to increase spatial reuse, and reserve higher power levels for intercluster communication. Selecting cluster heads is based on two parameters: a primary parameter and a secondary one. HEED uses the primary parameter to probabilistically select an initial set of cluster heads and the secondary parameter to "break ties." A tie in this context means that a node falls within the "cluster range" of more than one cluster head. The mechanism of breaking ties according to cost also ensures that the probability of having two cluster heads within the same cluster range is minimized, i.e., cluster heads are well-distributed in the network.

5.4. PEGASIS. System PEGASIS is another hierarchical routing protocol which considered as an improvement over LEACH. PEGASIS stands for power-efficient gathering in sensor information system. In PEGASIS, the primary idea is having each node to receive from and transmit to adjacent neighbours and then each node will take its turn later to be the chain leader. The nodes in PEGASIS are organized to form a chain either by the sensors themselves using a greedy algorithm starting from the randomly chosen node, usually the farthest nodes from the sink, or by having the sink construct the chain and transmits these information to the rest of sensors. In PEGASIS, the data aggregation is performed at every node on the chain except the end nodes in the chain and the network topology is assumed to be known. PEGASIS performs better than LEACH because it reduces the consumed energy in its phases. In its local gathering stage, the summation of distances among transmitting nodes is less than transmitting to a CH in LEACH. Also, the amount of data received by the leader of chain is much less from that in LEACH. Finally, in each round, only there is one node envoys the collected data to the sink node.

6. Related Works

Tremendous works have been done in the field of RFID regarding duplicate data elimination. We discuss here some of the works done so far. Traditionally, the data warehouse stores all the data after redundancy processing algorithm. Then, relevant query processing is done. But due to large volume of data stream in RFID, inaccurate queries are being processed sometimes. So, RFID cube has been evolved as a basic concept of filtering the redundant data according to the tag locations related to the movement and static state of the tags [22]. To eliminate this limitation, sliding window concept was evolved which stores the smaller size of data as compared to the actual size. But it faces problem when flooded data stream encounters smaller window size or a large window size computes the appropriate data with huge time constraint [23]. Depending on the type of RFID data stream, various works are done so far. A data cleaning algorithm (Improved SMRUF) is proposed by Wang et al. which is based on probability which considers the time complexity of sliding window. This adaptive process reduces redundancy for dynamic tags [24]. Luo et al. have proposed another approach for dynamic tags which takes time tolerance into account and sets up a threshold value for it to simplify the data instead of filtering [25]. Sometimes this kind of technique increases the system overhead due to nonfiltering. To eliminate this type of problems, some techniques like Dynamic Bayesian Networks (DBNs) [26] and Finite State Machine [27] are seen into picture. DBNs focus on not recording new data stream as the weight of new data is configured with the observed value of the data. Finite State

Input: RFID Data
x : $x.tid$, x time, $x.RSS$, $x.loc$, $x.RSSI$
Output: x is Redundant or not
Step 1: Begin
Create two events
Event (Hash mapping of arrival tid)
Event-check(for comparison)
for K no. of IDs(1)
$P[i] = Hash_i(x.tid)$[Hash mapping]
Step 2: Initiate proposed algorithm
for K no.of IDs(i)
if $\text{TDBF}_{P[i]}\text{x} \cdot \text{RSSI} \neq 0$ [Already detected that is why RSSI ≠ 0]
Update TDBF(x · RSSI)
else [detected for the first time]
if x · RSS > ω[no need to check time as already checked in RSSI condition above.
Only check the threshold within the range]
Update TDBF (x.time, x.RSS)
else if $x \cdot \text{time-TDBF}_{P[i]}[time] > \tau$ and $x \cdot RSS > \omega$
[x is in range or not decided by RSS and
whether after certain time interval τ checked by first condition]
Update TDBF (x.time, $x.RSS$)
Send x to Event$_c heck$
break
elseifx.timer¿$\text{TDBF}_{\text{P[i]}}$ [time]
and $x.RRS > TDBF_{P[i]}[RSS]$
Update TDBF (x.time, $x.RSS$)
Send x to Event check
break
end if
Identify x by comparing Event and Event check
END

Algorithm 1: Proposed algorithm (m-TDBF).

Machine filters the valid data according to the state machine but it is not suitable for large-scale environment.

To eliminate such kind of inefficiency, Bloom filter concept came into the picture. Bloom filter is a data structure based on probability of existence of an element belonging to the collection or not. If we compare the features of Bloom filter with that of data structure, then the result is very obvious that Bloom filter is more efficient as it is independent of storage capacity based in terms of space and time [28]. In [29], log-log-Bloom-filter concept is taken to estimate the smart grid data processing whereas in [30], Hyper-Log-Log Bloom Filter describes the performance improvement of cardinality estimation of large datasets. In [31], Compare Bloom Filter concept is evolved which is responsible for data filtering in physical space dimension. Approximate Probability Synthesis Bloom Filter (PSBF) concept in [32] filters and removes the overlapped redundant data based on probabilistic occurrences. Time Interval Bloom Filter proposed by Chen et al. in [33] takes into account the total number of timestamps in a total time period to evaluate and filter the redundant data. Time Space Bloom Filter discernments the mobility of the tags of RFID detected by based on time-space features [34]. All these algorithms based on Bloom filter concept are very efficient in terms of exploiting spatial and temporal characteristics of data redundancy but these algorithms fail to do the needful in case of real-world scenario and changing environment conditions. Also, the tag quality of the RFID is a factor which lacks these concepts behind. In order to find out the fittest and efficient algorithm for real-world scenario and to work it out in a certain range, in this paper, we propose the modified version of existing TDBF algorithm. The experimental results and performance evaluation along with comparison yield it to be fit to remove the limitations of TDBF algorithm.

6.1. RFID Redundant Data. RFID redundant data can be defined as invalid data reading which comes after first reading repeatedly. The inbuilt architecture of RFID makes it prone to collect repeated data not only from the nearest tags but also nearby readers adjacent to it. The environmental conditions also affect it. Usually, RFID data are listed as a description of tuple which can be written as $<\text{ID}_{\text{tag}}, \text{loc}, t>$ for capturing the data from the reader. For example, the RFID tags present near any device from the entrance of a shopping mall can read the data from the entrance as well as the readings of data captured from the RFID readers employed near the entrance gate. So the repetitive data can lead to the system overhead decreasing the overall network lifetime. To analyse the data complexity, it is important to note that a single object should be read once by the reader only.

In order to reduce the complexity of data analysis, previous works assume that each object is read once and read

by one reader only. Clearly, this assumption is difficult to enforce, and more importantly, it oversimplifies the reality. Because the RFID readings are of low quality, many applications have to employ nonredundant readers to cover the target area completely in order to improve the localization accuracy [35], which means the objects are read by multiple readers simultaneously. Indeed, in RFID systems, the spatial redundancy is very common. A RFID reader is located at the centre of each zone. Spatial overlapping of reader's detection ranges often leads to duplicate readings carried away by the multiple readers. Temporal overlapping can be formed by the duplicate readings recorded at a particular timestamp [36].

6.2. Bloom Filter. A Bloom filter can be classified as a kind of data structure which evaluates the presence of any element in a set or not. It consists of a fixed sized array and independent hash functions mapping to the elements of the array. Initially, the value is set to be 0, and gradually, it is incremented to 1 according to the mapping of hash function to the different elements of the set. To examine whether an element is present in the set, the numerical value 1 decides the needful as in Bloom filter; this value decides the existing element. When a new element arrives in the set, 0 indicates the newly arrive element in the set [37]. In RFID, the tag IDs are inserted in the set and checked their presence by hash functions for existence of duplicate values present in the set. Changed mapping positions of different tag IDs of RFID tags can be indicated from 0 to 1 depending on the examination of newly arrived tag IDs to the bit array [38]. Bloom filter takes the advantage of independent hash functions by saving all the incoming data in terms of space and time which makes it more efficient as compared to the other data structures. In terms of security and privacy concerns, the traditional Bloom filter does not overload the system by storing all the incoming data which makes it inefficient in terms of data rate loss, increased false negative rates, and decreased false positive rates [39].

6.3. TDBF Data Redundancy Approach. As compared to the traditional Bloom filters, the TDBF (Time-Distance Bloom Filter) algorithm can be extensively used to filter the repetitive data coming from the tags of the RFID readers. It takes into consideration of the parameters like time, distance, and Received Signal Strength (RSS) [40]. RSS is greatly affected by tag quality and environmental situations. It has some fluctuations in reading with the increase and decrease in distance. It has the advantages of limited error rate along with higher performance measures. TDBF algorithm measures the distance uniquely at particular space and time. Starting at a fixed sized array, the algorithm is initialized to a user defined notation where each cell represents the two-dimensional integer arrays consisting of time and RSS, respectively. The recorded timestamps of read tags are presented by the negative values to make the first value of TDBF as 0. The recoded distance value is a user-defined threshold function integer value. Upon arrival of new RFID data, the hash functions

Table 2: Comparison of compression rate of three datasets for TDBF and m-TDBF.

Algorithm	Data 1	Data 2	Data 3
m-TDBF	5.000	5.000	3.330
TDBF	5.000	2.000	2.000

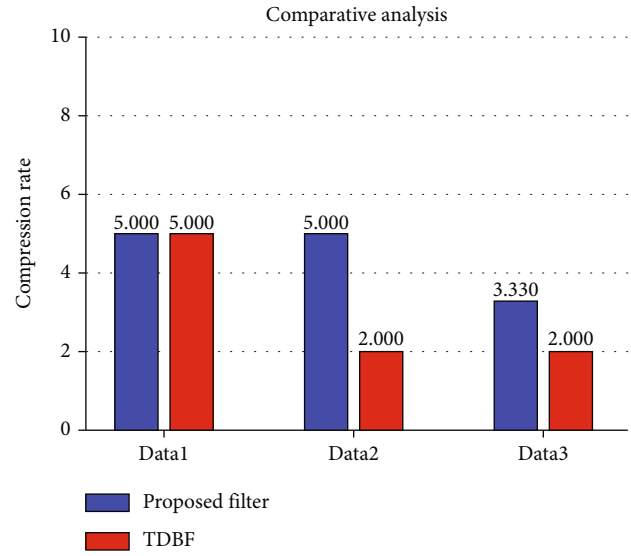

Figure 5: Comparison of 3 datasets in terms of compression ratio.

Table 3: Comparison of space saving of three datasets for TDBF and m-TDBF.

Algorithm	Data 1	Data 2	Data 3
m-TDBF	60.000	80.000	90.000
TDBF	60.000	60.000	70.000

filter the duplicate value on the basis of recorded timestamp and the RSS value of that particular tag. After detection of duplicate data, it deletes the unnecessary stuffs and keeps the useful information. Data compression ratio is considered as an important performance measure to detect the efficiency of filtering the duplicate data. The false negative rate (FNR) [38] and false positive rate (FPR) are also evaluated to measure the efficiency of filtering the data. Whenever the useful information is treated as redundant data, it generates greater FPR, and when the redundant data is mistaken as a nonredundant data, it generates higher FNR. However, excessive filtering is done when the data compression score is high, and it leads to higher rate of loss of useful data. TDBF algorithm is suitable for single-reader architecture and does not work well in large-scale environment.

To overcome such kind of limitations, we have proposed and implemented the modified version of existing TDBF algorithm in this paper which is examined to outperform the existing algorithm.

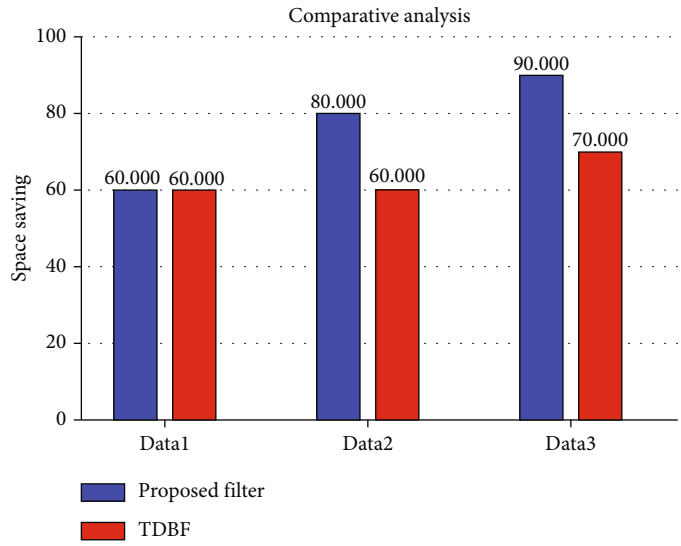

Figure 6: Comparison of 3 datasets in terms of space saving.

7. Proposed System Model

7.1. Data Generation. Let us consider a vehicular wireless sensor network where the tag identification of RFID devices has 5 fields which are

(i) ID (ID of RFID)

(ii) Time (total time)

(iii) RSS (Received Signal Strength)

(iv) loc (location of RFID)

(v) RSSI (Received Signal Strength Indicator)

7.2. Dynamic Environment Experiment. To set up the experiment in dynamic environment, we have considered some readings which are considered as "n." Here we have considered 10 different readings. So here, $n = 10$. To make these 10 readings random, we have used variable "n" that randomly takes "n" values from time, RSS, loc, and RSSI fields. Again to make it dynamic, the time, RSS, loc, and RSSI fields are chosen randomly from its corresponding matrices, e.g., time: l, RSS: r, loc: t, RSSI: i, and threshold: ω.

For the data generation of our proposed algorithm, we have considered a vehicular wireless sensor network where we have taken into consideration of "n" no. of vehicles randomly. The observations are taken on a random basis to check the effectiveness of our algorithm. As discussed earlier, every tag of RFID device is mapped through the hash function. After the hash function has been called, we consider the data generation part. As RFID comes with less no. of real dataset for our experiment, we have used synthetic dataset which is generated randomly by using Cooja Simulator. Now the dataset is ready by assigning alone values to tag IDs. All these values are also randomly developed to make it realise that reading time, RSS, loc, and RSSI are not the same always.

Table 4: Comparison of detection ratio of three datasets for TDBF and m-TDBF.

Algorithm	Data 1	Data 2	Data 3
m-TDBF	5.000	5.000	3.330
TDBF	5.000	2.000	2.000

Data generation file is executed 3 times to make it sure that reading is not repeating, i.e., reading changes as in the case of dynamic network.

Once the dataset is available, we then can generate two events which are

(i) Event (for initial hash mapping)

(ii) Event 1 (for comparing)

Data=tag=stores all tag data.

7.3. m-TDBF Algorithm. So now, we will process all IDs, and we are expecting it should return detected for IDs except last two. Initially, we calculate hash values or index for ID. Then, check corresponding index in m-TDBF matrix, third column (RSSI). If it is zero, then only we will go for time and RSS value checking. Else it is already detected so we assign RSS to it. No need to calculate hash. If it is 0, it is detected for first time and rest algorithms as per paper, i.e., it will calculate hash values and assign it to Event 1.

7.4. m-TDBF vs. TDBF. If we compare the existing TDBF algorithm with our proposed algorithm which is the modified version of TDBF algorithm (m-TDBF), then we can observe the following scenario

Condition 1

Vehicle is read when x.time == 0 that means RSS > ω.

Condition 2

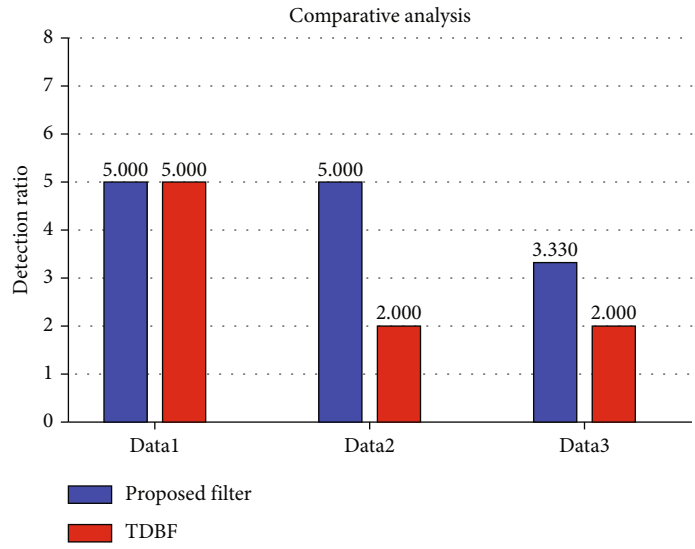

Figure 7: Comparison of 3 datasets in terms of detection ratio.

Next, vehicle will be read by the same tag where x.time $-$ TDBF(time) $> \tau$.

Condition 3

Repetitive reads were done by the same reader reading another vehicle.

In TDBF algorithm, the three above conditions are checked repetitively which enhances the time complexity factor of the algorithm and hence slows down the processor. But in m-TDBF algorithm, if RSSI is detected for the first time, then we can directly go to the third condition without wasting time on checking condition 1 and condition 2. But if the vehicle is already detected, then we can just update RSS which improves the time complexity factor. The same RSSI is repeated; hence, the above three conditions of TDBF will not proceed. This limitation of TDBF has been enhanced in m-TDBF which can be observed by the experimental results of the proposed algorithm.

8. Experimental Results and Performance Evaluation

8.1. Performance Measures. For our experiment, we have considered three datasets of our generated synthetic data to compare different parameters used for it. The performance measures which have been used for this experiment can be coined as follows:

8.2. Compression Ratio. It can be defined as the score of RFID data collected with respect to filtered tags. Error rate is minimized when we have higher compression rate. The comparison of compression ratio of two algorithms for three datasets data 1, data 2, and data 3 is given in Table 2. Equation (2) describes the mathematical formula of compression ratio in terms of data rate which is as follows:

$$\text{Compression ratio} = \frac{\text{uncompressed data rate}}{\text{compressed data rate}}. \quad (2)$$

Figure 5 describes the graphical representation of the comparative analysis of compression ratio.

8.3. Space Saving. Storage space of a Bloom filter relies on the average inaccurate rate. The space saving attribute yields different value for the EPC class of RFID reader. With the increase of false positive and false negative rate, the Bloom filter compresses the data nearly 10 times as compared to the original data. So in our proposed method, the warehouse model is efficient in space saving. Equation (3) formulates the space saving which is as follows:

$$\text{Space saving} = 1 - \left[\frac{\text{compressed size}}{\text{uncompressed size}}\right]. \quad (3)$$

Here we have made a comparative analysis of existing algorithm and proposed algorithm for three datasets which is provided in Table 3, and the bar chart is shown in Figure 6 graphically.

8.4. Detection Ratio. It can be defined as the score of false negative rates with respect to the false positive rate. In Table 4, we have given the comparison of two algorithms for 3 datasets and presented it graphically in Figure 7.

8.5. Time Complexity. The comparison of time difference between two algorithms is observed. Time is taken in seconds with respect to the number of tags. From 0 to 10 seconds, the graph is stagnant for both the algorithms. But after 15 seconds, the proposed algorithm shows a decreasing trend with respect to the number of tags which makes it more efficient than the other. For this experiment, we have used Impinj Revolution series of passive readers and several supporting tags.

8.6. Comparative Analysis. TDBF has always been best observed in closed room scenario as the tag is fluctuated by environment, but here in this paper, we have done the

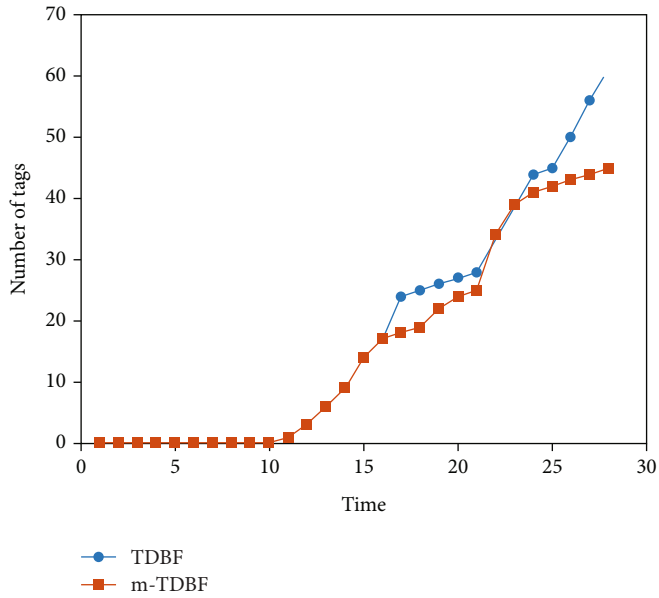

Figure 8: Comparison of time complexity between TDBF and m-TDBF.

experiment for the proposed algorithm in open air environment to taste the best fit practice. We have taken randomly selected 3 datasets, namely, data 1, data 2, and data 3. The tabular representation of three datasets is given below where we have compared TDBF and m-TDBF algorithms in terms of compression ratio, space saving, detection ratio, and time complexity. The obtained results of bar graphs shows a clear picture of consistency of the proposed algorithm. The graph of time complexity shows the efficiency of our proposed algorithm in terms of time management.

8.7. Results. From the above comparative analysis and graphical representation, it is clear that our proposed algorithm (m-TDBF) is more efficient than the existing TDBF in terms of compression ratio, space saving, detection ratio, and time complexity. So the m-TDBF algorithm maintains an increasing trend for all the performance measures as compared to TDBF. Hence, in both static and dynamic environment, m-TDBF can filter out more number of valid data enhancing the system requirements and limiting the loss of useful information. This comparative analysis can be graphically represented which is shown in Figure 8 below:

9. Conclusion and Future Work

In practical scenario the RFID data contains a huge variety of repetitive data which overburdens the system and results in slow processing. Existing data structure cannot enhance the performance due to overloading of huge amount of data streams coming to it. It is eventually difficult to process and filter the data resulting inefficiency in the system. So Bloom filter has come into picture which is considered to be a probabilistic data structure enhancing the system utility by managing the fruitful processing and filtering of huge amount of data streams in RFID. So in this paper, we have introduced the modified version of TDBF algorithm which is proven to be fit for enhancing system efficiency in large-scale application environment. The comparative study of TDBF algorithm with m-TDBF algorithm shows the results that the proposed algorithm is more efficient to reduce the error impact by introducing the RSSI. The fluctuations in RSS value can be handled by RSSI effectively without depending on the quality of tag or environment factors. Here we have compared the compression ratio, space saving, and detection ratio attributes as three important performance measures which gives a clear picture about the efficiency of the proposed algorithm. The proposed algorithm is fit for dynamic scenario in terms of effectively filtering the valid data into the system. To check the effectiveness of our proposed algorithm, by adding more number of parameters will be next step of our research.

Data Availability

The evaluation data used to support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] H. El Alami, A. Najid, H. El Alami, and A. Najid, "Optimization of energy efficiency in wireless sensor networks and internet of things: a review of related works," *Nature-Inspired Computing Applications in Advanced Communication Networks*, 2020.

[2] H. E. Alami and A. Najid, "EEA: clustering algorithm for energy-efficient adaptive in wireless sensor networks," *International Journal of Wireless Networks and Broadband Technologies (IJWNBT)*, vol. 7, no. 2, pp. 19–37, 2018.

[3] H. E. Alami and A. Najid, "(SET) smart energy management and throughput maximization: a new routing protocol for WSNs," in *Security Management in Mobile Cloud Computing*, pp. 1–28, IGI Global, 2017.

[4] J.-S. Lee and W.-L. Cheng, "Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication," *IEEE Sensors Journal*, vol. 12, no. 9, pp. 2891–2897, 2012.

[5] S. Wang, Z. Cao, Y. Zhang, W. Huang, and J. Jiang, "A temporal and spatial data redundancy processing algorithm for RFID surveillance data," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6937912, 12 pages, 2020.

[6] R. Angeles, "RFID technologies: supply-chain applications and implementation issues," *Information Systems Management*, vol. 22, no. 1, pp. 51–65, 2005.

[7] J.-P. Qian, X.-T. Yang, X.-M. Wu, L. Zhao, B.-L. Fan, and B. Xing, "A traceability system incorporating 2D barcode and RFID technology for wheat flour mills," *Computers and Electronics in Agriculture*, vol. 89, pp. 76–85, 2012.

[8] J. Yin, J. Yi, M. K. Law et al., "A system-on-chip EPC Gen-2 passive UHF RFID tag with embedded temperature sensor," *IEEE Journal of Solid-State Circuits*, vol. 45, pp. 2404–2420, 2010.

[9] E. Evizal, T. A. Rahman, and S. K. A. Rahim, "Active RFID technology for asset tracking and management system," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 11, no. 1, pp. 137–146, 2013.

[10] P. V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, and K. V. S. Rao, "Phase based spatial identification of UHF RFID tags," in *2010 IEEE International Conference on RFID (IEEE RFID 2010)*, pp. 102–109, Orlando, FL, USA, 2010.

[11] F. Thiesse and F. Michahelles, "An overview of EPC technology," *Sensor Review*, vol. 26, no. 2, pp. 101–105, 2006.

[12] J. Kim, S. Kumara, S.-T. Yee, and J. Tew, "Dynamic shipment planning in an automobile shipment yard using real-time radio frequency identification (RFID) information," in *IEEE International Conference on Automation Science and Engineering*, pp. 148–153, Edmonton, AB, Canada, 2005.

[13] B. Nath, F. Reynolds, and R. Want, "RFID technology and applications," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 22–24, 2006.

[14] H. Chen, W.-S. Ku, H. Wang, and M.-T. Sun, "Leveraging spatio-temporal redundancy for RFID data cleansing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, SIGMOD'10*, pp. 51–62, Indianapolis, Indiana, USA, 2010.

[15] D. Bleco and Y. Kotidis, "RFID data aggregation," in *Proceedings of the 3rd International Conference on GeoSensor Networks, GSN'09*, pp. 87–101, Oxford, UK, 2009.

[16] C. M. Roberts, "Radio frequency identification (RFID)," *Computers and Security*, vol. 25, no. 1, pp. 18–26, 2006.

[17] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith, "Design of an RFID-based battery-free programmable sensing platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 11, pp. 2608–2615, 2008.

[18] A. Juels, "RFID security and privacy: a research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[19] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Cryptographic Hardware and Embedded Systems-CHES 2004*, M. Joye and J.-J. Quisquater, Eds., pp. 357–370, Springer, Berlin, Heidelberg, 2004.

[20] H. Gonzalez, J. Han, X. Li, and D. Klabjan, "Warehousing and analyzing massive RFID data sets," in *22nd International Conference on Data Engineering (ICDE'06)*, p. 83, Atlanta, GA, USA, 2006.

[21] L. V. Massawe, J. D. M. Kinyua, and H. Vermaak, "Reducing false negative reads in RFID data streams using an adaptive sliding-window approach," *Sensors*, vol. 12, no. 4, pp. 4187–4212, 2012.

[22] K. Hu, L. Li, C. Hu, J. Xie, and Z. Lu, "A dynamic path data cleaning algorithm based on constraints for RFID data cleaning," in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 537–541, Xiamen, China, 2014.

[23] X. Wu, H. Liu, L. Zhang, M. J. Skibniewski, Q. Deng, and J. Teng, "A dynamic Bayesian network based approach to safety decision support in tunnel construction," *Reliability Engineering and System Safety*, vol. 134, pp. 157–168, 2015.

[24] B. Babagholami-Mohamadabadi, S. Yoon, and V. Pavlovic, "D-MFVI: distributed mean field variational inference using Bregman ADMM," 2015, https://arxiv.org/abs/1507.00824.

[25] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing Bloom filter: challenges, solutions, and comparisons," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1912–1949, 2019.

[26] Y. Yao, S. Xiong, H. Qi, Y. Liu, L. M. Tolbert, and Q. Cao, "Efficient histogram estimation for smart grid data processing with the loglog-Bloom-filter," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 199–208, 2015.

[27] Q. Xiao, Y. Zhou, and S. Chen, "Better with fewer bits: Improving the performance of cardinality estimation of large data streams," in *IEEE INFOCOM 2017- IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[28] H. Mahdin and J. Abawajy, "An approach for removing redundant data from RFID data streams," *Sensors*, vol. 11, no. 10, pp. 9863–9877, 2011.

[29] M. S. Mekala, R. Patan, S. H. Islam, D. Samanta, G. A. Mallah, and S. A. Chaudhry, "DAWM: cost-aware asset claim analysis approach on big data analytic computation model for cloud data centre," *Security and Communication Networks*, vol. 2021, Article ID 6688162, 16 pages, 2021.

[30] X. Wang, Y. Ji, and B. Zhao, "An Approximate Duplicate-Elimination in RFID Data Streams Based on d-Left Time Bloom Filter," in *Web Technologies and Applications*, L. Chen, Y. Jia, T. Sellis, and G. Liu, Eds., pp. 413–424, Springer International Publishing, Cham, 2014.

[31] W. Rui, L. Guoqiong, and D. Guoqiang, "Filtering redundant RFID data based on sliding windows," in *2014 International Conference on Management of e-Commerce and e-Government*, pp. 187–191, Shanghai, China, 2014.

[32] S. Ur Rehman, R. Liu, H. Zhang, G. Liang, Y. Fu, and A. Qayoom, "Localization of moving objects based on RFID tag array and laser ranging information," *Electronics*, vol. 8, no. 8, p. 887, 2019.

[33] D. Samanta, A. H. Alahmadi, M. P. Karthikeyan et al., "Cipher block chaining support vector machine for secured decentralized cloud enabled intelligent IoT architecture," *IEEE Access*, vol. 9, pp. 98013–98025.

[34] X. Li, "Collaborative localization with received-signal strength in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3807–3817, 2007.

[35] C. Metzger, S. Gershwin, and E. Fleisch, "The impact of false-negative reads on the performance of RFID-based shelf inventory control policies," *Computers and Operations Research*, vol. 40, no. 7, pp. 1864–1873, 2013.

[36] H. Ma, Y. Wang, and K. Wang, "Automatic detection of false positive RFID readings using machine learning algorithms," *Expert Systems with Applications: An International Journal*, vol. 91, pp. 442–451, 2018.

[37] E. Q. Shahra, T. R. Sheltami, and E. M. Shakshuki, "A comparative study of range-free and range-based localization protocols for wireless sensor network," *International Journal of Distributed Systems and Technologies*, vol. 8, no. 1, pp. 1–16, 2017.

[38] M. Maheswari, S. Geetha, S. S. Kumar, M. Karuppiah, D. Samanta, and Y. Park, "PEVRM: probabilistic evolution based version recommendation model for Mobile applications," *IEEE Access*, vol. 9, pp. 20819–20827, 2021.

[39] R. C. Costa and J. R. Sodré, "Compression ratio effects on an ethanol/gasoline fuelled engine performance," *Applied Thermal Engineering*, vol. 31, no. 2-3, pp. 278–283, 2011.

[40] M. Cafaro, M. Pulimeno, I. Epicoco, and G. Aloisio, "Parallel Space Saving on Multi- and Many-Core Processors," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 7, 2018.