Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 406316, 24 pages
http://dx.doi.org/10.1155/2013/406316

Review Article

Data Mining Techniques for Wireless Sensor Networks: A Survey

Azhar Mahmood, Ke Shi, Shaheen Khatoon, and Mi Xiao

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Ke Shi; keshi@mail.hust.edu.cn

Received 18 February 2013; Revised 15 June 2013; Accepted 20 June 2013

Academic Editor: Marimuthu Palaniswami

Recently, data management and processing for wireless sensor networks (WSNs) has become a topic of active research in several fields of computer science, such as the distributed systems, the database systems, and the data mining. The main aim of deploying the WSNs-based applications is to make the real-time decision which has been proved to be very challenging due to the highly resource-constrained computing, communicating capacities, and huge volume of fast-changed data generated by WSNs. This challenge motivates the research community to explore novel data mining techniques dealing with extracting knowledge from large continuous arriving data from WSNs. Traditional data mining techniques are not directly applicable to WSNs due to the nature of sensor data, their special characteristics, and limitations of the WSNs. This work provides an overview of how traditional data mining algorithms are revised and improved to achieve good performance in a wireless sensor network environment. A comprehensive survey of existing data mining techniques and their multilevel classification scheme is presented. The taxonomy together with the comparative tables can be used as a guideline to select a technique suitable for the application at hand. Based on the limitations of the existing technique, an adaptive data mining framework of WSNs for future research is proposed.

1. Introduction

Advances in wireless communication and microelectronic devices led to the development of low-power sensors and the deployment of large-scale sensor networks. With the capabilities of pervasive surveillance, sensor networks have attracted significant attention in many applications domains, such as habitat monitoring [1, 2], object tracking [3, 4], environment monitoring [5–7], military [8, 9], disaster management [10], as well as smart environments. In these applications, real-time and reliable monitoring is essential requirement. These applications yield huge volume of dynamic, geographically distributed and heterogeneous data. This raw data, if efficiently analyzed and transformed to usable information through data mining, can facilitate automated or human-induced tactical/strategic decision. Therefore, it is essential to develop techniques to mine the sensor data for patterns in order to make intelligent decisions promptly.

Recently, extracting knowledge from sensor data has received a great deal of attention by the data mining community. Different approaches focusing on clustering [11–14], association rules [15, 16], frequent patterns [17–20], sequential patterns [21–23], and classification [24–26] have been successfully used on sensor data. However, the design and deployment of sensor networks creates unique research challenges due to their large size (up to thousands of sensor nodes), random and hazardous deployment, lossy communicating environment, limited power supply, and high failure rate. These challenges make traditional mining techniques inapplicable because traditionally mining is centralized and computationally expensive, and it focuses on disk-resident transactional data. As a result, new algorithms have been created, and some of the data mining algorithms have been modified to handle the data generated from sensor networks. A plethora of knowledge discovery methodologies, techniques, and algorithms have been proposed during the last ten years.

For example, a decent amount of work is done for detection of the outlier in WSNs which is presented in [27–29]. Most of the techniques examined in [27, 28] heavily rely on data mining techniques, but their focus is detection of irregularities in WSNs data rather than information extraction and analysis. A survey [29] presented the anomaly

Table 1: Difference between traditional and sensor data processing.

	Traditional data	WSNs data
Processing architecture	Centralized	Distributed
Data type	Static	Dynamic
Memory usage	Unlimited	Restricted
Processing time	Unlimited	Restricted
Computational power	High	Weak
Energy	No constraints	Limited
Data flow	Stationary	Continuous
Data length	Bounded	Unbounded
Response time	Non-real-time	Real time
Update speed	Low	High
Number of passes	Multipass	Single

detection in multiple domains using data mining as well as statistical information theoretic and spectral techniques.

Since data mining is a broad discipline and can be applied to any domain data, more general surveys on data mining techniques can be found in [30], where authors examined the machine-learning and data mining techniques for analyzing medical data. Since the classification of data mining techniques in this survey is based on frequent pattern mining, clustering, and classification, there are plenty of surveys available on each of these techniques. For example, frequent pattern mining over data stream is presented in [31, 32]. A survey on clustering algorithm for WSNs is presented in [33, 34]. The clustering techniques examined in those papers exclusively focus on architecture and management of network rather than information discovery. A survey on classification methods over data stream is given in [35], where the author examined conventional classification techniques over data streams.

However, none of the above surveys examined data mining techniques that focus on information extraction and analysis from WSNs data. In comparison with the above-mentioned surveys, this paper examines algorithms and approaches specially designed for WSNs data, not only leading to a different classification, evaluation, and discussion on different domains but also presenting different choices of a solution. We examined how data mining algorithms will be utilized to make the sensor network applications intelligent. The research method consists of review of data mining techniques for WSNs such as frequent pattern mining, sequential pattern mining, clustering, and classification. Problem-based taxonomy is presented to classify and compare existing data mining techniques adopted for WSNs. In addition, evaluation of each technique is presented. Based on the limitations of existing techniques and special characteristics of WSNs, we proposed a new hybrid data mining architecture for WSNs, which combines the offline learning with distributive and online data processing.

The rest of the paper is organized as follows. After the introduction in Section 1, how traditional data mining process is different with data mining process in WSNs and challenges of data mining for WSNs data are discussed in Section 2. In Section 3, taxonomy of categorizing the existing data mining techniques for WSNs is presented. In Section 4, we analyzed a collection of published studies using the taxonomy framework. The comparison of data mining techniques for WSNs is presented in Section 5. The limitations of this work are given in Section 6, and future research directions are presented in Section 7. Finally, the paper ends with the conclusion in Section 8.

2. Fundamentals of Data Mining in WSNs

2.1. Data Mining Process in WSNs. Data mining in sensor networks is the process of extracting application-oriented models and patterns with acceptable accuracy from a continuous, rapid, and possibly nonended flow of data streams from sensor networks. In this case, whole data cannot be stored and must be processed immediately. Data mining algorithm has to be sufficiently fast to process high-speed arriving data. The conventional data mining algorithms are meant to handle the static data and use the multistep techniques and multiscan mining algorithms for analyzing static data-sets. Therefore, conventional data mining techniques are not suitable for handling the massive quantity, high dimensionality, and distributed nature of the data generated by the WSNs. Table 1 shows the summary of difference between traditional data and WSNs data mining process.

It can be observed from Table 1 that traditional data mining is centralized, computationally expensive, and focused on disk-resident transactional data. It directly collects data at the central site which is not bounded by computational resources. In comparison with traditional data-sets, the WSNs data flows continuously in systems with varying update rates. Due to huge amount and high storage cost, it is impossible to store the entire WSNs data or to scan through it multiple times. These characteristics of sensor data and the special design issues of sensor networks make traditional data mining techniques challenging. Hence, it is crucial to develop data mining technique that can analyze and process WSNs data in multidimensional, multilevel, single-pass, and online manner.

2.2. Challenges. According to the following reasons, conventional data mining techniques for handling sensor data in WSNs are challenging.

(i) *Resource Constraint.* The sensor nodes are resource constraints in terms of power, memory, communication bandwidth, and computational power. The main challenge faced by data mining techniques for WSNs is to satisfy the mining accuracy requirements while maintaining the resource consumption of WSNs to a minimum.

(ii) *Fast and Huge Data Arrival.* The inherent nature of WSNs data is its high speed. In many domains, data arrives faster than we are able to mine. Additionally, spatiotemporal embedding of sensor data plays an important role in WSNs application. This may cause many classical data processing techniques to perform poorly on spatiotemporal sensor data. The challenge for data mining techniques is how to cope with the continuous, rapid, and changing data streams and also how to incorporate user interaction during high-speed data arrival.

(iii) *Online Mining*. In WSNs, environment data is geographically distributed, inputs arrive continuously, and newer data items may change the results based on older data substantially. Most of data mining techniques that analyze data in an offline manner do not meet the requirement of handling distributed stream data. Thus, a challenge for data mining techniques is how to process distributed streaming data online.

(iv) *Modeling Changes of Mining Results Over Time*. When the data-generating phenomenon is changing over time, the extracted model at any time should be up-to-date. Due to the continuity of data streams, some researchers have pointed out that capturing the change of mining results is more important in this area than the mining results. The research issue is how to model this change in the results.

(v) *Data Transformation*. Since sensor nodes are limited in terms of bandwidth, transforming original data over the network is not feasible. Knowledge structure transformation is an important issue. After extracting model and patterns locally from WSNs data, the output is transferred to the base station. The challenge for data mining technique is how to efficiently represent data and discovered patterns over network for transmission.

(vi) *Dynamic Network Topology*. Sensor network deployed in potentially harsh, uncertain, heterogenic, and dynamic environments. Moreover, sensor nodes may move among different locations at any point over time. Such dynamicity and heterogeneity increase the complexity of designing an appropriate data mining technique for WSNs.

To address these challenges, researchers have modified the conventional data mining techniques and also proposed new data mining algorithms to handle the data generated from sensor networks. In the following section we have provided the taxonomy of these data mining techniques based on the discipline from which they adopt their ideas.

3. Taxonomy of Data Mining Techniques for WSNs

In this section, a classification scheme for existing approaches designed for mining WSNs data is presented. The highest-level classification is based upon the general data mining classes used such as *frequent pattern mining*, *sequential pattern mining, clustering,* and *classification*. Most of the *frequent pattern mining* and *sequential pattern mining* approaches have adapted the traditional frequent mining techniques such as the *Apriori* and *frequent pattern* (*FP*) growth-based algorithms to find the association among large WSNs data. Cluster-based approaches have adapted the *K-mean, hierarchical,* and *data correlation-based* clustering, based upon the distance among the datapoint, whereas, classification-based approaches have adapted the traditional classification techniques such as *decision tree, rule-based*, *nearest neighbor,* and *support vector machines* methods based on type of classification model that they used. These algorithms have very different and distinct roles; therefore, in order to choose the algorithm for WSNs application, one has to decide in term of these top-level classes.

The second level of classification is based upon each approach's ability to process data on centralized or distributed manner. Since WSNs nodes are limited in terms of resource such as power, computation, bandwidth, and memory, therefore, the approach meant for distributed processing requires one-pass algorithms to complete a part of data mining locally and then aggregate the results. The objective to use the distributed approaches is to limit the messages and communication energy of sensor nodes while transferring data to central server. It also helps to improve the WSNs lifetime and can extract maximum data from the environment, whereas, the centralized processing data from entire network is collected and stored at central server for analysis. Since the central server is rich in resources, therefore, there are no such constraints for choosing the accurate algorithm. This approach is always discouraged for the researchers because it generates huge amount of dataflow and communication which can create bottlenecks and wastage of communication bandwidth. These two data processing/storage architectures have a large impact on type of data mining algorithm to choose; therefore, one has to decide the processing\storage architecture for choosing the data mining algorithm for WSNs application.

The third level of classification is selected according to the attitude towards solving a specific problem. Research in WSNs area has focused on two separate aspects of issues, namely, WSNs performance issues and application issues. As WSNs nodes are usually resource constrained such as energy, communication bandwidth, memory, and resource, aware algorithms are needed to maximize the WSNs performance. On the other hand, a WSNs application requires data precision and accuracy, fault tolerance, event prediction, scalability, and robustness, and it often needs abundant use of energy, communication, and redundancies. This leads to resource tradeoff: whether someone sacrifices the application's performance in favor of network efficiency or wants to get the best application performance and deal with the network resource issues such as energy in some other way (larger battery; renewable sources with the nodes). For this reason, WSNs performances or application-specific-oriented approaches have been selected as the lowest-level classification criteria. The taxonomy of data mining techniques for WSNs is presented in Figure 1.

4. State of the Art of Data Mining Techniques for WSNs

In this section, data mining techniques designed for WSNs are classified using the taxonomy framework presented in Section 3, and the characteristics and performance analysis of each technique is discussed.

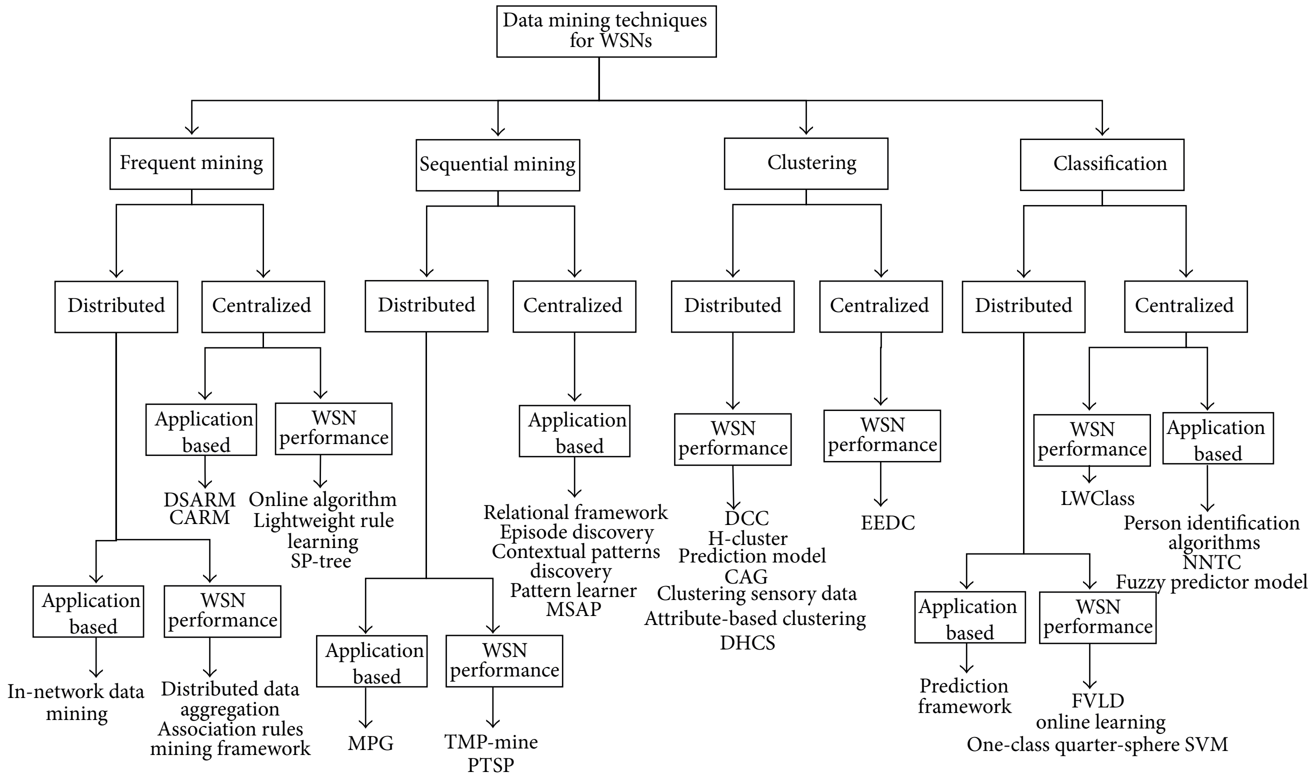


Figure 1: Taxonomy of data mining techniques for sensor networks.

4.1. Frequent Pattern Mining. In this section, we review some of the works that have been proposed for mining frequent patterns from WSNs data. Frequent pattern mining is used to find the group of variables that co-occur frequently in the data-set. The aim is to find the most interesting relations between variables. Traditional frequent pattern mining algorithms [36–39] are the CPU and the I/O intensive, making it very expensive to mine dynamic nature of WSN data. Unlike the mining static database, dynamic nature of WSNs data led to the study of online mining of frequent itemset. As a result, traditional frequent pattern mining algorithms are modified according to nature of WSNs data.

The basic frequent pattern mining technique is association rule mining technique. The first known association rule mining algorithm is *Apriori* [40]. It is based on level-wise candidate generation and test methodology by making several scans over database. In each iteration, the patterns found to be frequent are used to generate possible frequent patterns (the candidates) to be counted in the next iteration. Therefore, the *Apriori* technique finds the frequent patterns of length k from the set of already generated candidate patterns of length $k-1$. In the subsequent step, the association rules are generated by computing the support and confidence of each frequent item in given database D which is defined as follows:

$$\text{Support}\,(A) = \frac{\text{Sup}\,(A)}{D}, \tag{1}$$

where $\text{Sup}(A)$ is the number of occurrence of A in database D. Consider the following:

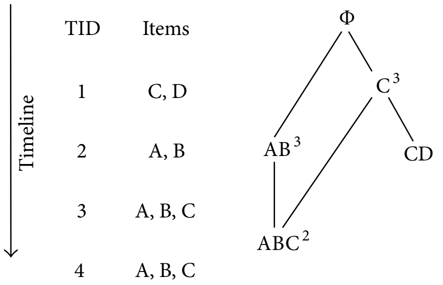
$$\text{Confidence}\,(A \longrightarrow B) = \frac{\text{Sup}\,(A \cup B)}{\text{Sup}\,(A)}. \tag{2}$$

This is impractical in the context of sensor networks as it implies that all data has to be stored somewhere. However, recently, there has been a growing amount of work on discovering frequent item-sets from a data *stream* of transactions such that every transaction is considered only once and can be deleted afterwards.

The other basic approach from mining association rule is FP-growth [41] which can discover frequent patterns by reducing the database scans by two and eliminating the requirement of candidate generation as compared with Apriori. With the first database scan, the algorithm finds the set of distinct items with respective support count (i.e., frequency) in the database. Then, with the second database, scan the algorithm summarizes the database in the form of a frequency-descending tree (i.e., the FP-tree). The complete set of frequent patterns is, then, mined from the FP-tree by recursively applying a divide-and-conquer-based pattern growth approach, called the FP-growth algorithm, without additional database scan. The highly compact FP-tree structure introduced a new wing of research in mining frequent patterns. However, the static nature of the FP-tree and two database scans still limit its applicability to frequent pattern mining over a WSNs data. Recently, several centralized and distributed solutions have been proposed with the aim to maximize the WSNs' performance and maximize the application-based performance by applying Apriori-like and FP-growth methods over WSNs data.

4.1.1. Centralized Approaches Aim to Solve WSNs' Application-Based Issues. Halatchev and Gruenwald [42] proposed a centralized methodology called data stream association rule mining (DSARM) to identify the missing sensor's readings. It uses the association rule mining algorithm to identify sensors that report the same data for a number of times in a sliding window called related sensors and then estimates the missing data from a sensor by using the data reported by its related sensors. Due to the stream nature of sensor data, applying an association mining algorithm such as Apriori directly to sensor data is not possible. This situation led the authors to propose the DSARM framework that adapts the Apriori algorithm to make it applicable to the data stream received from sensor nodes. This technique is evaluated by simulation experiments on real data collected by the Department of Transportation in Austin, TX, USA, to estimate missing value in related data streams. Performance evaluations were conducted to compare DSARM and alternative approaches. The results show that DSARM requires more memory space and takes longer to produce estimation than the considered alternative approaches; it achieves better accuracy of the estimated value than the alternative approaches do. However, there exist some *limitations* in DSARM. First, it is based on two frequent itemsets association rule mining, which means that it can discover the relationships only between two sensors and ignore the cases where missing values are related with multiple sensors. Second, it finds those relationships only when both sensors report the same value and ignores the cases where missing values can be estimated by the relationships between sensors that report different values.

Jiang and Gruenwald [43, 44] proposed a data estimation technique called CARM (closed item-sets-based association rule mining), which can derive the most recent association rules between the sensors in the current sliding window. The technique is based on the closed frequent item-sets mining algorithm of data streams called CFI-stream [45]. It maintains an in-memory data structure called direct update (DIU) tree to store closed item-sets. When a new transaction arrives, the algorithm checks each item-set in the transaction over a data stream sliding window online and incrementally updates the closed item-sets' support. If CRAM found some missing values in sensor reading, instead of generating all possible association rules, it generates the rules that have strong relationships with the current round of sensor readings where one or more readings are missing. Based on these rules and selected closed item-sets, CRAM generates the estimated values which contain item values that are not included in the original readings. Figure 2 redrawn from [43] shows the DIU tree after receiving first four transactions. It shows that currently there are four closed item-sets: C, AB, CD, and ABC in the DIU tree, and their associated supports at the right-upper corner are 3, 3, 1, and 2. A basic set of rules is generated from these frequent item-sets. All other rules can be inferred from this basic rule set.



TID	Items
1	C, D
2	A, B
3	A, B, C
4	A, B, C

Figure 2: Lexicographical-ordered direct update tree.

4.1.2. Centralized Approaches Aim to Maximize WSNs' Performance. Loo et al. [46] have proposed online one-pass algorithms for mining large sensor streams. They mine the frequent value set from sensor stream data by transforming the stream data into interval list (IL) under lossy counting framework [47]. The time is divided into equal-size interval and snapshot from the sensor reading is taken when there is an update on sensor reading. Sensors' value at that snapshot constructs the value sets stored in database. An Apriori-based strategy is used to mine the value sets. The analysis of IL-based presentation of stream data showed favorable results using synthetic data-set. However, while computing the IL of candidate value set, redundant intersection of IL is inevitable, which affects the performance in terms of time and computation cost. The proposed technique is evaluated by comparing the performance of ILB against an application of lossy counting (LC) using a weighted transformation method on synthetic dataset. According to their experiments, ILB outperforms LC significantly for large sensor networks. Moreover, both the processing time and memory consumption of ILB are more stable than those of LC.

Chong et al., [48] proposed a rule-learning model that finds strong rules from sensor readings. The rules are used as a trigger to control sensor network operations; for example, they can be used to sleep sensor or reduce data transmission to conserve energy. To mine the rules, *Apriori* is modified to count the number of transactions that are frequent instead of the item-sets within transactions, and transactions are processed in batches $b_1, b_2, \ldots, b_X$. Suppose, there is node M that collects light, temperature, and microphone reading from three other sensor streams S_0, S_1, and S_2. Initially, M is queried to collect all sensory values, it is used to generate a rule of the form of a_n which implies a_{n-1}; therefore, the rule is extracted and only a_n is sent to the base station. Upon receiving the reading a_n and utilizing knowledge of the rule, the reading of a_{n-1} can be inferred. All extracted rules are stored in rule repository. The proposed method is validated by using simulation implemented in C language on synthetic dataset. In the experiment, the first correlated data received from sensor is used to extract rules. For subsequent phase, these rules are used to infer reading of sensor for the next round.

Tanbeer et al. [49] proposed a tree-based data structure called sensor pattern tree (SP-tree) to generate association rules from WSNs data with one database scan. The main idea of the proposed approach is to obtain the frequency of all event-detecting sensors' data, construct a prefix-tree based on that in any canonical order, and then reorganize the tree in a frequency descending order. Through the reorganization, the SP-tree can maintain the frequently event-detecting sensors' nodes at the upper part of the tree, which in turn provides high compactness in the tree structure. Once the SP-tree is constructed, FP-growth mining technique is applied to find the frequent event-detecting sensor sets. Experiments are performed to verify the improvement in memory consumption and runtime that SP-tree achieves over PLT [50]. The experiments show that SP-tree outperforms PLT in time and memory consumption. The reason of such gain is two folds: first, the PLT construction requires two database scans, while SP-tree constructs the tree by scanning the database only once; second, the mining phase of SP-tree is highly efficient due to the frequency-descending tree structure.

4.1.3. Distributed Approaches Aim to Solve WSNs' Application-Based Issues. Romer [51] proposed an in-network data mining technique to discover frequent patterns of events with certain spatial and temporal properties. In this approach, user specifies the upper bound *maxscope* and *maxhistory* (variable to be measured in seconds) for the patterns of interest. The sensor collects these events and applies a mining algorithm to discover the pattern that satisfies the given parameters. Each node in the network collects the events from its neighbors within the maximum scope and keeps a history of their events for duration of the maximum history. After that, each node applies a mining algorithm to discover the local frequent patterns. The resulting frequent patterns are converted to association rules that describe an event of type E that occurs at node n with support S and confidence C. Local patterns are sent to the sink where secondary mining is performed to compute the global picture of entire network. The algorithm is implemented on BT node (bluetooth radio) platform [52], and the tradeoff between scope of the query and resource consumption on real dataset is evaluated. Results show, by reducing the scope of the query, that the proposed approach could decrease resource consumption. Major issues in this approach are memory consumption of itemset discovery algorithms and the communication overhead of event collection.

4.1.4. Distributed Approaches Aim to Maximize WSNs' Performance. Boukerche and Samarah [15] presented a distributed data extraction methodology to aggregate the data on sensor node which reduced the number of messages during transmission. The distributed solution sends some parameters such as support*, time-slot size, and historic period* from sink to all nodes within network. Each sensor node has its own buffer entry to set the support value. After each time slot, nodes check whether there are messages received during this time slot; if yes, then that node will set its buffer entry. When the historic period ended, each node will traverse its buffer; if the number of set value is more than or equal to support value provided initially, then the message would be transfered to sink. To evaluate the validity of the distributed approach, it is compared with the centralized methodology on real dataset. They conducted two experiments using historical periods of 5 and 10 days with minimum support values ranging from 10% to 90% and a time-slot size equal to 30 seconds. All of the reported results show a reduction in the number of messages and the data size while increasing in the support values. Major issues in this methodology are increase in cost for node buffer and also delay in crucial messages in case of high support value.

Boukerche and Samarah [50] proposed the positional lexicographic tree (PLT) structure for mining association rules in which the event-detecting sensors are the main objects of the rules regardless of their values. Similar to the FP-growth approach, PLT follows a pattern growth mining technique. The mining begins with the sensor having the maximum rank by generating the frequent patterns from its PLT in a recursive way. The computation is required at each recursion to update the PLT involved in the prefix part of a pattern. Therefore, two database scans requirement and the additional PLT update operations during mining limit the efficient use of this approach in handling WSNs data. The performance evaluation is done by comparing the PLT structure with the FP-growth algorithm. According to their results, PLT structure outperforms FP-growth in terms of CPU time and memory usage for all of the support values used; the enhanced performance using PLT when compared with FP-growth ranges from 30 percent to 50 percent.

4.2. Sequential Pattern Mining (SPM). Frequent pattern mining has been extended to find more complex structure such as sequential pattern mining. It discovers frequent subsequences as patterns in a sequence database. A sequence database stores a number of records, where all records are sequences of ordered events, with or without concrete notions of time. A large number of real-world domains such as user profiling, medicine, local weather forecast, and bioinformatics show an inherent tendency to be modeled by means of sequences of events/objects related to each other. This great variety of applications of sequential pattern mining makes this problem one of the central topics in WSNs data mining as shown by the research efforts produced in the recent years. The sequential pattern mining techniques in sensor network based on either traditional sequential mining algorithms such as Apriori-like algorithm [53], Apriori-based methods: *GSP* [54] PSP [55], and pattern growth approaches: *FreeSpan* and *PrefixSpan* [56, 57] or some new algorithm are devised specifically to work with sensor network environment.

4.2.1. Centralized Approaches Aim to Solve WSNs' Application-Based Issues. Esposito et al. [58, 59] presented a multi-dimensional relational sequence mining framework to identify the hidden frequent temporal correlations between sensor nodes. The algorithm is based on generic level-wise search method called APRIORI [60] for discovering correlated sensors. The framework exploits the relational language to describe the temporal evolution of a sensor network along with contextual information by working in two phases. Firstly, an abstraction step is to segment and label the real-valued time series into similar subsequences by using a kernel density estimator approach. Then, the knowledge is enriched by adding interval-based operators between the subsequences obtained in the discretization step, and the relation pattern mining algorithm has been extended in order to deal with these new operators. By taking into account the interval-based temporal data along with contextual information about events, it discovers interesting and more human-readable patterns. The framework is evaluated on real dataset collected from a wireless sensor network made up of 54 Mica2Dot [61] sensors deployed in the Intel Berkeley Research Lab [62]. Each sensor collected topology information, along with humidity, temperature, light, and voltage values once every 31 seconds. Results show the strong correlation among some measurements, which is useful for anomaly detection.

Cook et al. [21] present MavHome smart home architecture which focuses on the creation of an intelligent home, perceiving the state of the home through sensors and acting upon the environment through device controllers. An important characteristic of the proposed architecture is the ability to make decisions based on predicted activities. To predict the activities, an algorithm called episode discovery (ED) is proposed, which is based on the work of Srikant and Agrawal [54] for mining sequential patterns from time-ordered transactions. Values that can be predicted include the usage pattern of devices in the home, the movement patterns of the inhabitants, and the typical activities of the inhabitants. They utilize prediction algorithms on action sequences stored in inhabitant event history to forecast user actions. Actions can then be automated based on the significance of mined patterns as well as the predictive accuracy of the next event. A key disadvantage is the fact that the entire action history must be stored and processed off line, which is not practical for large prediction tasks over a long period of time. Cook et al. demonstrated the effectiveness of MavHome on synthetic smart home data and real data collected by students using X10 controllers in their homes. Experiments show a predictive accuracy as high as 53.4% on the real data and 94.4% on the synthetic data.

Rabatel et al. [22] presented a strategy to detect anomalies from sensor data to improve the railway maintenance. They extract sequential pattern from real railway data and identify the abnormal behavior. Based on these abnormal findings, alarms are automatically triggered to notify potential failures. This abnormal behavior depends on environmental (weather conditions, travel characteristics) and structural (route, episode index in the route) changes in data. The *PSP* [55] algorithm has been used to identify the sequential patterns. To tackle the environments conditions, a contextual knowledge-based method is proposed, which is able to provide information on the seriousness and possible causes of a deviation. The proposed technique helps in proactive maintenance of train. However, real-time context can be improved by providing precise and exact information for anomaly detection.

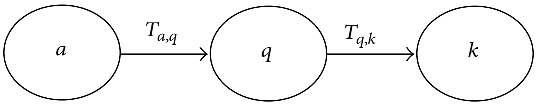


FIGURE 3: Example of sequential alarm pattern.

Guralnik and Haigh [23] use sequential pattern mining to learn typical behaviors of humans in their homes. Human behavior is inferred by using motion sensors, pressure pads, door latch sensors, and toilet flush sensors. They installed 10–20 sensors of different types in a home and built models of what sensor firings correspond to what activities, in what order, and at what time. For example, "In 60% of the days, the Kitchen-Motion sensor fires between 18h00 and 18h30, and then the Living-Room-Motion sensor fires between 18h20 and 20h00, and then the Bedroom-Motion sensor fires between 19h45 and 22h00." Their algorithm uses these data to learn the sequences of rooms in which the person was acting, and it uses domain knowledge to extract the sequences of rooms the person was acting in. These sequences are then analyzed by a human expert to identify complex behavior models. These models can be used to select the appropriate response plan to the action of elderly.

Wu et al. [63] proposed a new algorithm for mining sequential alarm patterns (MSAPs) from the alarm data generated by GSM system. Sequential events are identified from alarm data by defining time interval between adjacent events. For example, if time is set as six hours, then the sequential alarm pattern (a, b, c) indicates that a, b, and c happen in order and that the time interval between a and b and between b and c is less than six hours. An example of sequential alarm sequence redrawn from [63] is shown in Figure 3.

The number in circle represents the *error ID,* and $T_{a,q}$ denotes the time difference between alarm event a and alarm event q. The knowledge extracted is not only useful for identifying relevance between two events, but it is also predict the alarm sequence and takes proper steps to prevent the occurrence of the alarms if at all possible. For example, if the network operator detects that, the alarm a occurring at time t operator should dissipate this alarm before the time $t + T_{a,q}$ to alleviate the abnormal situations incurred. The limitation in this technique is that it cannot discover other possible time-interval patterns between the events.

It is observed that there is none of centralized solutions which aim to maximize the WSNs' performance.

4.2.2. Distributed Approaches Aim to Solve WSNs' Application-Based Issues. Tseng and Lu [64] proposed an object tracking strategy named the multilevel object tracking (MLOT) to discover sequential patterns in object tracking sensor networks (OTSNs) by mining the movement log in sensor networks. A multilevel hierarchical structure is adapted by using the clustering mechanism that represents the hierarchical relations among sensor nodes to achieve the goal of keeping track of moving objects in a real-time manner. The movement logs of the moving objects are analyzed by developing the data mining algorithm movement pattern generation (MPG) to obtain the movement patterns, which are then used to predict the next position of a moving object and to activate the least sensor node. The MPG is based on Apriori which uses the frequency of the inference pattern to evaluate the confidence of the pattern and which with the highest frequency serves as the basis of the prediction.

4.2.3. Distributed Approaches Aim to Maximize WSNs' Performance. Tseng and Lin [65] proposed an object tracking strategy named TMP-mine to discover sequential patterns in object tracking sensor networks (OTSNs) by mining the temporal movement patterns (TMPs) logs. The discovered temporal movement rules (TMRs) are used to predict the location of next objects for saving energy. In the proposed model object is able to record the sensor nodes it visited along with the arrival time at each node. The movement log is collected by equipping the sensor nodes with storage devices. The WSN collects and integrates the movement log of moving objects. The integrated movement log is used as the input to the data mining method named the TMP-miner which uses the pattern growth approach for discovering the TMPs. By applying the TMP-mine algorithm, the TMPs are discovered, and then the temporal movement rules (TMRs) are generated for predicting next location of moving object. Suppose that the following two rules are discovered by vehicle tracking system:

Rule 1. (Station A → interval 10 min → Station B → interval 5 min → Station C).

Rule 2. (Station A → interval 20 min → Station B → interval 5 min → Station → D).

By dispatching these rules to the corresponding sensor nodes, the tracking can be made in energy-efficient way. For example, if a car moves with the pattern as (Station A → interval 10 min → Station B → interval 5 min) that matches with Rule 1, then the node in Station B has only to activate the node in Station C rather than that in Station D or those around Station B.

Samarah et al. [66] proposed an energy-efficient prediction-based tracking technique by using the sequential patterns (PTSPs). This technique helps to predict the future location of a moving object with the minimum number of sensor nodes while keeping the other sensor nodes in the network in sleep mode. The PTSP is based on the inherited patterns of the objects movements in the network and the utilization of sequential patterns to predict in which sensor node the moving object will be heading next.

4.3. Clustering. Clustering is unsupervised learning, where given data is categorized into subsets so that each subset represents a cluster which has distinctive properties. It has been considered a useful technique especially for applications that require scalability to large number of sensor nodes. Clustering also supports aggregation of data in order to summarize the overall transmitted data.

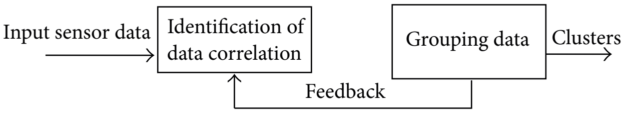


Figure 4: Data clustering for sensor networks.

In the current literatures, problems related to clustering are addressed by node clustering or data clustering. Recently, large numbers of node clustering algorithms have been designed for WSNs [67–83]. These clustering techniques widely vary in their objectives depending on the node deployment and bootstrapping schemes, the pursued network architecture, the characteristics of the cluster head (CH), and the network operation model. Although node clustering may be related to data clustering, for example, considering data similarity of neighboring node, many popular node clustering algorithms that partition the sensor nodes into a number of small groups and elect a cluster head for every group do not use the data mining techniques directly. In this study, we only focus on data clustering techniques to efficient data mining and find data correlations among the nodes. Figure 4 shows the commonly used data clustering in data mining process.

This work adapted the *K-mean, hierarchical,* and *data correlation-based methods*. The k-mean algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high, but the intercluster similarity is low. Cluster similarity is measured with respect to the mean value of the objects in a cluster. Hierarchical method creates a hierarchical decomposition of the given set of data objects. It works by grouping data objects into a tree of clusters, whereas, data correlation-based clustering forms clusters based on spatial and temporal correlations with similar node sensory values within a given threshold, and these clusters remain fixed until the sensory value threshold has changed over time. When the threshold values change, the related sensor nodes will then communicate with neighboring nodes associated with other clusters to change their cluster memberships. The drawback of this type of clustering is that it does not consider node residual energy. It is observed from the survey that the centralized and distributed clustering solutions are aim to maximize the WSNs performance.

4.3.1. Centralized Approaches Aim to Maximize WSNs' Performance. Liu et al. [84] proposed a centralized graph-based energy-efficient data collection (EEDC). EEDC is on-demand clustering algorithm that clusters node into groups such that members have similar sensor readings, and thus the protocol clusters the network with an awareness of the phenomena being sensed. EEDC is a centralized approach where the sink compares data from different nodes with a user-defined dissimilarity measure. EEDC models the cluster creation process as a clique-covering problem by constructing a graph G such that each sensor node is a vertex in the graph. An edge (u, v) is drawn if the dissimilarity measure between vertex u and vertex v is less than or equal to the given intracluster dissimilarity measure threshold *max_dst*. A cluster is a clique in the graph, and the clustering problem uses the minimum number of cliques to cover all vertices in the graph. This process minimizes the number of clusters and maximizes the energy saving. The sink also dynamically adjusts the clusters based on spatial correlation and the received data from the sensors. The algorithm produces robust and well-balanced clusters. However, due to centralized processings it is not suitable for large-scale WSNs.

4.3.2. Distributed Approaches Aim to Maximize WSNs' Performance. Guo et al. [85] proposed the H-cluster, a distributed algorithm to cluster sensory data. The input of this algorithm is the set of sensory data collected by all of the sensors from the time WSN starts working up to the current time. The output of the algorithm is a set of cluster features that summarize the clusters of the input sensory data-set. Hilbert-Map mapping algorithm has been used to map a d-dimensional sensory data space into a 2-dimensional area covered by a given WSN. H-cluster has 2 phases: (1) it merges connected grid features with local cluster features of (sensory dimensional) D at each destination node; (2) it combines the connected local clusters to global clusters. The experiments on the centralized and distributed data are carried out to compare the H-Cluster with C-Corner and C-Center algorithms. During experiment, four types of environment attributes are sensed by the sensors, which are temperature, humidity, light, and voltage. The results show that H-Cluster algorithm is much efficient in data loss, energy, and the quality of cluster data in small WSN. The results also shows that as the amount of sensory data delivered increases the amount of data loss also increases and energy efficiency decreases by increasing the size of WSNs.

Yeo et al. [86] proposed data correlation-based clustering scheme (DCC) based on similarity of sensor data along a spatial suppression scheme which helps to reduce the data size. DCC enhances the advertisement phase of HEED [71] in which cluster heads are selected according to probability of becoming a cluster head; during this phase, sensor nodes communicate with each other, and the resulting clusters are organized by sensor nodes which have similar readings. Spatial suppression is performed on cluster head, and it also computes the difference between sensor reading and representative value. If a cluster head has redundant data, it will remove it except for the node identification. The experimental results justify the hypothesis claim that the clustering based on data correlation has better compression performance than ordinary clustering based on locality of communication, they show that DCC reduces 40% of data size through suppression and prolongs network lifetime 20%–30%. However, for the large-scale network applications (nodes > 500), DCC is inefficient because each cluster head needs more energy to collect similar data readings and also to communicate with several nodes. Also in case of low percentage of similar data reading, DCC is ineffective due to higher rate of cluster head creation.

Beyens et al. [87] proposed a cluster-based architecture for wireless sensor networks in which cluster heads spatiotemporally correlate and predict the measurements of the cluster members by executing their prediction model. In their approach, the cluster heads execute a prediction model, while gateway nodes at the circumference of the clusters are responsible for the routing task. Prediction model is used to select a suitable node of the cluster to be activated. The idea is to put a sensor node to sleep when there are no objects in its sensing region.

Yoon and Shahabi [88] present the clustered aggregation (CAG) algorithm that forms clusters of nodes sensing similar values within a given threshold (spatial correlation), and these clusters remain unchanged as long as the sensor values stay within a threshold over time (temporal correlation). By grouping nodes on similar values, CAG only transmits one reading per group. When the threshold values change, the related sensor nodes will then communicate with neighboring nodes associated with other clusters to change their cluster memberships. CAG guarantees the result to be within a user-specified error-tolerance threshold. Cluster formation is performed while queries are disseminated to the network (query phase), where clusters group nodes sensing similar values. Subsequently, CAG enters the response phase wherein only one aggregated value per cluster is transmitted up the aggregation tree. CAG is a lossy clustering algorithm (most sensory readings are never reported) which trades a lower result precision for a significant energy, storage, computation, and communication saving.

Taherkordi et al. [67] proposed a communication-efficient distributed protocol for clustering sensory data. A distributed version of K-Mean clustering algorithm is proposed and sends summarized data towards sink which reduces the communication transmission, time, and power consumption of sensor nodes. The sensor network is divided into clusters and cluster head node will only communicate with sink. Initially, base station transmits current center locations to cluster heads. Cluster head collects data from its sensor node and sends it to the base station including count and vector sum of its local sensory data points as well as sum of the squared distance from each local point to its center. On receiving data from CH, the base station updates the cluster mean, and the algorithm repeats until the function convergence is met. The efficiency of the algorithm is evaluated via simulations. Several programs are run to get the average number of transmissions over the network during each test. According to results, the communication cost is independent of the number of sensors (N) and increases linearly by increasing the number of centers. Major issues are extra memory for cluster head and computation power for summarization of data before transmitting to sink. Also the algorithm requires multiple rounds of message passing between cluster heads and the base station; this may have a serious effect on communication efficiency when the number of sensors is relatively high.

Wang et al. [89] promoted the idea of clustering the WSNs based on the queries and attributes of the data. The main motive is to achieve efficient dissemination of data in the network. The concept resembles the data-centric design model of WSNs. The clustering is established by mapping a hierarchy of data attributes to the network topology. The base station starts the clustering process by asking nodes

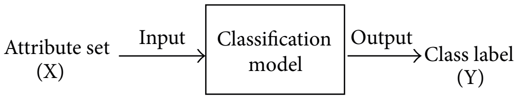


Figure 5: Classification maps input attribute set (X) to class label (Y).

to form clusters. Those nodes that hear the request decide whether they should nominate themselves as CHs based on their energy. After receiving the base-station request, sensor nodes having intention to become CHs wait for a random time period that is based on the remaining battery supply. If a node nominates itself, then it broadcasts an announcement to all nodes. A node joins the CH that it can reach over the least number of hops. Upon hearing a CH announcement from a node whose attribute is different, the recipient node establishes a new cluster for that attribute and becomes a CH. To evaluate the attribute-based clustering scheme, the authors have provided the theoretical analysis of it with flooding-based schemes. Analysis shows its attribute-based clustering scheme yield that gains over flooding-based schemes when there are subregions in the sensor network that are more targeted than others, that is, when the distribution of inquiries is not uniformly distributed over time and space.

Ma et al. [90] the proposed distributed, hierarchical clustering and Summarization algorithm (DHCS) for online data analysis and mining in sensor networks. The proposed method clusters sensor nodes based on their current data values as well as their geographical proximity, and it computes a summary for each cluster. The algorithm adopts several techniques, such as *difference* and *hop count* thresholds, to model node, and distance-based clustering. Initially, each node treats itself as an active cluster. Then, similar adjacent clusters are merged into larger clusters round by round. In each round, each cluster will try to combine with its most similar adjacent cluster simultaneously. Two clusters can be merged only if both consider one another as the most similar neighbor. DHCS terminates when no merging happens any more. The final clusters, which cannot be merged any more, are called steady clusters.

4.4. Classification. Classification is a task of assigning new object into a class of predefined object categories. Classification model is learned using the set of training data and classifies new data into one of the learned class. Figure 5 shows that classification maps input attribute set (X) to class label (Y).

Classification-based approaches have adapted the traditional classification techniques such as *decision tree-based*, *rule-based*, *nearest neighbor-based,* and *support vector machines-based* techniques based on type of the classification model that they used. Decision tree is a classifier in the form of tree and classifies the instance by starting at the root of tree and moving through it until a leaf node where class label is assigned. The internal nodes are used to partition data into subsets by applying test condition to separate instances that have different characteristics. Nearest neighbor-based approaches classify dataset based on closet training examples. The training examples are vectors in a multidimensional feature space with corresponding class labels. A nearest neighbor classifier is a lazy learner that does not process patterns during training [91]. To respond, a request to classify a query vector is made to locate the closest training vectors according to the distance metric. The classes of these training vectors are used to assign a class to the query vector.

Rule-based classifier groups the dataset in predefined classes by using "if. . .then. . ." rules of following form:

> (Condition) → Y: condition is a conjunction of attribute, and Y is a class label.

SVM (support vector machine) techniques partition the data belonging to different classes by fitting a hyperplane between them which maximizes the partition. The data is mapped into a higher-dimensional feature space where it can be easily partitioned by a hyperplane. Furthermore, a kernel function is used to approximate the dot products between the mapped vectors in the feature space to find the hyperplane.

4.4.1. Centralized Approaches Aim to Solve WSNs' Application-Based Issues. Chikhaoui et al. [92] proposed the decision Tree (DT-) based classification technique for sensor data. They applied the classification model to identify the persons in ubiquitous environment. In order to identify persons, the proposed approach first extracts frequent patterns called episodes from the datasets using the Apriori algorithm [53]. The next step evaluates the extracted patterns and assigns weights to these episodes to construct frequent episode weight matrix (FEWM).

Finally, the classification algorithm Decision tree (DT) is applied on FEWM. DT builds pattern classifier from a labeled training data-set using a divide-and-conquer approach. To build up a DT model, it recursively selects the attribute that is used to partition the training data-set into subsets until each leaf node in the tree has uniform class membership. The proposed approach is validated by experiment using data collected from the Domus Laboratory [93] and the Testbed smart home [94]. The general performance and classification accuracy of algorithm are evaluated by using the Weka framework version 3.7.0 [95]. Experiment results show good classification. However, using frequent episodes alone without temporal constraints and deep analysis does not guarantee good identification.

Sharma et al. [96] proposed a methodology for classifying the sensors data by using nearest neighbor trajectory classification (NNTC). The training phase simply stores every training example with its label. To make a prediction for a test example, first, its distance to every training example is computed. Then, k closest training examples are stored, where k is a fixed integer and $k \geq 1$; among the k examples, it looks for the label that is most frequent. This label is the prediction for this test example. The algorithm is evaluated by building a classifier from the preprocessed training data generated from NS2 [97] and test trajectory data [98] using class labels. Experimental investigation yields a significant output in terms of the correctly classified success rate, 92.3%.

Akhlaghinia et al. [99] proposed the prediction technique in smart home environments to predict the behavior pattern of occupants. The sensor NWs collect the variety of attributes including environmental changes and occupant's interaction with the environment. The collected data is then used by the learning approach to construct a classification-based predictive model to predict the ambient intelligence environment occupancy. The occupancy is predicted by using the fuzzy rules which are modeled by using the past value of time series data. In the learning process, input from the sensor is compared with stored rules to take appropriate action. The prediction-based approach improves the energy saving in smart homes and enhances the safety and security of occupants. The result shows the ability of the proposed technique to predict the combined occupancy time series. However, the model is implemented in single-user environment and unable to predict the complex environmental patterns in multi-user environment over long period.

4.4.2. Centralized Approaches Aim to Maximize WSNs' Performance. Gaber et al. [100] proposed the lightweight classification (LWClass), a one-pass algorithm for on-board mining of data streams in sensor networks. They used the algorithm output granularity (AOG) [101, 102] technique to preserve the limited memory size and change the algorithm output rate according to data rate, available memory, algorithm output rate history, and time constraints to fill the available memory with generated knowledge. The algorithm works by searching for the nearest instance stored in main memory when a new element arrives. All instances are already stored in the main memory according to a prespecified distance threshold. The threshold here represents the similarity measure acceptable by the algorithm to consider two or more elements as one element according to the elements attribute values. If the algorithm finds this element, then it checks the class label. If the class label is the same, then it increases the weight for this instance by one; otherwise, it decrements the weight by one. If the weight becomes zero, then this element is released from the memory. The algorithm is empirically validated using synthetic streaming data under the resource-constrained environment of a common handheld computer.

4.4.3. Distributed Approaches Aim to Solve WSNs' Application-Based Issues. McConnell and Skillicorn [103] presented a distributed framework for building and deploying predictors in sensor networks. By using the computational power of each sensor, a powerful learning structure on whole network is constructed. A distributed voting approach is proposed in which each sensor is a leaf of tree (DT) to perform local prediction. Instead of sending the raw data, the local predictive models built on sensors transmit the target class to the sink. At sink, the local predication models are combined to construct global prediction model. It shows how the local model enables sensors to respond to the change in target by relearning local models. The proposed framework is useful especially for sensor networks with limited energy, computation, and bandwidth resources. It makes efficient the distributed data mining in the presence of moving class boundaries. Data is also confidentially achieved by transmitting a predictive model instead of original data to the sink. The distributed prediction model is evaluated using J48 decision tree (implemented in WEKA) on variety of dataset for both simple and weighted voting schemes. According to results, distributed prediction model has the potential of an increase in accuracy combined with a reduction in model size and runtime as compared with a centralized approach. Major issues in this framework are the need of an expensive CPU on each sensor node for computing and building local predictive model, and also extra memory is required to store local predictive model.

4.4.4. Distributed Approaches Aim to Maximize WSNs' Performance. Malhotra et al. [104] proposed a distributed classification scheme to generate effective feature vectors of low dimension (FVLD) for wireless audio network. A distributed cluster-based algorithm for detection and classification of vehicles has been proposed. Sensors form clusters on-demand for the sake of running a classification task based on the produced feature vectors. The monitoring area is divided into clusters, and a cluster head is selected for each cluster. All sensors send their feature vector to cluster heads. The cluster head combines all received feature vectors (including one from itself), executes the classification task using, for example, KNN or ML classifiers, and makes decision on the class of the unknown vehicle. Two approaches were proposed: the first combines extracted features and the second combines individual decisions. Classification using decision fusion and a maximum likelihood (ML) classifier led to the best results. ML is also compared with KNN classifier with various settings of data and decision fusion schemes. The proposed technique produced the best classification accuracy of 89.46% as compared with all other approaches.

Flouri et al. [105–107] have proposed distributed and incremental techniques for learning classification rules using SVM-based (support vector machine) technique in a sensor network. The authors proposed two distributed algorithms: the distributed fix partition SVM (DFP-SVM) and the weighted distributed fix partition SVM (WDFP-SVM) for training a SVM applied to the classification problem in a WSN. SVM is incrementally trained on example set called support vector. The fact with SVM is that the number of support vectors is very small compared with the number of all sample values. Besides, the support vectors (and offset) reveal compressed representation of separating SVM hyperplane. That is why sending only the support vectors instead of all training samples to the next cluster head is obviously very energy efficient due to communication reduction. After training, the required parameters of the kernel functions are transferred to each node for classification. The performance of the proposed approach is evaluated by running number of simulation, and comparison is made with centralized algorithm. The results show that energy consumption decreases when the SVM is trained incrementally as compared with the centralized case. However, the challenges for SVM formulations are computational complexity and the choice of proper kernel function.

Rajasegarar et al. [108] proposed the SVM-based technique for outlier detection in sensor data. This technique uses one-class quarter-sphere SVM to identify local outliers at each node and to minimize the computational complexity. The sensor data that lies outside the quarter sphere is considered as an outlier. Each node communicates only the radius information of sphere with its parent for outlier classification. This technique identifies outliers from the data measurements collected after a long-time window and is not performed in real time. The technique also ignores spatial correlation of neighboring nodes, which makes the results of local outliers inaccurate. The technique is evaluated by using the real sensor measurement collected from deployment of wireless sensors in the Great Duck Island Project [2] for monitoring the habitat of sea birds. The algorithm is implemented in Matlab and two simulations were run to measure the computational strategy and various kernel functions. Results reveal that the proposed technique achieves significant energy savings in terms of communication overhead in the network.

5. Comparison of Data Mining Techniques for WSNs

This section identifies several common and different aspects of data mining techniques specially designed for WSNs discussed above. These aspects will be used as metrics in the comparative Tables 2, 3, 4, 5, and 6. First, evaluation aspects for different techniques are discussed, and, then, comparative tables are presented to compare and differentiate existing data mining techniques for WSNs data.

5.1. Input Sensor Data. Sensor data can be viewed as large volume of real-valued data that is continuously collected from WSNs. The type of input sensor data demonstrates which data mining techniques can be used to analyze the data. Data mining techniques usually consider following two characteristics of data.

Attribute. Mining techniques can identify the association between data attributes. Attributes can be *homogenous* [50] or *heterogeneou*s [33, 48]. *Homogenous* attribute means sensing single-value attribute, for example, temperature only. For *heterogeneous* case, each node may be equipped with multiple sensors and can sense multiple attributes, for example, temperature, humidity, and pressure. The data mining technique should be able to identify the correlation between multiple attributes.

Correlation. Two types of data correlation appear at each sensor node. The first type is *attribute correlation,* that is, dependency among data attributes. The second type is in terms of time and space, that is, temporal and spatial correlation. *Temporal* correlation indicates that the readings from different sensor node are observed at the same time instant, and readings observed at one time instant are related to the readings observed at the previous time instant, whereas, *spatial* correlation indicates that the readings from sensor nodes geographically close to each other are expected to be largely correlated. Capturing spatiotemporal correlation helps to predict future trend of sensor reading and identification of dead node if reading from correlated sensor is missing.

5.2. Processing Architecture. In order to apply data mining technique on sensor data, we need to determine the models of computation. There are two general models. Consider the following.

Centralized. The simplest way to analyze WSNs data is to use a centralized model. In this approach, entire raw data collected from WSNs is transferred to central server which maintains a database of readings from all of the sensors. The central server performs offline extensive analysis in order to find interesting patterns from the aggregated data. With the size of WSNs increasing, the amount of data transmitted in the system will become huge. The obvious drawback of this approach is high consumption of energy and bandwidth. Furthermore, it is not scalable to very large number of sensors.

Distributed. Another computation approach uses distributed model, in which sensor nodes use their processing abilities to carry out some mining tasks locally and transmit only the required and partially processed data called *local* model. Local models contain the compact event patterns rather than raw data. For example, data collected from different sensor can be aggregated before being transmitted to central server. In these systems, an intermediate node called "*aggregator*" is used to collect and aggregate the data from different sensors. Since sensor nodes are constrained in resources, the challenge for this approach is how to satisfy the mining accuracy while keeping the communication overhead, memory, and computational cost low.

5.3. Data Mining Method. It refers to the data mining algorithm adapted or developed for unique characteristic of WSNs data. Distributed approaches use one-scan algorithms for real-time processing in order to deal with the high data arrival rate; the mining results are expected to be available within short response times, whereas centralized approaches collect the sensory data to single site and applies offline multiscan technique for extensive data analysis.

5.4. Node Properties. The proposed techniques are largely influenced by following types of node properties.

Connectivity. Single-hop communication is a direct communication between the sensor node and the base station. It is simple and easy to implement but limited by communication distance. *Multihop* communication uses some kinds of nodes as relays when transmitting data packets from the source to the sink, which is more complex.

Mobility. Node mobility increases the complexity of designing an appropriate data mining technique for WSNs. The majority of techniques assumes that sensor nodes are *static,* only a few techniques consider the node *mobility.* When nodes are mobile, maintaining a certain structure for data

Table 2: Comparison of data mining techniques for wireless sensor networks.

Approach	Objective	DM method	Processing		Sensor data					Node properties							Node task	Application area	Implementation					Limitations
			Architecture		Attributes		Correlation			Connectivity		Mobility		Node role					Evaluation method		Data source		Opt. objective	
			Distributed	Central	Homogenous	Heterogeneous	Attribute	Spatial	Temporal	Single hop	Multihops	Static	Mobile	Cluster head	Sensor	Relay			Simulation	Analytical Mod.	Real	Synthetic		
											Frequent pattern mining													
DSARM [42]	Missing data estimation	Apriori like		√	√				√	√		√			√		Sense and send	Traffic monitoring	√		√		Data accuracy	Ignore the sensor that reports different values
In-network data mining [51]	Events patterns discovery	Apriori like	√		√			√	√		√	√				√	Aggregation, local pattern mining	Environmental monitoring		√	√	√	Scalability	High memory and communication
Distributed data aggregation [15]	Improve WSN performance	Apriori like	√		√		√				√	√				√	Support-based aggregation	WSNs performance monitoring	√		√		Data size	Increases buffer cost, delayed crucial messages
Online algorithm [46]	Interval list ofrepresentation of WSNs data	Lossy counting		√	√				√	√		√			√		Periodical sensing	WSNs monitoring		√		√	Time and memory	Data redundancy
Lightweight rule learning [48]	Identify highly correlated rules for sensing	Apriori like		√		√	√			√		√			√		Query-based data sensing	Control WSNs operations	√			√	Energy	Not validated well on real data
CARM [43]	Missing data estimation	FP-growth based		√		√			√	√		√			√		Sense and send	Data analysis	√			√	Data accuracy	Inefficient for handling high-speed data

Table 3: Comparison of data mining techniques for wireless sensor networks, continued.

			Processing		Sensor data					Node properties									Implementation					
Approach	Objective	DM method	Architecture		Attributes		Correlation			Connectivity		Mobility		Node role			Node task	Application area	Evaluation method		Data source		Opt. objective	Limitations
			Distributed	Central	Homogenous	Heterogeneous	Attribute	Spatial	Temporal	Single hop	Multihops	Static	Mobile	Cluster head	Sensor	Relay			Simulation	Analytical mod.	Real	Synthetic		
Frequent pattern mining																								
Association rules mining framework [50]	Fault and future event prediction	FP-growth using PLT-structure	√		√				√		√	√			√	√	Aggregation	Monitor WSNs quality of service	√		√		No. of messages	Increase cost due to multiple DB scan
SP-tree [49]	Discover events patterns	FP-growth based		√	√				√	√		√			√		Sense and send	Generic monitoring		√	√	√	Memory	High tree construction cost
Sequential pattern mining																								
Relational framework [58]	Multi-dimensional correlation discovery	Apriori like		√		√			√	√		√			√		Sense and send	Environmental monitoring		√	√		Data representation	Memory and time consuming
Episode discovery (ED) [21]	Action prediction	Generalized sequential pattern (GSP)		√	√				√			√			√		Sense and send	Inhabitants behavior prediction		√	√	√	Prediction accuracy	Inefficient for complex activities
MPG [64]	Predict object's future movement	Apriori like	√		√				√			√		√	√		Clustering	Real-time object tracking		√		√	Tracking time and energy	Not analyzed on real dataset
Contextual patterns discovery [22]	Anomaly detection	PSP		√		√	√			√			√		√		Sense and send	Railway maintenance		√	√		Anomaly precision	Missing real-time anomaly prediction

Table 4: Comparison of data mining techniques for wireless sensor networks, continued.

			Processing		Sensor data					Node properties									Implementation					
Approach	Objective	DM method	Architecture		Attributes		Correlation			Connectivity		Mobility		Node role			Node task	Application area	Evaluation method		Data source		Opt. objective	Limitations
			Distributed	Central	Homogenous	Heterogeneous	Attribute	Spatial	Temporal	Single hop	Multihops	Static	Mobile	Cluster head	Sensor	Relay			Simulation	Analytical mod.	Real	Synthetic		
Sequential pattern mining																								
TMP-mine [65]	Predict object's future movement	Pattern growth using TMP-tree construction	√		√				√		√	√			√		Rule-based node activation	Real-time object tracking	√		√		Energy	High missing rate and time
Pattern learner [23]	Behavior recognition	Tree projection		√		√	√			√		√			√		Sense and send	Behavior monitoring		√	√		No. of patterns learned	Complex and redundant patterns
MSAP [63]	Fault prediction	Candidate construction		√	√		√				√	√			√		Sense and send	Telecommunication		√	√		Patterns accuracy	Candidate construction is expensive to compute
PTSP [66]	Object's future movement prediction	Sequential pattern generation	√		√				√		√	√			√		Rule-based node activation	Object tracking	√			√	Energy	Inefficient to predict high-speed objects
Clustering																								
DCC [86]	WSNs longevity	Data correlation-based clustering	√		√			√			√	√		√	√		Data suppression	Generic WSNs application	√			√	Energy and data size	High clustering rate
H-cluster [85]	In-network communication	Data correlation-based clustering	√			√		√			√	√		√	√		Data summarization	Real-time monitoring	√		√	√	Communication	High data loss rate

Table 5: Comparison of data mining techniques for wireless sensor networks, continued.

Approach	Objective	DM method	Processing		Sensor data					Node properties							Node task	Application area	Implementation					Limitations
			Architecture		Attributes		Correlation			Connectivity		Mobility		Role					Evaluation method		Data source		Opt. objective	
			Distributed	Central	Homogenous	Heterogeneous	Attribute	Spatial	Temporal	Single hop	Multihops	Static	Mobile	Cluster head	Sensor	Relay			Simulation	Analytical mod.	Real	Synthetic		
												Clustering												
Prediction model [87]	Prediction-based monitoring	Heuristic scheme	√		√			√	√		√	√		√	√	√	Local prediction model	Environmental monitoring	√		√		Communication	Cluster overlapping
CAG [88]	WSNs bandwidth gain	Data correlation-based clustering	√		√			√	√		√	√		√	√		Data aggregation	Generic WSNs applications	√		√		Communication	Sensory data loss
EEDC [84]	On-demand clustering	Data correlation-based clustering		√		√		√	√		√	√			√		Sense and send	Surveillance data analysis		√	√	√	Energy	Inefficient for large WSNs
Clustering sensory data [67]	Communication efficiency	K-means	√			√	√				√	√		√	√		Data summarization	Data analysis		√		√	Communication	Inefficient for large WSNs
Attribute based clustering [89]	WSNs bandwidth gain	Hierarchal clustering	√			√	√				√	√		√	√		Data clustering	Monitoring and tracking		√		√	Communication	High computation cost
DHCS [90]	Uniform data distribution	Hierarchal clustering	√			√	√	√			√	√		√	√		Data clustering and summarization	Interactive data analysis			√		Message reduction	Nodes energy is ignored.

Table 6: Comparison of data mining techniques for wireless sensor networks, continued.

Approach	Objective	DM method	Processing		Sensor data					Node properties							Node task	Application area	Implementation				Opt. objective	Limitations
			Architecture		Attributes		Correlation			Connectivity		Mobility		Role					Evaluation method		Data source			
			Distributed	Central	Homogenous	Heterogeneous	Attribute	Spatial	Temporal	Single hop	Multihops	Static	Mobile	Cluster head	Sensor	Relay			Simulation	Analytical mod.	Real	Synthetic		
Classification																								
Person identification algorithms [109]	Identify human behavior	Decision tree		√		√	√			√			√		√		Sense and send	Healthcare		√	√		Classification accuracy	Does not guarantee the correctness
Prediction framework [103]	Distributed prediction	Decision tree	√		√		√					√	√		√	√	Local prediction	Generic	√		√		Prediction accuracy	Computational complexity
NNTC [96]	Real-time classification	Nearest neighbor		√	√			√		√		√			√		Sense and send	Generic	√		√	√	Classification accuracy	Not evaluated on real dataset
LWClass [100]	Preserve WSNs resources	KNN		√		√		√		√		√			√		Sense and send	Ubiquitous environments		√		√	Resource awareness	Nonadaption to concept drift
FVLD [104]	Low-dimension feature vector generation	KNN, ML	√			√		√			√	√		√	√		Classification	Vehicle classification	√		√		Energy	High cost of feature vector transmission
Fuzzy predictor model [99]	Occupancy prediction	Fuzzy rules		√		√			√	√		√			√		Sense and send	Healthcare		√	√		Prediction accuracy	Inefficient for complex scenarios
Online learning [105]	Incremental classification	SVM	√			√	√				√	√		√	√		Classification	Environmental monitoring	√			√	Energy	Computational complexity
One-class quarter-sphere SVM [108]	Anomaly detection	SVM	√			√			√		√	√		√	√		Local anomaly detection	Habitat monitoring	√		√		Energy	Ignores spatial correlation

mining becomes difficult because updates on this structure should be persisted over time.

Node Role. Node can perform three types of role [33] as follows.

(i) *Regular Sensor.* These are the nodes with limited resources, and they are used to sense the phenomena and send the sensed data to the base station.

(ii) *Cluster Head.* Cluster head can be a regular sensor node, or it can be rich in resources. In centralized approaches, cluster head is a regular sensor node that only controls the cluster membership. In distributed approaches, besides responding for cluster formation, CHs perform aggregation/fusion of collected sensors' data. Therefore, they are equipped with significantly more computation and communication resources.

(iii) *Relay.* It is the node that acts as medium to transmit the data packet from one node to the others.

Node Task. In centralized approach, node task is to sense the phenomena being monitored and send the sensed data to the base station. In distributed approaches, node can perform computation and can take action based on the detected phenomena or target.

5.5. Application Area. We also evaluated the type of application benefited from WSNs data mining techniques. Here, we exemplify some real-world applications as follows.

(i) *First is the environmental monitoring* [5–7, 51, 58, 87], in which sensors are deployed in harsh and unattended regions to monitor the natural environment. Data mining techniques can identify when and where an event may occur and trigger an alarm upon detection.

(ii) *Second is the habitant and health monitoring* [1, 2, 99, 109], in which patients/humans are equipped with small sensors on multiple different positions of their body to monitor their health or behavior. Data mining technique can identify the abnormal behavior and help to take effective action.

(iii) *Third is the object tracking* [3, 4, 65, 66]. in which sensors are embedded in moving targets to track them in real-time. Data mining techniques help to improve the estimation of the location of targets and also to make tracking more efficient and accurate.

(iv) *Fourth is the WSNs performance* [46, 48, 50, 51]. WSNs are usually unattended and deployed in harsh environment. Sensor nodes are resource constrained especially in terms of power. Data mining techniques help to identify the faulty or dead nodes. They also help to conserve energy by using in-network processing in which aggregated data is sent to central side.

(v) *Fifth is the data analysis* [67, 84, 90]. Data mining techniques help to discover potentially interesting data patterns in a sensor network for a certain application.

(vi) *Sixth is the real-time monitoring* [64, 65, 85]. Data mining techniques especially distributed techniques help to identify certain patterns and predict future events in a given time window, which make real-time response and action feasible.

5.6. Implementation. Each technique is also evaluated in terms of experimental validation, that is, which dataset is used, which WSNs optimization objectives are achieved, and so forth.

Evaluation Method. Analytical modeling, simulation, and real deployment are the most commonly used techniques to analyze the performance of data mining technique for WSNs.

(i) *Analytical Modeling.* This method is very complex, and usually certain simplifications are assumed to predict the performance of the proposed scheme. Such assumptions and simplifications may lead to imprecise results with limited confidence.

(ii) *Simulation*. It is the most popular and effective approach to design and test any proposed scheme in terms of cost and time; it also provides higher level of details as compared with real implementation. However, the appropriate selection of a simulation framework according to problem and network characteristics is a critical task.

(iii) *Real Deployment*. It may not be feasible to evaluate the performance of these techniques through real deployment due to the unavailability of appropriate hardware in terms of technical and design limitations. Usually, the real deployment requires hundreds of sensor nodes, and cost becomes another important issue. In a nutshell, evaluating any technique proposed for WSNs through real deployment can get the most convincing results although the evaluating process is complex, costly, and time consuming.

Data Source. It refers to dataset use to experimentally validate the proposed technique. Two types of dataset are used generally, that is, *synthetic* and *real*. It is observed from this paper that most of the techniques use the simulation on *synthetic* dataset to validate the result. In this paper it is observed that most of the studies used the simulation due to limited processing power of sensor nodes.

Optimization Objective. Since WSNs are constrained in terms of different resources, the technique is also evaluated in the optimization objective that has been achieved. Most of the techniques consider the resource constraint and different design philosophies of network. None of them can work efficiently for all of the performance metrics like network size, communication overhead, energy efficiency, memory consumption, node mobility, and, and so forth. The large variations in the performance metrics make it a difficult task to present a comprehensive evaluation.

6. Limitations of Existing Data Mining Techniques for WSNs

Tables 2–6 show the characteristics of data mining techniques designed for WSNs. It is observed from comparative analysis that the existing techniques have the following shortcomings.

(i) Most of the techniques do not take into account the heterogeneous data and assume that the sensor data is homogenous [42, 46, 49–51, 65, 87, 110]. They ignore the fact that different attributes together can improve the mining accuracy. In some cases, homogenous data cannot contribute appropriately toward real-time decision.

(ii) The majority of techniques only considers the spatial, or temporal or spatiotemporal correlations [65–67, 87, 88] among sensor data of neighboring nodes and does not consider the attribute dependency among sensor nodes. This in turn increases the computational complexity and reduces the accuracy of mining technique.

(iii) The techniques which consider spatial correlation [51] among sensor data of neighboring nodes suffer from the choice of appropriate neighborhood range. Techniques which consider temporal correlation among sensor data suffers from the choice of the size of the sliding window.

(iv) The majority of techniques uses centralized approach [21, 42–44, 46, 58, 84, 101] in which all data is transmitted to the sink node for identifying certain patterns. These techniques cause much communication overhead and delay the response time. While the techniques that used distributed architecture optimize response time and energy consumption, they have the same problem as that of the centralized approach if the aggregator/cluster head has a large number of nodes under its membership.

(v) Excluding a few, the performance of all of the schemes discussed in this paper has been evaluated with the help of different simulation tools. Although the number of simulators is available and plays an important role for developing and testing new technique, there is always some kind of risk involved as simulation results may not be accurate. In order to analyze a protocol more effectively, it is important to know different available tools and understand the associated benefits and limitations. Due to different performance requirements according to specific applications, a general tool for sensor networks is still lacking at present.

(vi) The techniques evaluated by using analytical modeling [21, 23, 46, 49, 100, 109] used certain simplification and assumption to evaluate the performance of proposed technique. Such assumptions and simplifications may lead to imprecise results with limited confidence. None of the proposed technique is evaluated by using real deployment. Although real deployment is complex, costly, and time consuming, accurate results can only be obtained by using real deployment.

(vii) Excluding a few [22, 103, 109], the majority of techniques assumes that sensor nodes are stationary and do not consider nodes mobility. Applying these techniques for mobile networks or the networks with dynamic changed topology would be challenging.

(viii) Most of the techniques used the synthetic data. Although synthetic data is easily available, there always been chances that results generated on synthetic data are not accurate.

(ix) For the data mining techniques themselves, frequent pattern mining [15–20] approaches suffer from choice of proper and flexible support and confidence threshold. Clustering techniques [11–14] suffer from the choice of an appropriate parameter of cluster width, and computing the distance between data instances in heterogeneous data is computationally expensive, whereas classification-based techniques [24–26] require some prior knowledge to classify the incoming data stream. However, learning accurate classification model is challenging if the number of variables is large in deployed WSNs.

7. Future Research Directions

It is observed from the analysis of existing data mining work on sensor network-based application there are still shortcomings in existing techniques. By seeing these shortcomings and special characteristics of WSNs, there is a need for data mining technique designed for WSNs. The technique should be based on the following requirements.

(i) The technique should combine offline learning mechanisms with distributed and online data processing.

(ii) It should also consider the resource constraint of WSN and its special characteristics such as node mobility and network topology.

(iii) The technique should consider heterogeneous data and dependencies among spatial, temporal, and attribute correlations which may exist between adjacent nodes.

(iv) During online mining, the technique should be capable for incremental learning.

(v) The technique should have low computation complexity and be easy to be implemented.

Based on aforementioned requirements for WSN, a hybrid data mining framework is proposed as shown in Figure 6. In this framework, sensor nodes use their processing abilities to locally carry out mining processing and transmit only the required and partially processed data called *local models*. Single-pass algorithms are applied for network data processing as the data is continuously arriving and not available for the next scan.

Local models contain the compact event patterns rather than raw data which address the issue of communication

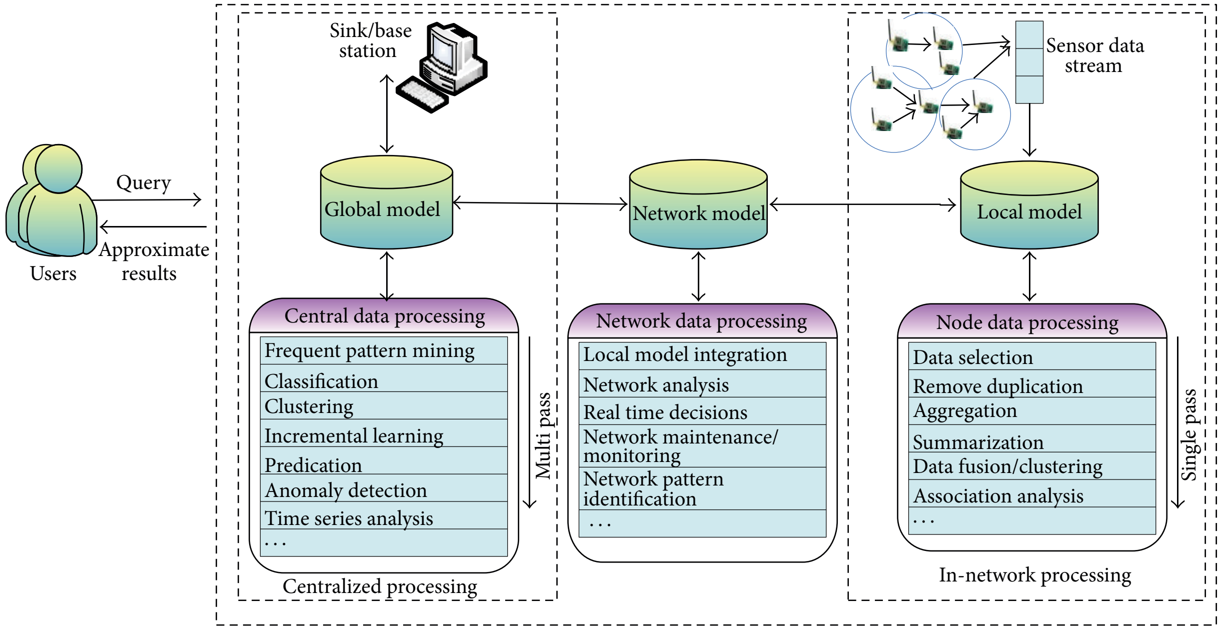


Figure 6: Proposed hybrid framework for sensor network based applications.

overhead associated with data transfer. Local models are distributed on entire network, which are integrated at special node which is resource sufficient as compared with other sensor nodes. As a result, a *network model* is computed that is more abstract than local model and is transferred to the base stationsink in multihop fashion. The network models are then integrated at base stationsink to get the global view of entire network named the *global model*. As a result, approximate query answers are returned to endusers.

This framework addresses the following shortcomings of the existing techniques.

(i) It combines the offline learning mechanisms with distributed and online data processing. The dynamic nature of WSNs data requires real-time analysis methodologies and systems. Centralized processing through high-end computing is also required for generating offline predictive insights, which in turn can facilitate real-time analysis. The applications that require real-time response and actions can use network model for decision and knowledge extraction. The applications that need extensive data analysis for their decision making can use global model and perform central processing on base the station/sink. The network model forwards the processed information to global model for extensive predictive insight.

(ii) Since the data management is a crucial issue in WSNs data [111], in order to deal with large-scale data from WSNs, the proposed framework splits the data processing tasks at multiple locations, in-network processing and processing at central server. In-network processing splits the large task into smaller ones at node level and cluster head which is distributed over the entire network and executes parallelly. At the node level, storage capacities of single nodes are used to compute the local model, which contains aggregated data from single node, whereas cluster head acquires the data from group of nodes and aggregate data readings over a certain region or period. As a result, network model is computed at each cluster head which contains compact data from set of nodes and reduces data size to be transmitted. Network models can be integrated at sink to get the global view of real-time applications. Since the sink at network level has restricted resource and cannot process large-scale data for predictive analysis, therefore, network models are sent to central server where global models can be computed for predictive offline analysis. Historical query from the user can also be addressed from central server, whereas instant query can be handled by sink to support real-time response. In this way of data distribution, the proposed framework is feasible to deal with large amount of data obtained from WSNs.

(iii) It can consider the resource constraint of sensor node by using context-awareness techniques. Memory, energy [79], and bandwidth are considered in the implementation of data processing on the sensors; for example, many summarization and aggregation techniques can be adopted to reduce energy and bandwidth consumption.

(iv) The framework can address the problem quickly changing nature of WSNs data, where characteristics of the monitored process may change over time and render the old models outdated. This problem can be addressed using the incremental learning mechanism [39, 112] that helps the model to update new information.

(v) The framework can identified the spatial-temporal correlation at local model by using data correlation-based clustering, whereas attribute correlation can be identified at global model by using the multipass data mining algorithms.

Currently, we are working on implementation of this hybrid framework, and the implementation will be completed in the near future.

8. Conclusion

The emerging need for the data mining techniques in the field of WSNs resulted in the development of numerous algorithms. Each one of these algorithms solves certain issues related to the appropriate WSNs type and application. In this paper, we analyzed, discussed, and compared the related existing research approaches. We observed that the techniques intended for mining sensor data at the network side are helpful for taking real-time decision as well as serve as prerequisite for development of effective mechanism for data storage, retrieval, query, and transaction processing at central side. Moreover, we have presented problem-based taxonomy, an overall analysis and review of the past research and their limitations which can provide insights for endusers in applying or developing an appropriate data mining method and appropriate technology for WSNs. Based on these limitations, we have proposed a hybrid framework which can address the shortcomings of existing work. We have also discussed the challenges for implementing data mining techniques in resource-constrained WSNs. Besides, there are a number of open issues in existing studies which need to be addressed. Surely, the number of WSNs applications presented here is neither complete nor exhaustive but merely a sample of applications that demonstrate the usefulness and possible applications of data mining method in sensor network.

We believe that WSNs applications will become more mature and popular with the advancement of sensor technology, and sensor data will become more information rich. Mining techniques will then be very significant in order to conduct advanced analysis, such as determining trends and finding interesting patterns thus enhancing WSNs performance and operation. The intention to present this paper is to stimulate interests in utilizing and developing the previous studies into emerging applications.

Acknowledgments

This work was supported in part by the Joint Funds of NSFC-Microsoft Research, Asia, under Grant no. 60933012, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20110142110062, and International S&T Cooperation Program of Hubei Province under Grant no. 2010BFA008.

References

[1] A. Rozyyev, H. Hasbullah, and F. Subhan, “Indoor child tracking in wireless sensor network using fuzzy logic technique,” *Research Journal of Information Technology*, vol. 3, no. 2, pp. 81–92, 2011.

[2] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, “Habitat monitoring with sensor networks,” *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.

[3] S. H. Chauhdary, A. K. Bashir, S. C. Shah, and M. S. Park, “EOATR: energy efficient object tracking by auto adjusting transmission range in wireless sensor network,” *Journal of Applied Sciences*, vol. 9, no. 24, pp. 4247–4252, 2009.

[4] P. K. Biswas and S. Phoha, “Self-organizing sensor networks for integrated target surveillance,” *IEEE Transactions on Computers*, vol. 55, no. 8, pp. 1033–1047, 2006.

[5] L. T. Lee and C. W. Chen, “Synchronizing sensor networks with pulse coupled and cluster based approaches,” *Information Technology Journal*, vol. 7, no. 5, pp. 737–745, 2008.

[6] N. Sabri, S. A. Aljunid, B. Ahmad, A. Yahya, R. Kamaruddin, and M. S. Salim, “Wireless sensor actor network based on fuzzy inference system for greenhouse climate control,” *Journal of Applied Sciences*, vol. 11, no. 17, pp. 3104–3116, 2011.

[7] D. Kumar, “Monitoring forest cover changes using remote sensing and GIS: a global prospective,” *Research Journal of Environmental Sciences*, vol. 5, pp. 105–123, 2011.

[8] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[9] T. Arampatzis, J. Lygeros, and S. Manesis, “A survey of applications of wireless sensors and wireless sensor networks,” in *Proceedings of the 20th IEEE International Symposium on Intelligent Control (ISIC ’05)*, pp. 719–724, June 2005.

[10] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai, “Wireless sensor networks for emergency navigation,” *Computer*, vol. 39, no. 7, pp. 55–62, 2006.

[11] T. Yairi, Y. Kato, and K. Hori, “Fault detection by mining association rules from house-keeping data,” in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp. 18–21, 2001.

[12] O. Horovitz, S. Krishnaswamy, and M. M. Gaber, “A fuzzy approach for interpretation of ubiquitous data stream clustering and its application in road safety,” *Intelligent Data Analysis*, vol. 11, no. 1, pp. 89–108, 2007.

[13] J. Gama, P. P. Rodrigues, and L. Lopes, “Clustering distributed sensor data streams using local processing and reduced communication,” *Intelligent Data Analysis*, vol. 15, no. 1, pp. 3–28, 2011.

[14] Z. A. Aghbari, I. Kamel, and T. Awad, “On clustering large number of data streams,” *Intelligent Data Analysis*, vol. 16, no. 1, pp. 69–91, 2012.

[15] A. Boukerche and S. Samarah, “An efficient data extraction mechanism for mining association rules from wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC ’07)*, pp. 3936–3941, June 2007.

[16] Y. Chi, H. Wang, P. S. Yu, and R. R. Muntz, “Moment: maintaining closed frequent itemsets over a stream sliding window,” in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM ’04)*, pp. 59–66, November 2004.

[17] M. Deypir and M. H. Sadreddini, “EclatDS: an efficient sliding window based frequent pattern mining method for data

streams," *Intelligent Data Analysis*, vol. 15, no. 4, pp. 571–587, 2011.

[18] J. Gama, A. Ganguly, O. Omitaomu, R. Vatsavai, and M. Gaber, "Knowledge discovery from data streams," *Intelligent Data Analysis*, vol. 13, no. 3, pp. 403–404, 2009.

[19] B. George, J. M. Kang, and S. Shekhar, "Spatio-temporal sensor graphs (STSG): a data model for the discovery of spatio-temporal patterns," *Intelligent Data Analysis*, vol. 13, no. 3, pp. 457–475, 2009.

[20] A. Mahmood, K. Shi, and S. Khatoon, "Mining data generated by sensor networks: a survey," *Information Technology Journal*, vol. 11, pp. 1534–1543, 2012.

[21] D. J. Cook, M. Youngblood, E. O. Heierman III et al., "MavHome: an agent-based smart home," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, pp. 521–524, March 2003.

[22] J. Rabatel, S. Bringay, and P. Poncelet, "SO_MAD: sensor mining for anomaly detection in railway data," in *Advances in Data Mining. Applications and Theoretical Aspects*, pp. 191–205, 2009.

[23] V. Guralnik and K. Z. Haigh, "Learning models of human behaviour with sequential patterns," in *Proceedings of the AAAI-02 Workshop on Automation as Caregiver*, pp. 24–30, 2002.

[24] S. Huang and Y. Dong, "An active learning system for mining time-changing data streams," *Intelligent Data Analysis*, vol. 11, no. 4, pp. 401–419, 2007.

[25] J. Beringer and E. Hüllermeier, "Efficient instance-based learning on data streams," *Intelligent Data Analysis*, vol. 11, no. 6, pp. 627–650, 2007.

[26] E. J. Spinosaa, A. P. D. L. F. de Carvalhoa, and J. Gamab, "Novelty detection with application to data streams," *Intelligent Data Analysis*, vol. 13, no. 3, pp. 405–422, 2009.

[27] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[28] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[29] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, article 15, 2009.

[30] V. Maojo and J. Sanandré, "A survey of data mining techniques," *Medical Data Analysis, Lecture Notes in Computer Science*, vol. 1933, pp. 17–22, 2000.

[31] W. Jinlong, X. Congfu, C. Weidong, and P. Yunhe, "Survey of the study on frequent pattern mining in data streams," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 5917–5922, October 2004.

[32] J. Cheng, Y. Ke, and W. Ng, "A survey on algorithms for mining frequent itemsets over data streams," *Knowledge and Information Systems*, vol. 16, no. 1, pp. 1–27, 2008.

[33] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[34] O. Boyinbode, H. Le, and M. Takizawa, "A survey on clustering algorithms for wireless sensor networks," *International Journal of Space-Based and Situated Computing*, vol. 1, no. 2, pp. 130–136, 2007.

[35] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "A survey of classification methods in data streams," in *Data Streams*, pp. 39–59, Springer, 2007.

[36] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference Very Large Data Bases (VLDB '94)*, pp. 487–499, Citeseer, 1994.

[37] R. J. Bayardo Jr., "Efficiently mining long patterns from databases," *SIGMOD Record*, vol. 27, no. 2, pp. 85–93, 1998.

[38] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: generalizing association rules to correlations," *SIGMOD Record*, vol. 26, no. 2, pp. 265–276, 1997.

[39] W. Cheung and O. R. Zaiane, "Incremental mining of frequent patterns without candidate generation or support constraint," in *Proceedings of 7th International Database Engineering and Applications Symposium*, pp. 111–116, 2003.

[40] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceeding of SIGMOD*, pp. 207–216.

[41] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[42] M. Halatchev and L. Gruenwald, "Estimating missing values in related sensor data streams," in *Proceedings of the 11th International Conference on Management of Data (COMAD '05)*, 2005.

[43] N. Jiang, "Discovering association rules in data streams based on closed pattern mining," in *Proceedings of the SIGMOD Workshop on Innovative Database Research*, 2007.

[44] N. Jiang and L. Gruenwald, "Estimating missing data in data streams," *Advances in Databases: Concepts, Systems and Applications*, pp. 981–987, 2007.

[45] N. Jiang and L. Gruenwald, "CFI-stream: mining closed frequent itemsets in data streams," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 592–597, August 2006.

[46] K. Loo, I. Tong, and B. Kao, "Online algorithms for mining inter-stream associations from large sensor networks," in *Advances in Knowledge Discovery and Data Mining*, pp. 291–302, 2005.

[47] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," in *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 346–357, 2002.

[48] S. K. Chong, S. Krishnaswamy, S. W. Loke, and M. M. Gaber, "Using association rules for energy conservation in wireless sensor networks," in *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC '08)*, pp. 971–975, March 2008.

[49] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Efficient mining of association rules from wireless sensor networks," in *Proceedings of the 11th International Conference on Advanced Communication Technology (ICACT '09)*, pp. 719–724, February 2009.

[50] A. Boukerche and S. Samarah, "A novel algorithm for mining association rules in Wireless Ad Hoc Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 865–877, 2008.

[51] K. Romer, "Distributed mining of spatio-temporal event patterns in sensor networks," in *Proceedings of the 1st Euro-American Workshop on Middleware for Sensor Networks (EAWMS '06)*, 2006.

[52] BTnode platform, http://www.btnode.ethz.ch/.

[53] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the IEEE 11th International Conference on Data Engineering*, pp. 3–14, March 1995.

[54] R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," in *Proceedings of the Advances in Database Technology (EDBT '96)*, pp. 1–17, 1996.

[55] F. Masseglia, F. Cathala, and P. Poncelet, "The PSP approach for mining sequential patterns," *Principles of Data Mining and Knowledge Discovery*, pp. 176–184, 1998.

[56] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu, "FreeSpan: frequent pattern-projected sequential pattern mining," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 355–359, August 2000.

[57] J. Pei, J. Han, B. Mortazavi-Asl et al., "PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth," in *Proceedings of the 17th International Conference on Data Engineering*, pp. 215–224, April 2001.

[58] F. Esposito, T. M. A. Basile, N. Di Mauro, and S. Ferilli, "A relational approach to sensor network data mining," *Information Retrieval and Mining in Distributed Environments*, pp. 163–181, 2010.

[59] F. Esposito, N. Di Mauro, T. M. A. Basile, and S. Ferilli, "Multi-dimensional relational sequence mining," *Fundamenta Informaticae*, vol. 89, no. 1, pp. 23–43, 2008.

[60] R. Agrawal, H. Mannila, R. Srikant et al., "Fast discovery of association rules," in *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, AAAI Press, Menlo Park, Calif, USA, 1996.

[61] Mica2Dot, CrossBow, 2005, http://www.xbow.com/.

[62] Intel Berkeley Research Lab Data, http://db.csail.mit.edu/labdata/labdata.html.

[63] P. H. Wu, W. C. Peng, and M. S. Chen, "Mining sequential alarm patterns in a telecommunication database," in *Databases in Telecommunications II*, pp. 37–51, 2001.

[64] V. S. Tseng and E. H.-C. Lu, "Energy-efficient real-time object tracking in multi-level sensor networks by mining and predicting movement patterns," *Journal of Systems and Software*, vol. 82, no. 4, pp. 697–706, 2009.

[65] V. S. Tseng and K. W. Lin, "Energy efficient strategies for object tracking in sensor networks: a data mining approach," *Journal of Systems and Software*, vol. 80, no. 10, pp. 1678–1698, 2007.

[66] S. Samarah, M. Al-Hajri, and A. Boukerche, "A predictive energy-efficient technique to support object-tracking sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 656–663, 2011.

[67] A. Taherkordi, R. Mohammadi, and F. Eliassen, "A communication-efficient distributed clustering algorithm for sensor networks," in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINA '08)*, pp. 634–638, March 2008.

[68] G. Gupta and M. Younis, "Load-balanced clustering of wireless sensor networks," in *Proceedings of the International Conference on Communications (ICC '03)*, vol. 3, pp. 1848–1852, May 2003.

[69] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1713–1723, April 2003.

[70] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, "Optimal energy aware clustering in sensor networks," *Sensors*, vol. 2, no. 7, pp. 258–269, 2002.

[71] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[72] M. Younis, M. Youssef, and K. Arisha, "Energy-aware management for cluster-based sensor networks," *Computer Networks*, vol. 43, no. 5, pp. 649–668, 2003.

[73] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2579–2590, 2005.

[74] T. Wu and S. Biswas, "A self-reorganizing slot allocation protocol for multi-cluster sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 309–316, April 2005.

[75] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, vol. 3, pp. 1948–1953, 2003.

[76] M. Demirbas, A. Arora, and V. Mittal, "FLOC: A fast local clustering service for wireless sensor networks," in *Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '04)*, 2004.

[77] P. Ding, J. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, pp. 466–467, July 2005.

[78] H. Chan and A. Perrig, "ACE: an emergent algorithm for highly uniform cluster formation," *Wireless Sensor Networks*, vol. 2920, pp. 154–171, 2004.

[79] H. Chan, M. Luk, and A. Perrig, "Using clustering information for sensor network localization," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, pp. 109–125, July 2005.

[80] H. Huang and J. Wu, "A probabilistic clustering algorithm in wireless sensor networks," in *Proceeding of IEEE 62nd Semiannual Vehicular Technology Conference (VTC '05)*, p. 1796, 2005.

[81] A. Youssef, M. Younis, M. Youssef, and A. Agrawala, "Distributed formation of overlapping multi-hop clusters in wireless sensor networks," in *Proceedings of the 49th Annual IEEE Global Communication Conference (Globecom '06)*, pp. 1–6, December 2006.

[82] S. Dai, P. Wang, L. Gao, and S. Zheng, "Mining clustering algorithm in wireless sensor networks," in *Proceedings of the IEEE International Conference on Granular Computing (GRC '08)*, pp. 178–182, August 2008.

[83] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, vol. 2, p. 223, January 2000.

[84] C. Liu, K. Wu, and J. Pei, "A dynamic clustering and scheduling approach to energy saving in data collection from wireless sensor networks," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and AdHoc Communications and Networks (SECON '05)*, pp. 374–385, September 2005.

[85] L. Guo, C. Ai, X. Wang, Z. Cai, and Y. Li, "Real time clustering of sensory data in wireless sensor networks," in *Proceedings of the IEEE 28th International Performance Computing and Communications Conference (IPCCC '09)*, pp. 33–40, December 2009.

[86] M. H. Yeo, M. S. Lee, S. J. Lee, and J. S. Yoo, "Data correlation-based clustering in sensor networks," in *Proceedings of the International Symposium on Computer Science and its Applications (CSA '08)*, pp. 332–337, October 2008.

[87] P. Beyens, A. Nowé, and K. Steenhaut, "High-density wireless sensor networks: a new clustering approach for prediction-based monitoring," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN '05)*, pp. 188–196, February 2005.

[88] S. Yoon and C. Shahabi, "The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 1, Article ID 1210672, 2007.

[89] K. Wang, S. A. Ayyash, T. D. C. Little, and P. Basu, "Attribute-based clustering for information dissemination in wireless sensor networks," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and AdHoc Communications and Networks (SECON '05)*, pp. 498–509, Santa Clara, Calif, USA, September 2005.

[90] X. Ma, S. Li, Q. Luo et al., "Distributed, hierarchical clustering and summarization in sensor networks," in *Advances in Data and Web Management*, pp. 168–175, 2007.

[91] L. K. Sharma, O. P. Vyas, S. Schieder et al., "Nearest neighbour classification for trajectory data," *Information and Communication Technologies*, vol. 101, pp. 180–185, 2010.

[92] B. Chikhaoui, S. Wang, and H. Pigot, "A new algorithm based on sequential pattern mining for person identification in ubiquitous environments," in *Proceedings of the 4th International Workshop on Knowledge Discovery form Sensor Data (ACM SensorKDD '10)*, pp. 20–28, Washington, DC, USA, 2010.

[93] J. R. M. Bauchet, S. Giroux, H. Pigot et al., "Pervasive assistance in smart homes for people with intellectual disabilities: a case study on meal preparation," *International Journal of Assistive Robotics and Mechatronics*, vol. 9, no. 4, pp. 42–54, 2008.

[94] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods of Information in Medicine*, vol. 48, no. 5, pp. 480–485, 2009.

[95] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques With Java Implementation*, Morgan Kaufmann, 2000.

[96] K. Sharma, M. Rajpoot, and L. K. Sharma, "Nearest neighbour classification for wireless sensor network data," *International Journal of Computer Trends and Technology*, no. 2, 2011.

[97] NS2 Simulator, http://www.isi.edu/nsnam/ns/.

[98] O. P. V. L. K. Sharma, S. Schieder, and A. K. Akasapu, "A nearest neighbour classification for trajectory data," in *Springer CCIS*, vol. 101, pp. 180–185, 2010.

[99] M. J. Akhlaghinia, A. Lotfi, C. Langensiepen, and N. Sherkat, "A fuzzy predictor model for the occupancy prediction of an intelligent inhabited environment," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '08)*, pp. 939–946, June 2008.

[100] M. Gaber, S. Krishnaswamy, and A. Zaslavsky, "On-board mining of data streams in sensor networks," in *Advanced Methods for Knowledge Discovery from Complex Data*, pp. 307–335, 2005.

[101] M. M. Gaber, S. Krishnaswamy, and A. Zaslavsky, "Adaptive mining techniques for data streams using algorithm output granularity," in *Proceedings of the Australasian Data Mining Workshop*, 2003.

[102] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Resource-aware knowledge discovery in data streams," in *Proceedings of 1st International Workshop on Knowledge Discovery in Data Streams, held in Conjunction ECML and PKDD*, 2004.

[103] S. M. McConnell and D. B. Skillicorn, "A distributed approach for prediction in sensor networks," in *Proceedings of the Workshop on Data Mining in Sensor Networks*, Newport Beach, Calif, USA, 2005.

[104] B. Malhotra, I. Nikolaidis, and J. Harms, "Distributed classification of acoustic targets in wireless audio-sensor networks," *Computer Networks*, vol. 52, no. 13, pp. 2582–2593, 2008.

[105] K. Flouri, B. Beferull-Lozano, and T. Tsakalides, "Training a SVM-based classifier in distributed sensor networks," in *Proceedings of the 14th International Conference on Digital Signal Processing (DSP '09)*, pp. 1–5, 2006.

[106] K. Flouri, B. Beferull-Lozano, and T. Tsakalides, "Energy-efficient distributed support vector machines for wireless sensor networks," in *Proceedings of the European Workshop on Wireless Sensor Networks*, 2006.

[107] K. Flouri, B. Beferull-Lozano, and T. Tsakalides, "Distributed consensus algorithms for SVM training in wireless sensor networks," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO 09)*, 2008.

[108] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 3864–3869, June 2007.

[109] B. Chikhaoui, S. Wang, and H. Pigot, "A new algorithm based on sequential pattern mining for person identification in ubiquitous environments," in *Proceedings of the 4th International Workshop on Knowledge Discovery form Sensor Data (ACM SensorKDD '10)*, pp. 20–28, Washington, DC, USA, 2010.

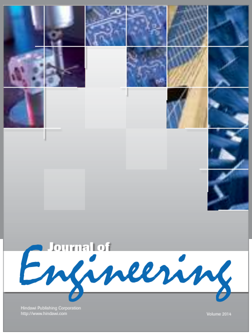
[110] K. Römer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.

[111] O. Diallo, J. J. P. C. Rodrigues, and M. Sene, "Real-time data management on wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1013–1021, 2012.

[112] Y. Yao, L. Feng, B. Jin, and F. Chen, "An incremental learning approach with Support Vector Machine for network data stream classification problem," *Information Technology Journal*, vol. 11, no. 2, pp. 200–208, 2012.

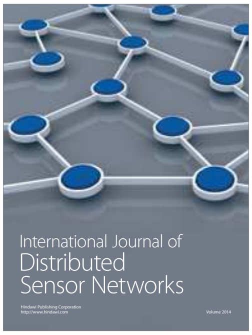
The Scientific World Journal

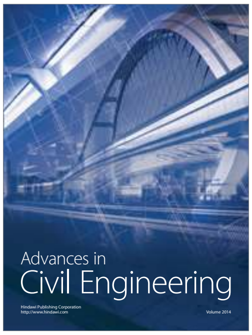
International Journal of
Rotating Machinery

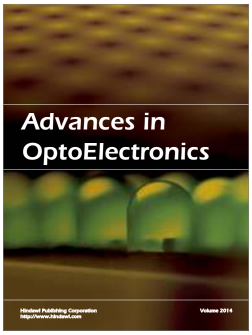

Journal of
Engineering

Journal of
Sensors

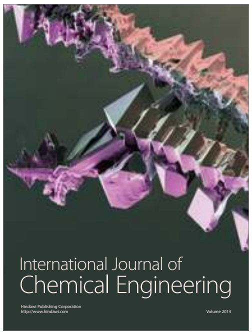
Advances in
Mechanical Engineering


International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering

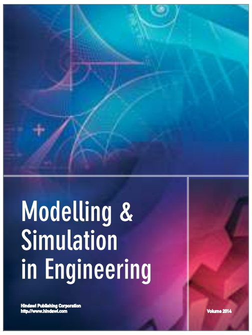

Advances in
OptoElectronics

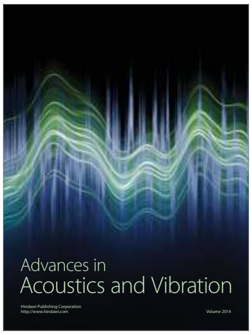
Journal of
Robotics


International Journal of
Chemical Engineering

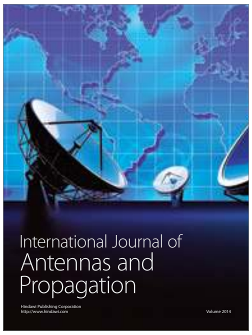
VLSI Design

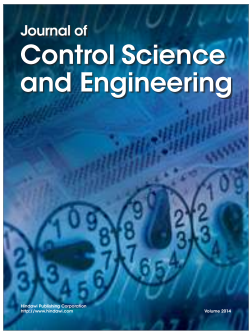
International Journal of
Navigation and Observation

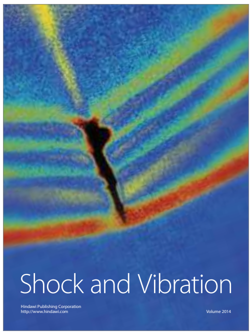

Modelling & Simulation in Engineering


Advances in
Acoustics and Vibration

Active and Passive
Electronic Components


International Journal of
Antennas and Propagation


Journal of
Control Science and Engineering


Shock and Vibration

Journal of
Electrical and Computer Engineering