Machine Learning in Wireless Sensor Networks: Algorithms, Strategies and Applications

Mohammad Abu Alsheikh[1,2], Shaowei Lin[2], Dusit Niyato[1] and Hwee-Pink Tan[2]

[1]*School of Computer Engineering, Nanyang Technological University, Singapore 639798*
[2]*Sense and Sense-abilities Programme, Institute for Infocomm Research, Singapore 138632*

***Abstract*—Wireless sensor networks monitor dynamic environments that change rapidly over time. This dynamic behavior is either caused by external factors or initiated by the system designers themselves. To adapt to such conditions, sensor networks often adopt machine learning techniques to eliminate the need for unnecessary redesign. Machine learning also inspires many practical solutions that maximize resource utilization and prolong the lifespan of the network. In this paper, we present an extensive literature review over the period 2002-2013 of machine learning methods that were used to address common issues in wireless sensor networks (WSNs). The advantages and disadvantages of each proposed algorithm are evaluated against the corresponding problem. We also provide a comparative guide to aid WSN designers in developing suitable machine learning solutions for their specific application challenges.**

***Index Terms*—Wireless sensor networks, machine learning, data mining, security, localization, clustering, data aggregation, event detection, query processing, data integrity, fault detection, medium access control, compressive sensing.**

I. INTRODUCTION

A wireless sensor network (WSN) is composed typically of multiple autonomous, tiny, low cost and low power sensor nodes. These nodes gather data about their environment and collaborate to forward sensed data to centralized backend units called base stations or sinks for further processing. The sensor nodes could be equipped with various types of sensors, such as thermal, acoustic, chemical, pressure, weather, and optical sensors. Because of this diversity, WSNs have tremendous potential for building powerful applications, each with its own individual characteristics and requirements. Developing efficient algorithms that are suitable for many different application scenarios is a challenging task. In particular, WSN designers have to address common issues related to data aggregation, data reliability, localization, node clustering, energy aware routing, events scheduling, fault detection and security.

Machine learning (ML) was introduced in the late 1950's as a technique for artificial intelligence (AI) [1]. Over time, its focus evolved and shifted more to algorithms which are computationally viable and robust. In the last decade, machine learning techniques have been used extensively for a wide range of tasks including classification, regression and density estimation in a variety of application areas such as bioinformatics, speech recognition, spam detection, computer vision, fraud detection and advertising networks. The algorithms and techniques used come from many diverse fields including statistics, mathematics, neuroscience, and computer science. The following two classical definitions capture the essence of machine learning:

1) The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems [2].
2) The adoption of computational methods for improving machine performance by detecting and describing consistencies and patterns in training data [3].

Applying these definitions to WSNs, we see that the promise of machine learning lies in exploiting historical data to improve the performance of sensor networks on given tasks without the need for re-programming. More specifically, machine learning is important in WSN applications for the following main reasons:

1) Sensor networks usually monitor dynamic environments that change rapidly over time. For example, a node's location may change due to soil erosion or sea turbulence. It is desirable to develop sensor networks that can adapt and operate efficiently in such environments.
2) WSNs may be used for collecting new knowledge about unreachable, dangerous locations [4] (e.g., volcano eruption and waste water monitoring) in exploratory applications. Due to the unexpected behavior patterns that may arise in such scenarios, system designers may develop solutions that initially may not operate as expected. System designers would rather have robust machine learning algorithms that are able to calibrate itself to newly acquired knowledge.
3) WSNs are usually deployed in complicated environments where researchers cannot build accurate mathematical models to describe the system behavior. Meanwhile, some tasks in WSNs can be prescribed using simple mathematical models but may still need complex algorithms to solve them (e.g., the routing problem [5], [6]). Under similar circumstances, machine learning provides low-complexity estimates for the system model.
4) Sensor network designers often have access to large amounts of data but may be unable to extract important correlations in them. For example, in addition to ensuring communication connectivity and energy sustainability, the WSN application often comes with minimum data coverage requirements that have to be fulfilled by limited sensor hardware resources [7]. Machine learning methods can then be used to discover important corre-

lations in the sensor data and propose improved sensor deployment for maximum data coverage.

5) New uses and integrations of WSNs, such as in cyber-physical systems (CPS), machine-to-machine (M2M) communications, and Internet of things (IoT) technologies, have been introduced with a motivation of supporting more intelligent decision-making and autonomous control [8]. Here, machine learning is important to extract the different levels of abstractions needed to perform the AI tasks with limited human intervention [9].

However, there are a few drawbacks and limitations that should be considered when using machine learning techniques in wireless sensor networks. Some of these are:

1) As a resource limited framework, WSN drains a considerable percentage of its energy budget to predict the accurate hypothesis and extract the consensus relationship among data samples. Thus, the designers should consider the trade-off between the algorithm's computational requirements and the learned model's accuracy. Specifically, the higher the required accuracy, the higher the computational requirements, and the higher energy consumptions. Otherwise, the developed systems might be employed with centralized and resource capable computational units to perform the learning task.
2) Generally speaking, learning by examples requires a large data set of samples to achieve the intended generalization capabilities (i.e., fairly small error bounds), and the algorithm's designer will not have the full control over the knowledge formulation process [10].

During the past decade, WSNs have seen increasingly intensive adoption of advanced machine learning techniques. In [11], a short survey of machine learning algorithms applied in WSNs for information processing and for improving network performance was presented. A related survey that discussed the applications of machine learning in wireless ad-hoc networks was published in [12]. The authors of [13] discussed applications of three popular machine learning algorithms (i.e., reinforcement learning, neural networks and decision trees) at all communication layers in the WSNs. In contrast, specialized surveys that touch on machine learning usage in specific WSN challenges have also been written. For instance, [14], [15] addressed the development of efficient outlier detection techniques so that proper actions can be taken, and some of these techniques are based on concepts from machine learning. Meanwhile, [16] discusses computational intelligence methods for tackling challenges in WSNs such as data aggregation and fusion, routing, task scheduling, optimal deployment and localization. Here, computational intelligence is a branch of machine learning that focuses on biologically-inspired approaches such as neural networks, fuzzy systems and evolutionary algorithms [17].

Generally, these early surveys concentrated on reinforcement learning, neural networks and decision trees which were popular due to their efficiency in both theory and practice. In this paper, we decided instead to include a wide variety of important *up-to-date* machine learning algorithms for a comparison of their strengths and weaknesses. In particular, we provide a comprehensive overview which groups these recent techniques roughly into supervised, unsupervised and reinforcement learning methods. Another distinction between our survey and earlier works is the way that machine learning techniques are presented. Our work discusses machine learning algorithms based on their target WSN challenges, so as to encourage the adoption of existing machine learning solutions in WSN applications. Lastly, we build on existing surveys and go beyond classifying and comparing previous efforts, by providing useful and practical guidelines for WSN researchers and engineers who are interested in exploring new machine learning paradigms for future research.

The rest of the paper is organized as follows:

- Section II introduces the reader to machine learning algorithms and themes that will be referred to in later sections. Simple examples will be given in the context of WSNs.
- In Section III, we review existing machine learning efforts to address *functional* issues in WSNs such as routing, localization, clustering, data aggregation, query processing and medium access control. Here, an issue is functional if it is essential to the basic operation of the wireless sensor network.
- Section IV investigates machine learning solutions in WSNs for fulfilling *non-functional* requirements, i.e. those which determine the quality or enhance the performance of functional behaviors. Examples of such requirements include security, quality of service (QoS) and data integrity. In this section, we also highlight some unique efforts in specialized WSN applications.
- Section V outlines major difficulties and open research problems for machine learning in WSNs.
- Finally, we conclude in Section VI and present a comparative guide with useful paradigms for furthering machine learning research in various WSN applications.

II. INTRODUCTION TO MACHINE LEARNING IN WIRELESS SENSOR NETWORKS

Usually, sensor network designers characterize machine learning as a collection of tools and algorithms that are used to create prediction models. However, machine learning experts recognize it as a rich field with very large themes and patterns. Understanding such themes will be beneficial to those who wish to apply machine learning to WSNs. Applied to numerous WSNs applications, machine learning algorithms provide tremendous flexibility benefits. This section provides some of the theoretical concepts and strategies of adopting machine learning in the context of WSNs.

Existing machine learning algorithms can be categorized by the intended structure of the model. Most machine learning algorithms fall into the categories of supervised, unsupervised and reinforcement learning [18]. In the first category, machine learning algorithms are provided with a labeled training data set. This set is used to build the system model representing the learned relation between the input, output and system parameters. In contrast to supervised learning, unsupervised

learning algorithms are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithm is to classify the sample sets to different groups (i.e., clusters) by investigating the similarity between the input samples. The third category includes reinforcement learning algorithms, in which the agent learns by interacting with its environment (i.e., online learning). Finally, some machine learning algorithms do not naturally fit into this classification since they share characteristics of both supervised and unsupervised learning methods. These hybrid algorithms (often termed as semi-supervised learning) aim to inherit the strengths of these main categories, while minimizing their weaknesses [19].

This section is mainly to introduce the reader to the algorithms that will be referred to in later sections. Moreover, examples will be given to demonstrate the process of adopting machine learning in WSNs. In Sections III and IV, such details will be omitted. For interested reader, please refer to [18], [20] and references therein, for thorough discussions of machine learning theory and its classical concepts.

A. Supervised Learning

In supervised learning, a labeled training set (i.e., predefined inputs and known outputs) is used to build the system model. This model is used to represent the learned relation between the input, output and system parameters. In this subsection, the major supervised learning algorithms are discussed in the context of WSNs. In fact, supervised learning algorithms are extensively used to solve several challenges in WSNs such as localization and objects targeting (e.g., [21], [22], [23]), event detection and query processing (e.g., [24], [25], [26], [27]), media access control (e.g., [28], [29], [30]), security and intrusion detection (e.g., [31], [32], [33], [34]), and quality of service (QoS), data integrity and fault detection (e.g., [35], [36], [37]).

1) K-nearest neighbor (k-NN): This supervised learning algorithm classifies a data sample (called a query point) based on the labels (i.e., the output values) of the near data samples. For example, missing readings of a sensor node can be predicted using the average measurements of neighboring sensors within specific diameter limits. There are several functions to determine the nearest set of nodes. A simple method is to use the Euclidean distance between different sensors. K-nearest neighbor does not need high computational power, as the function is computed relative to local points (i.e., k-nearest points, where k is a small positive integer). This factor coupled with the correlated readings of neighboring nodes makes k-nearest neighbor a suitable distributed learning algorithm for WSNs. In [38], it has been shown that the k-NN algorithm may provide inaccurate results when analyzing problems with high-dimensional spaces (more than 10-15 dimensions) as the distance to different data samples becomes invariant (i.e., the distances to the nearest and farthest neighbors are slightly similar). In WSNs, the most important application of the k-nearest neighbor algorithm is in the query processing subsystem (e.g., [24], [25]).

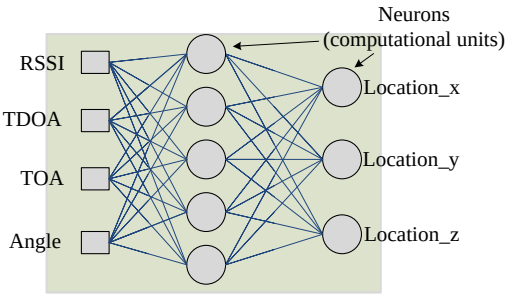


Figure 1: Illustration example of node localization in WSNs in 3D space using supervised neural networks.

2) Decision tree (DT): It is a classification method for predicting labels of data by iterating the input data through a learning tree [39]. During this process, the feature properties are compared relative to decision conditions to reach a specific category. The literature is very rich with solutions that use DT algorithm to resolve different WSNs' design challenges. For example, DT provides a simple, but efficient method to identify link reliability in WSNs by identifying a few critical features such as loss rate, corruption rate, mean time to failure (MTTF) and mean time to restore (MTTR). However, DT works only with linearly separable data and the process of building optimal learning trees is NP-complete [40].

3) Neural networks (NNs): This learning algorithm could be constructed by cascading chains of decision units (e.g., perceptrons or radial basis functions) used to recognize non-linear and complex functions [9]. In WSNs, using neural networks in distributed manners is still not so pervasive due to the high computational requirements for learning the network weights, as well as the high management overhead. However, in centralized solutions, neural networks can learn multiple outputs and decision boundaries at once [41], which makes them suitable for solving several network challenges using the same model.

We consider a sensor node localization problem (i.e., determining node's geographical position) as an application example of neural network in WSNs. Node localization can be based on propagating angle and distance measurements of the received signals from anchor nodes [42]. Such measurements may include received signal strength indicator (RSSI), time of arrival (TOA), and time difference of arrival (TDOA) as illustrated in Figure 1. After supervised training, neural network generates an estimated node location as vector-valued coordinates in 3D space. Related algorithms to neural networks include self-organizing map (or Kohonen's maps) and learning vector quantization (LVQ) (see [43] and references therein for an introduction to these methods). In addition to function estimation, one of the important applications of neural networks is for big data (high-dimensional and complex data set) tuning and dimensionality reduction [44].

4) Support vector machines (SVMs): It is a machine learning algorithm that learns to classify data points using labeled training samples [45]. For example, one approach for detecting malicious behavior of a node is by using SVM to investigate temporal and spatial correlations of data. To illustrate, given

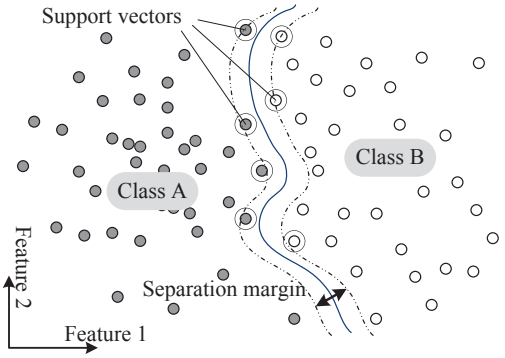


Figure 2: An example of non-linear support vector machines.

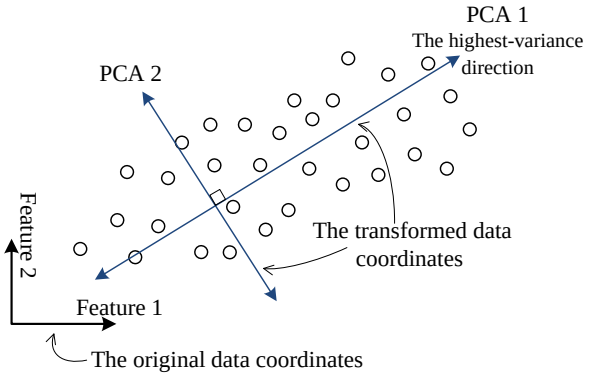


Figure 3: A simple 2D visualization of the principal component analysis algorithm. It is important to note that the potential of the PCA algorithm is high mainly when dealing with high-dimensional data [62].

WSN's observations as points in the feature space, SVM divides the space into parts. These parts are separated by as wide as possible margins (i.e., separation gaps), and new reading will be classified based on which side of the gaps they fall on as shown in Fig. 2. An SVM algorithm, which includes optimizing a quadratic function with linear constraints (that is, the problem of constructing a set of hyperplanes), provides an alternative method to the multi-layer neural network with non-convex and unconstrained optimization problem [39]. Potential applications of SVM in WSNs are security (e.g., [33], [34], [46], [47], [48]) and localization (e.g., [49], [50], [51]). For a detailed discussion of the SVM theory, please refer to [45].

5) Bayesian statistics: Unlike most machine learning algorithms, Bayesian inference requires a relatively small number of training samples [52]. Bayesian methods adapt probability distribution to efficiently learn uncertain concepts (e.g., θ) without over-fitting. The crux of the matter is to use the current knowledge (e.g., collected data abbreviated as D) to update prior beliefs into posterior beliefs $p(\theta|D) \propto p(\theta)p(D|\theta)$, where $p(\theta|D)$ is the posterior probability of the parameter θ given the observation D, and $p(D|\theta)$ is the likelihood of the observation D given the parameter θ. One application of Bayesian inference in WSNs is assessing event consistency (θ) using incomplete data sets (D) by investigating prior knowledge about the environment. However, such statistical knowledge requirement limits the wide adoption of Bayesian algorithms in WSNs. A related statistical learning algorithm is Gaussian process regression (GPR) model [53].

B. Unsupervised Learning

Unsupervised learners are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithm is to classify the sample set into different groups by investigating the similarity between them. As expected, this theme of learning algorithms is widely used in node clustering and data aggregation problems (e.g., [54], [55], [56], [57], [58], [59], [60]). Indeed, this wide adoption is due to data structures (i.e., no labeled data is available) and the desired outcome in such problems.

1) K-means clustering: The k-means algorithm [61] is used to recognize data into different classes (known as clusters). This unsupervised learning algorithm is widely used in sensor node clustering problem due to its linear complexity and simple implementation. The k-means steps to resolve such node clustering problem are (a) randomly choose k nodes to be the initial centroids for different clusters; (b) label each node with the closest centroid using a distance function; (c) recompute the centroids using the current node memberships and (d) stop if the convergence condition is valid (e.g., a predefined threshold for the sum of distances between nodes and their perspective centroids), otherwise go back to step (b).

2) Principal component analysis (PCA): It is a multivariate method for data compression and dimensionality reduction that aims to extract important information from data and present it as a set of new orthogonal variables called principal components [62]. As shown in Fig. 3, the principal components are ordered such that the first component corresponds to the highest-variance direction of the data, and so on for the other components. Hence, the least-variance components can be discarded as they contain the least information content. For example, PCA reduces the amount of transmitted data among sensor nodes by finding a small set of uncorrelated linear combinations of original readings. Furthermore, the PCA method simplifies the problem solving by considering only few conditions in very large variable problems (i.e., tuning big data into tiny data representation) [63]. A thorough discussion of the PCA theory (e.g., the eigenvalue, eigenvector, and covariance matrix analysis) is given in [62].

C. Reinforcement Learning

Reinforcement learning enables an agent (e.g., a sensor node) to learn by interacting with its environment. The agent will learn to take the best actions that maximize its long-term rewards by using its own experience. The most well-known reinforcement learning technique is Q-learning [64]. As shown in Fig. 4, an agent regularly updates its achieved rewards based on the taken action at a given state. The future total reward (i.e., the Q-value) of performing an action a_t at a given state s_t is computed using Eq. (1).

$$Q\left(s_{t+1}, a_{t+1}\right) = Q\left(s_t, a_t\right) + \gamma\left(r\left(s_t, a_t\right) - Q\left(s_t, a_t\right)\right) \quad (1)$$

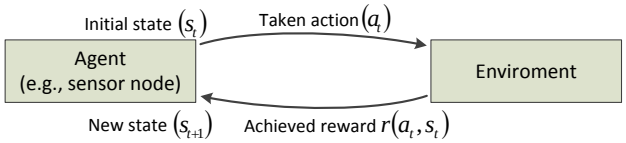


Figure 4: A Visualization of the Q-learning method.

where $r(s_t, a_t)$ denotes the immediate reward of performing an action a_t at a given state s_t, and γ is the learning rate that determines how fast learning occurs (usually set to value between 0 and 1). This algorithm can be easily implemented in a distributed architecture like WSNs, where each node seeks to choose actions that are expected to maximize its long term rewards. It is important to note that Q-learning has been extensively and efficiently used in WSN routing problem (e.g., [65], [66], [67], [68]).

III. FUNCTIONAL CHALLENGES

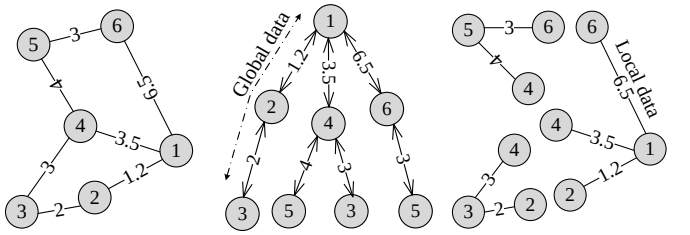
In the design of WSNs, it is important to consider power and memory constraints of sensor nodes, topology changes, communication link failures, and decentralized management. Machine learning paradigms have been successfully adopted to address various functional challenges of wireless sensor networks such as energy aware and real-time routing, query processing and event detection, localization, node clustering and data aggregation.

A. Routing in WSNs

Designing a routing protocol for WSNs has to consider various design challenges such as energy consumption, fault tolerance, scalability, and data coverage [6]. Sensor nodes are provided with limited processing capabilities, small memory and low bandwidth. Traditionally, it is common to formulate a routing problem in wireless sensor networks as a graph $G = (V, E)$, where V represents the set of all nodes, and E represents the set of bidirectional communication channels connecting the nodes. Using this model, the routing problem can be defined as the process of finding the minimum cost path starting at the source vertex, and reaching all destination vertices, by using the available graph edges. This path is actually a spanning tree $T = (V, E)$ whose vertices include the source (i.e., a root node) and destinations (i.e., leaf nodes that do not have any child nodes). Solving such a tree with optimal data aggregation is found to be NP-hard, even when the full topology is known [5].

Machine learning allows a sensor network to learn from previous experiences, make optimal routing actions and adapt to the dynamic environment. The benefits can be summarized as follows:

- Able to learn the optimal routing paths that will result in energy saving and prolonging the lifetime of dynamically changing WSNs.
- Reduce the complexity of a typical routing problem by dividing it into simpler sub-routing problems. In each sub-problem, nodes formulate the graph structures by considering only their local neighbors, thus achieving low cost, efficient and real-time routing.
- Meet QoS requirements in routing problem using relatively simple computational methods and classifiers.



(a) Original graph. (b) Traditional routing. (c) Simplified problems using machine learning.

Figure 5: An example of a sensor network routing problem using a graph along with each path routing cost, traditional spanning tree routing, and the generated sub-problems using machine learning that require only local communication to achieve optimal routing (i.e., require only single-hop neighborhood information exchange).

Figures 5a and 5b illustrate a simple sensor network routing problem using a graph, and the traditional spanning tree routing algorithm, respectively. To find the optimal routing paths, the network nodes have to exchange their routing information with each other. In the other side, Figure 5c demonstrates how machine learning reduces the complexity of a typical routing problem by only considering neighboring nodes' information that will be used to predict the full path quality. Each node will independently perform the routing procedures to decide which channels to assign, and the optimal transmission power. As we will discuss in this subsection, such mechanism is proven to provide a near optimal routing decision with a very low computational complexity.

In this subsection, a wide range of machine learning-based routing protocols developed for WSNs are described. Table I provides a summary and comparison of these routing protocols. The column "Scalability" implies the solutions' capability to route data in large scale networks.

1) Distributed regression framework: In [69], Guestrin *et al.* introduced a general framework for sensors data modeling. This distributed framework relies on the network nodes for fitting a global function to match their own measurement. The nodes are used to execute a kernel linear regression in the form of weighted components. Kernel functions map the training samples into some feature space to facilitate data manipulation (refer to [71], [72] for an introduction to kernel methods). The proposed framework exploits the fact that the readings of multiple sensors are highly correlated. This will minimize the communication overhead for detecting the structure of the sensor data. Collectively, these results serve as an important step in developing a distributed learning framework for wireless networks using linear regression methods. The main advantages of utilizing this algorithm are the good fitting results, and the small overhead of the learning phase. However, it cannot learn non-linear and complex functions.

Table I: Summary of wireless sensor network routing protocols that adopt machine learning paradigms.

ROUTING PROTOCOLS	TOPOLOGY	MACHINE LEARNING ALGORITHM(S)	OVERHEAD	SCALABILITY	DELAY	DISTRIBUTED / CENTRALIZED	QOS
Distributed regression [69]	Flat / multi-hop	kernel linear regression	Low	Limited	High	Distributed	No
SIR [70]	Flat / multi-hop	SOM	High	Limited	Moderate	Hybrid	Yes
Q-MAP multicast [65]	Flat /multi-hop	Q-learning	Low	Moderate	High	Distributed	No
RLGR [66]	Hierarchical / geographic routing	Q-learning	Low	Good	Low	Distributed	No
Q-Probabilistic [68]	Flat / geographic routing	Q-learning	Low	Limited	High	Distributed	Yes
FROMS [67]	Flat - multi-hop	Q-learning	High	Limited	Moderate	Distributed	No

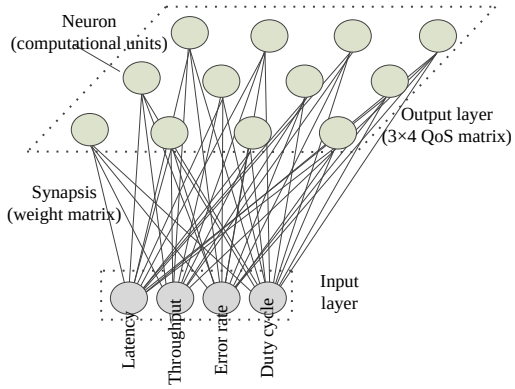


Figure 6: The SOM construction of the SIR algorithm, where routing link is selected based on the multi-hop path QoS metrics (latency, throughput, error rate, and duty cycle) and the Dijkstra's algorithm [70].

2) Data routing using self-organizing map (SOM): Barbancho *et al.* [70] introduced "Sensor Intelligence Routing" (SIR) by using SOM unsupervised learning to detect optimal routing paths as illustrated in Fig. 6. SIR introduces a slight modification on the Dijkstra's algorithm to form the network backbone and shortest paths from a base station to every node in the network. During route learning, the second layer neurons compete with each other to reserve high weights in the learning chain. Accordingly, the weights of the winning neuron and its neighboring neurons are updated to further match the input patterns. Clearly, the learning phase is a highly computational process due to the neural network generation task. As a result, it should be performed within a resourceful central station. However, the execution phase does not incur computational cost, and can be run on the network nodes. As a result, this hybrid technique (i.e., a combination of the Dijkstra's algorithm and the SOM model) takes into account the QoS requirements (latency, throughput, packet error rate, and duty cycle) during the process of updating neurons' weights. The main obstacles of applying such an algorithm are the complexity of the algorithm and the overhead of the learning phase in the case that the network's topology and setting change.

3) Routing enhancement using reinforcement learning (RL): In multicast routing, a node sends the same message to several receivers. Sun *et al.* [65] demonstrated the use of Q-learning algorithm to enhance multicast routing in wireless ad hoc networks. Basically, the Q-MAP multicast routing algorithm is designed to guarantee reliable resource allocation. A mobile ad hoc network may consist of heterogeneous nodes, where different nodes have different capabilities. In addition, it is not feasible to maintain a global, up-to-date knowledge about the whole network structure. The multicast routes are determined in two phases. The first phase is "Join Query Forward" that discovers an optimal route, as well as updates the Q-values (a prediction of future rewards) of the Q-learning algorithm. The second phase, called "Join Reply Backward", creates the optimal path to allow multicast transmissions. Using Q-learning for multicast routing in mobile ad hoc networks can reduce the overhead for route searching. However, energy efficiency is the key requirement for WSNs, so Q-MAP needs to be modified for WSNs (e.g., considering hierarchical and geographic routing).

The Federal Communications Commission (FCC) has dedicated the frequency band from 3.1 to 10.6 GHz (7,500 MHz of spectrum) for the use of unlicensed ultra-wideband (UWB) communication [73]. UWB is a technique for transmitting bulky data for short distances using a wide spectrum of frequency bands with relatively low power. In [66], Dong *et al.* used a similar idea as [65] to enhance geographic routing in UWB equipped sensor networks. "Reinforcement Learning based Geographic Routing" (RLGR) protocol considers the sensor node energy and delay as metrics for formulating the learning reward function. This hierarchical geographic routing uses the UWB technology for detecting the nodes' locations, where only the cluster heads are equipped with UWB devices. Moreover, each node uses a simple look-up table to maintain the information about its neighbors (as location and energy of the neighbors are needed during network learning). These information are exchanged between nodes using short "hello" messages to learn the best routing actions. The main benefit of using reinforcement learning in routing is that it does not require information about the global network structure to achieve an acceptable routing solution.

In [68], Arroyo-Valles *et al.* introduced "Q-Probabilistic Routing" (Q-PR), an enhanced geographic routing algorithm for WSNs that learns from previous routing decisions (e.g., to select the routing path that has the highest delivery rate over the past period of time). This protocol differs from RLGR [66] in the QoS support. Depending on the importance of messages, expected delivery rate, and the power constraints, Q-

PR determines the optimal routes using reinforcement learning and a Bayesian decision model. This algorithm discovers the next hop during the message routing time (i.e., an on-line opeartion). A Bayesian method is used to handle the decision of transmitting the packets to the set of candidate neighbor nodes, taking into account the data importance, nodes' profiles, and expected transmission and reception energy.

Förster and Murphy [67] also introduced an enhancement to routing in WSN using reinforcement learning. A novel technique for exchanging node local information as a feedback response to other nodes, named "Feedback Routing for Optimizing Multiple Sinks in WSN" (FROMS) is introduced. The main advantage of FROMS is to allow efficient routing from multiple sources to multiple sinks. The Q-values are initialized based on the hop counts to every node in the network. The hop counts can be collected using short "hello messages", exchanged between the nodes at earlier stages of the network deployment. FROMS extends the basic mechanism of RLGR [66] by assuming that all nodes can directly communicate with their neighbors.

The key disadvantage of reinforcement learning-based routing algorithms is the limited recognition of future knowledge (i.e., inability to look ahead). Therefore, the algorithms are not suitable for highly dynamic environments as they require a long time to learn optimal routes.

B. Clustering and Data Aggregation

In large scale energy-constrained sensor networks, it is inefficient to transmit all data directly to the sink [74]. One efficient solution is to pass the data to a local aggregator (known as a cluster head) which aggregates data from all the sensors within its cluster and transmits to the sink. This will typically result in energy savings. There are several works that have discussed the optimal selection of the cluster head (i.e., cluster head election process), such as in [75], [76], [77]. Taxonomy and comparison of classical clustering algorithms are presented in [78].

Figure 7 represents the cluster-based data aggregation from sources to a base station in WSNs. In this case, there could be some faulty nodes which must be removed from the network. Such faulty nodes may generate incorrect readings that could negatively affect the accuracy of the overall operation of the network. Principally, ML techniques improve the operation of node clustering and data aggregation as follows:

- Usage of machine learning to compress data locally at cluster heads by efficiently extracting similarity and dissimilarity (e.g., from faulty nodes) in different sensors' readings.
- Machine learning algorithms are employed to efficiently elect the cluster head, where appropriate cluster head selection will significantly reduce energy consumption and enhance the network's lifetime.

Table II compares data aggregation and node clustering solutions. The column "Balancing energy consumption" indicates whether the protocol distributes computationally intensive tasks into all nodes while considering the remaining energy information. The column "Topology aware" indicates the requirement for full network topology knowledge.

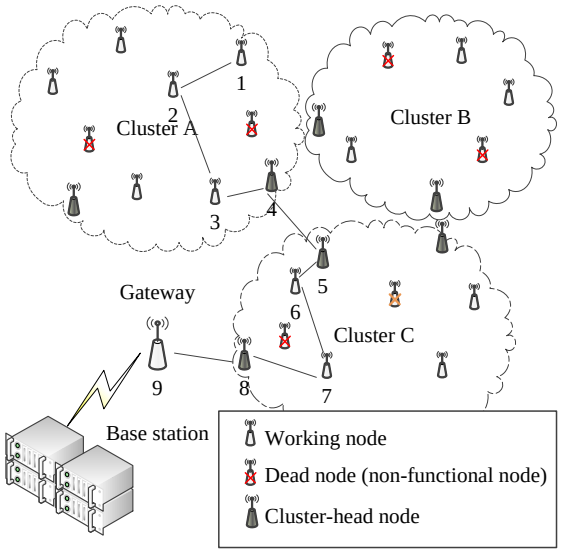


Figure 7: Data aggregation example in a clustered architecture, where the nodes are marked as working, dead and cluster heads.

1) Large scale network clustering using neural network: Hongmei *et al.* [79] discussed the development of self-managed clusters using neural networks. This scheme targets the clustering problem in large scale network with short transmissions radii in which centralized algorithms may not work efficiently. However, for large transmission radii, the performance of this algorithm is close to that of centralized algorithms in terms of efficiency and quality of service.

2) Electing a cluster head using decision trees: Ahmed *et al.* [80] applied a decision tree algorithm to solve the cluster head election problem. This approach uses several critical features while iterating the input vector through the decision tree such as distance to the cluster centroids, battery level, the degree of mobility, and the vulnerability indications. The simulation reveals that this scheme enhances the overall performance of cluster head selections when compared to the "Low Energy Adaptive Clustering Hierarchy" (LEACH) [87] algorithm.

3) Gaussian process models for sensor readings: Gaussian process (GP) is a combination of random variables (stochastic variables) that is parameterized using mean and covariance functions. Ertin [81] presented a scheme for initializing probabilistic models of the readings based on Gaussian process regression. Comparatively, Kho *et al.* [82] also extended Gaussian process regression to adaptively sample sensor data depending on its importance. Focusing on energy consumption, [82] studied a trade-off between computational cost and solution optimality. Broadly speaking, Gaussian process models are preferable in the problems with small training data sets (less than a few thousand samples) and for predicting smooth functions [53]. However, WSN designers must consider the high computational complexity of such methods when dealing with large scale networks.

4) Data aggregation using self-organizing map (SOM): The SOM algorithm is an unsupervised, competitive learning

Table II: Compassion of different machine learning-based data aggregation and node clustering mechanisms.

MECHANISMS	MACHINE LEARNING ALGORITHM(S)	COMPLEXITY	BALANCING ENERGY CONSUMPTION	DELAY	OVERHEAD	TOPOLOGY AWARE
Large scale network clustering [79]	NNs	Moderate	Yes	High	Low	Yes
Cluster head election [80]	DT	Low	Yes	Low	Low	Yes
Gaussian process models for censored sensor readings [81]	GPR	Moderate	No	Moderate	Moderate	No
Adaptive sampling [82]		High	Yes	High	High	No
Clustering using SOM and sink distance [54]	SOM	Moderate	No	High	Moderate	Yes
Online data compression [83]	LVQ	High	No	High	High	Yes
Data acquisition using compressive sensing [55], [56]	PCA	High	Yes	High	High	Yes
Transmission reduction [57]		Moderate	No	High	High	Yes
Consensus-based distributed PCA [58]		Moderate	Yes	High	High	No
Lossy data compression [84]		Moderate	No	Moderate	High	Yes
Collaborative signal processing [60]	k-means	Low	Yes	Moderate	Moderate	No
Advanced surveillance systems [59]		Moderate	Yes	Low	Low	Yes
Role-free clustering [85]	Q-learning	Low	No	Low	Low	No
Decentralized learning for data latency [86]	RL	Moderate	Yes	Low	Low	No

method for mapping from high dimensional spaces to low dimensions. Lee *et al.* [54] proposed a novel network architecture called "Cluster-based self-Organizing Data Aggregation" (CODA). In this architecture, the nodes are able to classify the aggregated data using a self-organizing algorithm. The winning neuron j^*, that has a weight vector $w(t)$ closest to the input vector $x(t)$, is defined as:

$$j^* = \arg\min_j \|x_j(t) - w_j(t)\|, j = 1, ..., N \tag{2}$$

where N represents the number of neurons in the second layer. Further, the winning node and its neighbors are updated as follows:

$$w_j(t+1) = w_j(t) + h(t)(x_j(t) - w_j(t)) \tag{3}$$

where $w(t)$ and $w(t+1)$ represent the values of a neuron at time t and $t+1$, respectively. In addition, $h(t)$ is the Gaussian neighborhood function given as:

$$h(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|j^* - j\|^2}{2\sigma^2(t)}\right). \tag{4}$$

Using CODA for data aggregation will result in enhancing the quality of data, saving network energy, and reducing the network traffic.

5) Applying learning vector quantization for online data compression: While the above methods require a complete knowledge about the network topology, some algorithms may not have such a restriction. For example, Lin *et al.* [83] introduced a technique called "Adaptive Learning Vector Quantization" (ALVQ) to accurately retrieve compressed versions of readings from the sensor nodes. Using data correlation and historical patterns, ALVQ uses the LVQ learning algorithm to predict the code-book using past training samples. The ALVQ algorithm minimizes the required bandwidth during transmission, and enhances the accuracy of original reading recovery from the compressed data.

The crucial disadvantage of using LVQ for online data aggregation is that dead neurons, that are far away from the training samples, will never take part in the competition. Therefore, it is important to develop algorithms that are robust against outliers. By the same token, LVQ is suitable for representing big data set by few vectors [43].

6) Data aggregation using principal component analysis: We begin by introducing two important algorithms that are efficiently used in combination with principal component analysis (PCA) to enhance data aggregation in WSNs.

- **Compressive sensing (CS)** has been recently explored to replace the traditional scheme of "sample then compress" with "sample while compressing". CS explores sparsity property of signals to recover the original signal from few random measurements. A simple introduction to CS is provided in [88].
- **Expectation-maximization (EM)** [89] is an iterative algorithm composed of two steps, i.e., an expectation (E) step and a maximization (M) step. During its E-step, EM formulates the cost function while fixing the current expectation of the system parameters. Subsequently, the M-step recomputes parameters that minimizes the estimation error of the cost function.

Masiero *et al.* [55], [56] developed a method for estimating distributed observations using few collected samples from a WSN. This solution is based on the PCA technique to produce orthogonal components used by compressive sensing to reconstruct the original readings. Moreover, this method is independent of the routing protocol due to its ability to estimate data spatial and temporal correlations. Similarly, Rooshenas *et al.* [57] applied PCA to optimize the direct transmission of readings to a base station. PCA results in considerable traffic reduction by combining nodes' collected data into fewer packets. This distributed technique is executed in intermediate nodes to combine all the incoming packets instead of forwarding them to destinations.

Equally important, Macua *et al.* [58] introduced distributed consensus-based methods for data compression using PCA and maximum likelihood of the observed data. These methods are "Consensus-based Distributed PCA" (CB-DPCA) which relies on exploring the eigenvectors of local covariance ma-

trices, and "Consensus-based EM Distributed PCA" (CB-EM-DPCA). The latter uses a distributed EM algorithm. These methods adopt the consensus algorithm [90] to predict the probability distribution of the data, and hence calculate the global dominant eigenvectors using only local communication parameters (i.e., single hop communications). CB-DPCA and CB-EM-DPCA can be tuned to provide a trade-off between the achieved approximation quality and the communication cost by adjusting the consensus round parameter. For example, to increase the algorithm accuracy, the number of consensus rounds should be increased which will increase the computational requirements of the algorithm.

Recently, Fenxiong *et al.* [84] have tackled the problem of data compression using PCA by transforming the data from a high dimensional space to a lower one. The data is collected over time, and then it is transmitted from each node to its corresponding cluster head. At the cluster head, the data matrix is compressed to eliminate the data redundancy. The data compression is achieved by ignoring principal components through which the data has the least variation values.

Energy saving and high computational requirements are the main issues of PCA-based data aggregation solutions. Other than increasing throughput, these solutions elegantly cope with the high dimensionality of collected data by keeping only important information (data dimensionality reduction).

7) Collaborative data processing through k-means algorithm: Li *et al.* [60] addressed the fundamental concepts for distributed detection and tracking of a single target using sensor networks. "Collaborative Signal Processing" (CSP) is a framework for information gathering from the monitored environment. Additionally, this algorithm can track multiple targets using classification techniques such as SVM and k-nearest neighbors.

Classical surveillance systems have to collect massive data from surveillance cameras. Together with the requirement of highly complex computation and analysis process, this introduces the need for more practical methods. Therefore, Tseng *et al.* [59] proposed "Integrated Mobile Surveillance and Wireless Sensor System" (iMouse) which adopts powerful mobile sensors to enhance traditional surveillance systems. iMouse divides the monitored sites into a number of clusters using the k-means unsupervised learning algorithm. Each cluster will be repeatedly monitored by only one mobile sensor.

Although these ideas (using k-means for data processing) are appealing because of the straightforward implementations and low complexity, they are still sensitive to outliers and to initial seed selections.

8) Role-free clustering: In [85], Förster and Murphy introduced the WSN cluster formulation method called "Role-Free Clustering with Q-Learning for Wireless Sensor Networks" (CLIQUE). Instead of performing an election process, CLIQUE enables each node to investigate its ability to function as a cluster-head node. This is achieved through the use of Q-learning algorithm in combination with some dynamic network parameters such as energy levels.

9) Decentralized learning for data latency: Mihaylov *et al.* [86] addressed the problem of high data latency in random topology sensor networks using reinforcement learning. Each node executes the learning algorithm locally to optimize the data aggregation without the need for a central control station. Consequently, the efficiency of the whole network is enhanced with smaller learning transmission overhead. The approach saves the node energy budget during data gathering process, and hence prolong the network lifetime.

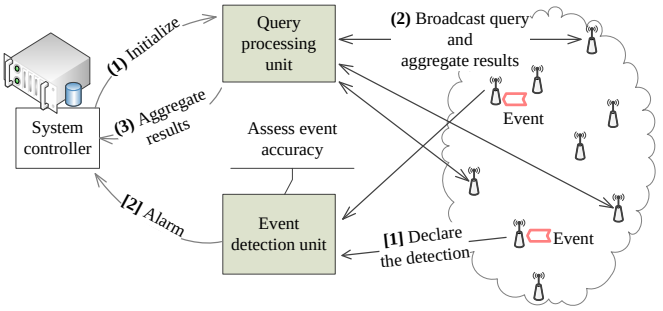


Figure 8: Event detection and query processing enhancement using machine learning methods by assessing event validity and delimiting queried areas. System controller initiates query that is spread by the query processing unit to intended nodes. In contrast, events are detected by nodes to monitor specific signs within the monitored area.

C. Event Detection and Query Processing

Event detection and query processing are considered to be functional requirements of any large scale sensor network. This introduces the need for trustworthy event scheduling and detection with minimal human intervention. Monitoring in WSNs can be classified as: event-driven, continuous, or query-driven [6]. Figure 8 illustrates event detection and query processing operations in WSNs. Fundamentally, machine learning offers solutions to restrict query areas and assess event validity for efficient event detection and query processing mechanisms. This adoption will result in the following benefits:

- Learning algorithms enable the development of efficient event detection mechanisms with limited requirements of storage and computing resources. Besides they are able to assess the accuracy of such events using simple classifiers.
- Machine learning facilitates the development of effective query processing techniques for WSNs, that determine the search regions whenever a query is received without flooding the whole network.

The design of effective event detection and query processing solutions has recently received increased attention from WSNs research community. The simplest techniques rely on defining a strict threshold value for the sensed phenomenon and alarming the system manager of any violations. However, in most recent applications of WSNs, event and query processing units are often complicated and require more than a predefined threshold value. One such emerging technique is to use machine learning to develop advanced event detection and query processing solutions. Table III presents a comparison of functional aspects of different machine learning-based event detection and query processing solutions for WSNs.

Table III: Comparison of functional aspects of different machine learning-based event detection and query processing solutions for WSNs.

Approaches	Machine learning algorithm(s)	Data Delivery Models	Complexity	Characteristics
Event region detection [91]	Bayesian	Event-driven	Low	Fault-tolerant event region detection
Activity recognition [92]			Moderate	Real-time activity recognition
In-network query processing [24]	k-NN	Query-driven	Low	In-network query processing
Query processing in 3D space [25]			Moderate	Enhance 3D space query processing
Forest fire detection [26]	NNs	Event-driven	Moderate	Real-time and lightweight forest fire detection
Distributed event detection [27]	DT	Event-driven	Low	Disaster prevention system
Query optimization [93]	PCA	Query-driven	High	Query optimization and dimensionality reduction

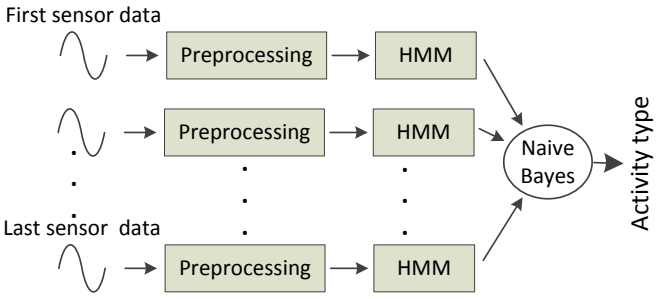


Figure 9: Human activity recognition using the hidden Markov model and the naive Bayes classifier [92].

1) Event recognition through Bayesian algorithms: Krishnamachari and Iyengar [91] investigated the use of WSNs for detecting environmental phenomenon in a distributed manner. Readings will be considered as faulty if their values exceed a specific threshold. This study employs decentralized Bayesian learning that detects up to 95 percent of the faults, and will result in recognizing the event region. It is important to note that Chen *et al.* [94] provided corrections to several errors related to the distributed Bayesian algorithms that have been derived in [91]. In summary, these corrections result in enhanced error and performance calculations for the distributed Bayesian algorithm proposed in [91].

Additionally, Zappi *et al.* [92] presented a real-time approach for activity recognition using WSNs that accurately detects body gesture and motion. Initially, the nodes, that are spread throughout the body, detect the organ motion using an accelerometer sensor with three axis measurements (positive, negative and null), where these measurements are used by a hidden Markov model (HMM) to predict the activity at each sensor. Sensor activation and selection rely on the sensor's potential contributions in classifier accuracy (i.e., select the sensors that provide the most informative description of the gesture). To generate a final gesture decision, a naive Bayes classifier is used to combine the independent node predictions so as to maximize the posterior probability of the Bayes theorem. The architecture of the proposed system is shown in Fig. 9.

2) Forest fire detection through neural network: WSNs were actively used in fire detection and rescue systems (see [95] and references therein for requirements and challenges of such systems). Moreover, the use of WSNs for forest fire detection can achieve better performance than using satellite-based solutions while costing much less. Yu *et al.* [26] presented a real-time forest fire detection scheme based on a neural network method. Data processing will be distributed to cluster heads, and only important information will be aggregated to a final decision maker. Although the idea is creative and beneficial to the environment, the classification task and system core are hardly interpretable when introducing such systems to decision makers.

3) Query processing through k-nearest neighbors: K-nearest neighbor query is considered as a highly effective query processing technique in WSNs. For example, Winter *et al.* [24] developed an in-network query processing solution using the k-nearest neighbor algorithm, namely the "K-NN Boundary Tree" (KBT) algorithm. Each node that is aware of its location will determine its k-NN search region whenever a query is received from the application manager.

Correspondingly, Jayaraman *et al.* [25] extended the query processing design of [24]. "3D-KNN" is a query processing scheme for WSNs that adopts the k-nearest neighbor algorithm. This approach restricts the query region to bound at least k-nearest nodes deployed within a 3D space. In addition, signal-to-noise ratio (SNR) and distance measurements are used to refine the k-nearest neighbor.

The primary concerns of such k-NN-based algorithms for query processing are the requirement of large memory footprint to store every collected sample and the high processing delay in large scale sensor networks.

4) Distributed event detection for disaster management using decision tree: Bahrepour *et al.* [27] developed decision tree-based event detection and recognition for sensor network disaster prevention systems. The main application of this decentralized mechanism is the fire detection in residential areas. Most noteworthy, the final event detection decision is made by using a simple vote from the highest reputation nodes.

5) Query optimization using principal component analysis (PCA): Malik *et al.* [93] optimized traditional query processing in WSNs using data attributes and PCA, thus reducing the overhead of such a process. PCA has been used to dynamically detect important attributes (i.e., dominant principal components) among the whole correlated data set. Figure 10 shows the workflow of the proposed algorithm in four fundamental

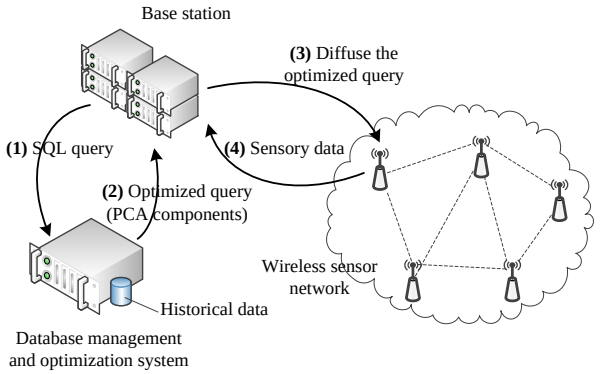


Figure 10: Workflow of the query optimization and reduction system using PCA proposed in [93]

steps. In Step 1, the structured query language (SQL) request, which contains the human intelligible attributes, is sent to the database management and optimization system. At the database management and optimization system, the original query is optimized where the high-variance components are extracted from historical data using the PCA algorithm (Step 2). Then, the optimized query is diffused to the wireless sensor network to extract the sensory data as shown in Steps 3 and 4, respectively. Later, the original attributes (i.e., human intelligible attributes) can be extracted from the optimized attributes by reversing the process of PCA.

As a result, this algorithm guarantees 25 percent improvement in energy saving of the network nodes while achieving 93 percent of accuracy rates. However, this enhancement is at the cost of accuracy of the collected data (as some of the data components will be ignored). Therefore, this solution may not be ideal for the applications with high accuracy and precision requirements.

D. Localization and Objects Targeting

Localization is the process of determining the geographic coordinates of network's nodes and components. Position awareness of sensor nodes is an important capability, since most sensor network operations are typically based on the location [96]. In most large scale systems, it is financially infeasible to use global positioning system (GPS) hardware in each node for this purpose. Moreover, GPS service may not be available in the observed environment (e.g., indoor). Relative location measurement is sufficient for certain uses. However, by using the absolute locations for a small group of nodes, relative locations can be transformed into absolute ones [97]. In order to enhance the performance of proximity based localization, additional measurements relying on distance, angle or a hybrid of them can be used. Distance measurements can be obtained by utilizing various techniques such as RSSI, TOA, and TDOA. Furthermore, angle of the received signal can be measured using compasses or special smart antennas [98]. A valuable introduction about the basics of different range-based localization techniques is provided in [42].

Sensor nodes may encounter changes in their location after deployment (e.g., due to movement). The benefits of

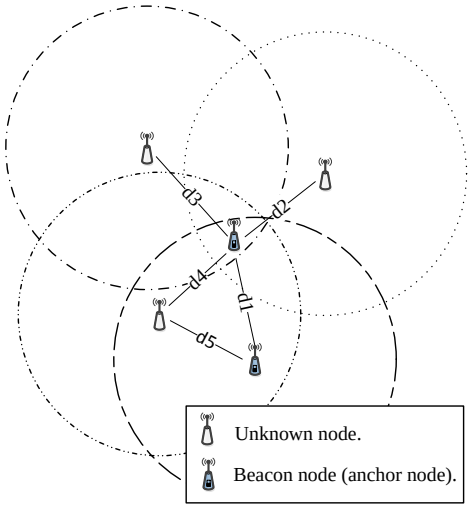


Figure 11: Localization using few beacon nodes by utilizing machine learning algorithms and other signal strength indicators (reformulated from [96]).

using machine learning algorithms in sensor node localization process can be summarized as follows:

- Converting the relative locations of nodes to absolute ones using few anchor points. This will eliminate the need for range measurement hardware to obtain distance estimations.
- In surveillance and object targeting systems, machine learning can be used to divide the monitored sites into a number of clusters, where each cluster represents specific location indicator.

We begin by defining some terms that are widely used in WSN localization literature, as illustrated in Figure 11.

- **Unknown node** is a node that cannot determine its current location.
- **Beacon node (or anchor node)** is any node that is able to recognize its location by using positioning hardware or from its manual placement. In most systems, the beacon node is used as a reference point to estimate the coordinates of other unknown nodes.
- **Received signal strength indication (RSSI)** is an indicator of the received signal strength, used to represent transmission performance or distance.

Next, we discuss some seminal WSN localization techniques that use machine learning and summarize our reviews in Table IV.

1) Bayesian node localization: Morelande *et al.* [21] used a Bayesian algorithm to develop a localization scheme for WSNs using only few anchor points. This study focuses on the enhancement of progressive correction [109], which is a method for predicting samples from likelihoods to get closer to the posterior likelihood. The proposed algorithm is efficiently applicable for node localization in large scale systems (i.e., networks with a few thousands of nodes). The idea of using the Bayesian algorithm for localization is appealing as it can handle incomplete data sets by investigating prior knowledge and probabilities.

Table IV: Summary of localization algorithms in WSNs that adopt machine learning concepts and their prime advantages. The column "Applications" specifies the targeted application(s) of the proposed solution (either general-purpose or a specific application).

Studies description	Machine learning algorithm(s)	Complexity	Distributed / centralized	Beacon / Beacon-less	Applications
Bayesian node localization [21]	Bayesian	Moderate	Centralized	Beacon	General-purpose
Location-aware activity recognition [22]		Moderate	Centralized	Beacon	Smart homes
Localization based on NNs [23]	NNs	High	Centralized	Beacon	General-purpose
Soft localization [99]		Moderate	Distributed	Beacon	General-purpose
Localization based on NNs [100]		High	Distributed	Beacon	General-purpose
Area localization [51]	SVM	Moderate	Distributed	Beacon	General-purpose
Localization using SVM [50]		Moderate	Distributed	Beacon	General-purpose
Localization using SVR [49]	SVR	Moderate	Distributed	Beacon	General-purpose
Target classification and information fusion [101]	DT	Low	Distributed	Beacon	Acoustic WSNs
Underwater surveillance system [102]		Moderate	Centralized	Beacon	Submarine surveillance
Sensor placements [103]	GP	Low	Distributed	Beacon	Node deployment
Spatial Gaussian process regression [104]	GPR	Moderate	Distributed	Beacon	Collective node motion
Localization in 2D space [105]	SOM	Moderate	Distributed	Beacon	Large scale WSNs
Localization using SOM [106]		Low	Centralized	Beacon-less	General-purpose
Distributed localization [107]		Moderate	Distributed	Beacon-less	General-purpose
Path determination [108]	RL	Low	Distributed	Beacon	Mobile nodes

2) Robust location-aware activity recognition: Lu and Fu [22] addressed the problem of sensor and activity localization in smart homes. The activities of interest include using the phone, listening to the music, using the refrigerator, studying, etc. In such applications, designers need to comply with both human and environment constraints in a convenient and easily operated way. The proposed framework, named "Ambient Intelligence Compliant Object" (AICO), facilities the human interaction with the home electric devices in a more intelligent manner (e.g., automatic power supply management). At its core, AICO uses multiple naive Bayes classifiers to determine the resident's current location and evaluate the reliability of the system by detecting any malfunctioned sensors. Although this method provides a robust mechanism for localization, it is still application-dependent and the designers must predefine a set of supported activities in advance. This is because the used learning features are selected and evaluated manually depending on the activities and the domain of interest. To overcome this limitation in this centralized system, we recommend investigating unsupervised machine learning algorithms for automatic feature extraction such as the deep learning methods [9] and the non-negative matrix factorization algorithm [110].

3) Localization based on neural network: Shareef *et al.* [23] compared three localization schemes that are based on different types of neural networks. In particular, this study considers WSN localization using multi-layer perceptron (MLP), radial basis function (RBF), and recurrent neural networks (RNN). In summary, the RBF neural network results in the minimum error at the cost of high resource requirements. In contrast, MLP consumes the minimum computational and memory resources.

Likewise, Yun *et al.* [99] adopted a similar design, in which two classes of algorithms for sensor node localization using RSSI from anchor nodes are proposed. The first class utilizes the fuzzy logic system and genetic algorithm. In the second class, the neural network is adopted to predict the sensor location by using RSSI measurements from all anchor nodes as an input vector. In the same way, Chagas *et al.* [100] applied neural networks for WSNs localization with RSSI as an input to the learning network.

The main advantage of these NNs-based localization algorithms is their ability to provide coordinates in the form of continuous-valued vectors (e.g., coordinates in 3D space). However, unlike statistical or Bayesian alternatives, neural network is a non-probabilistic method. This fact limits the designers' certainty about precision of unknown node's predicted coordinates, and hence restricts their ability to manage the cost of localization errors.

4) Localization using support vector machine (SVM): The SVM technique has been widely used for node localization in WSNs, where having a self-positioning device to each sensor is infeasible. As an illustration, Yang *et al.* [51] developed a mobile node localization scheme by employing SVM and connectivity information capabilities. In its initial step, the proposed method has to detect node movement using their radio frequency oscillation such as RSSI metric. For movement detection, SVM will be executed to provide the new location.

Similar to [51], Tran and Nguyen [50] proposed "Localization Based on Support Vector Machines" (LSVM) method for node localization in WSNs. To achieve its design goals and given an appropriate training data, LSVM adopts several decision metrics such as connectivity information and indicators. Even though LSVM offers distributed localization in a fast and effective manner, its performance is still sensitive to outliers in training samples.

5) Localization using support vector regression (SVR): Limited resources and high data dimensionality impede the wide adoption of SVR learning in WSNs. Therefore, Kim *et al.* [49] developed the idea of using lightweight implementation of SVR by dividing the original regression problem into several

sub-problems. Basically, the algorithm starts by dividing the network into a set of sub-networks, thus a small number of data has to be processed by each regression algorithm (i.e., SVR's sub-predictors). Then, the learned hypothesis models of the sub-predictors are combined together using a customized ensemble combination technique. Thus, in addition to its low computational requirements and robustness against noisy data, this solution converges to the preferred solution with low computational requirement.

6) Decision tree-based localization: Based on decision tree learning, Merhi *et al.* [101] developed an acoustic target localization method for WSNs. Exact locations of targets are determined using the time difference of arrival (TDOA) metric in a spatial correlation decision tree. Also, this work proposed the design of "Event Based MAC" (EB-MAC) protocol, that enables event-based localization and targeting in acoustic WSNs. The proposed framework was implemented using a MicaZ board that supports ZigBee 802.15.4 specifications for personal area networks.

Using the GPS functionality to support localization in underwater wireless sensor network's applications may not be feasible due to the propagation limitation of the GPS signal through water [111]. Erdal *et al.* [102] developed a system for submarine detection in underwater surveillance systems, so that a randomly deployed node finds its location in the 3D space based on beacon node coordinates. Each monitoring unit consists of a sensor that is fixed with a cable to a surface buoy. Data is collected using the buoys, where they are transmitted to the central processing unit. At the central unit, a decision tree classifier is used to recognize any submarines in the monitored sites.

7) Sensor placements through Gaussian processes: Krause *et al.* [103] provided an optimized solution to sensor placement in applications with spatially correlated data such as temperature monitoring systems. One interesting feature of this solution is the development of a lazy learning scheme based on Gaussian process model for the investigated phenomenon. Lazy learning algorithms store training samples and delay the major processing task until a classification request is received. Moreover, this solution aims to achieve robustness against node failures and model ambiguity when choosing optimal locations for sensors.

8) Spatial Gaussian process regression: Gu and Hu [104] developed a distributed protocol for collective node motion. This approach employs distributed Gaussian process regression (DGPR) to predict optimal locations for mobile nodes' movements. Traditional Gaussian process regression (GPR) algorithms have computational complexity of $\mathcal{O}(N^3)$, where N is the size of samples. However, this solution adopts a sparse Gaussian process regression algorithm to reduce such computational complexity. Each node will execute the regression algorithm independently using only spatiotemporal information from local neighbors.

9) Localization using self-organizing map (SOM): Given some anchor positions, Paladina *et al.* [105] introduced the SOM-based positioning solution for WSNs consisting of thousands of nodes. The proposed scheme is executed in each node with a simple SOM algorithm that consists of a 3x3 input layer connected to the 2 neurons of the output layer. In particular, the input layer is formulated using the spatial coordinates of 8 anchor nodes surrounding the unknown node. After a sufficient training, the output layer is used to represent the unknown node's spatial coordinates in a 2D space. The main disadvantage of this scheme is that the nodes should be distributed uniformly and equally spaced throughout the monitored area.

Unlike traditional methods that require absolute locations of a few nodes to find the positions of the unknown nodes, Giorgetti *et al.* [106] introduced a localization algorithm that is only based on connectivity information and the SOM algorithm. The developed method is highly suitable for networks with limited resources, as it does not require a GPS-enabled device. However, since this is a centralized algorithm, each node transmits the information of its neighbors to the central processing unit to determine the adjacency matrix and hence the node's location. Similarly, Hu and Lee [107] presented a scheme that provides node localization service in WSNs without the need for anchor nodes. The algorithm is based on SOM, and it operates efficiently for any number of nodes. The contribution of [107] over [106] is that the proposed algorithm distributes the computation tasks to all nodes in the network, which eliminates the needs for a central unit and minimizes the transmission overhead of the algorithm.

10) Path determination using reinforcement learning: Li *et al.* [108] developed a reinforcement learning-based localization method for WSNs, called "Dynamic Path determination of Mobile Beacons" (DPMB), suitable for real-time management of the mobile beacons. The mobile beacon (MB), which is aware of the physical location during its movement, will be used to determine the positions of large number of sensor nodes. In brief, the states of the Q-learning algorithm are used to represent the different positions of the MB, and the algorithm target is to cover all the sensors in the monitored area (i.e., all the sensors should hear a location update message from the MB at some stages). The entire operation will be run in the mobile beacon, and hence, this will save the resources of the unknown nodes. However, as a centralized method, the entire system will fail in the event of mobile beacon malfunctions.

E. Medium Access Control (MAC)

In WSNs, a number of sensors cooperate to efficiently transfer data. Therefore, designing MAC protocols for WSNs poses different challenges from typical wireless networks, as well as energy consumption and latency [112]. Also, the duty cycle (i.e., fraction of time that a sensor node is active) of the node has to be controlled to conserve energy. Therefore, the MAC protocols have to be modified to support efficient data transmission and reception of the sensor nodes. A comprehensive survey of MAC protocols in WSNs is provided in [113].

Recently, machine learning methods have been used to enhance the performance of MAC protocols in WSNs. Specifically, this is achieved through the following points:

- Machine learning can be used to adaptively determine the duty cycle of a node using the transmission history

of the network. In particular, the nodes, which are able to predict when the other nodes' transmissions will finish, can sleep in the meantime and wake up (to transmit data) just when the channel is expected to be idle (i.e., when no other node is transmitting). For WSNs, many factors, such as energy consumption and latency, are more important than fairness when designing MAC protocols.

- Achieving secured data transmission by combining the concepts of machine learning and MAC protocols. Such MAC layer security schemes are independent of the proposed application and are able to iteratively learn sporadic attack patterns.

Table V gives a brief comparison between MAC protocols reviewed in this subsection. The column "Synchronization" indicates whether the protocol assumes that time synchronization is achieved externally, and "Adaptivity to changes" indicates the ability to handle topology changes such as nodes failure.

1) Bayesian statistical model for MAC: Kim and Park [28] presented a contention-based MAC protocol for managing active and sleep times in WSNs. Instead of continuously sensing the medium, this scheme utilizes a Bayesian statistical model to learn when the channel can be allocated, and hence save network energy. Furthermore, in its basic design, this scheme is targeted for CSMA contention-based protocols such as "Sensor MAC" (S-MAC) [116], and "Timeout MAC" (T-MAC) [117].

2) Neural network-based MAC: Time division multiple access (TDMA)-based protocols employ periodic time frames to separate the medium access of different nodes. This process requires a central unit to broadcast a transmission schedule in case of topology changes. Shen and Wang [29] proposed a solution to broadcast the transmission schedule in TDMA using a fuzzy hopfield neural network (FHNN) technique. Time slots are distributed among the nodes in a network while maximizing the cycle length, preventing any potential transmission collisions and reducing the processing time.

In the same way, Kulkarni and Venayagamoorthy [30] presented an innovative CSMA-based MAC solution, that can prevent denial-of-service (DoS) attacks in WSNs. Denial-of-service is a type of attacks that generates huge useless traffic (i.e., flood the network), thus preventing the delivery of useful data. In such cases, attackers exploit the limitations of WSNs such as limited bandwidth and buffering capabilities. The proposed solution is based on neural network learning to prevent flooding the network with untruthful data traffic by investigating the network properties and variations such as packet request rate and average packet waiting time. Consequently, the MAC layer will be blocked if the neural network output exceeds a predefined threshold level. More importantly, only nodes in the affected sites will be blocked, as this solution is designed to work in a distributed manner.

3) Duty cycle management using reinforcement learning: Liu and Elhanany [114] employed a reinforcement learning technique to introduce RL-MAC, an adaptive MAC protocol for WSNs. Basically, RL-MAC reduces energy usage and increases throughput by optimizing the duty cycle of the network node. Similar to S-MAC [116] and T-MAC [117], RL-MAC synchronizes node's transmission on a common schedule in a frame-based structure. RL-MAC adaptively determines the slot length, duty cycle and transmission active time depending on the traffic load and the channel bandwidth.

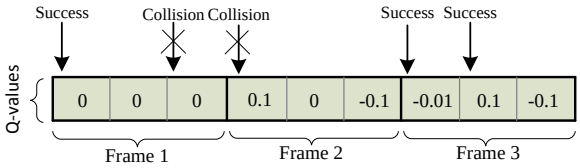


Figure 12: An example of the Q-values of a node over three frames in a WSN that employs ALOHA-QIR to manage medium access [114].

Similarly, Chu *et al.* [112] integrated slotted ALOHA and Q-Learning algorithms to introduce a new MAC protocol for WSNs, called "ALOHA and Q-Learning based MAC with Informed Receiving" (ALOHA-QIR). ALOHA-QIR inherits the features of both ALOHA and Q-Learning to achieve the benefits of simple design, low resource requirements and low collision probability. During their transmission frames, nodes broadcast their future transmission allocation such that other nodes can sleep during reserved frame. The Q-value map in each node represents the willingness for slot reservation, where the node with higher Q-value will attain the right of slot allocation and hence transmission of its own data. Figure 12 demonstrates the steps of updating the Q-values over three frames of a node that is allowed to transmit a maximum of two packets in each frame. Initially, the Q-values are initialized to zero, i.e., $Q\,(frame\#0) = \{0, 0, 0\}$. Upon successful transmission, the Q-value of each time slot is updated using the Q-learning update rule given by Eq. (1) where the learning rate is set to 0.1. Upon successful transmission, the reward value is equal to +1, and it is -1 for a failed transmission. Certainly, the nodes will decide to transmit data using the time slots with the maximum Q-values.

Although the idea of using reinforcement learning for duty cycle management is appealing because of its distributed operation and small memory and computational resource requirement, it may result in high collision rates during the initial exploration phases.

4) Adaptive MAC layer: In a variety of modern applications, such as in healthcare and assisted living systems, WSNs are used to directly share the collected data with the users' mobile phones. This introduces new design challenges that are related to the dynamic communication patterns and service requirements over time. Sha *et al.* [115] studied this problem at the MAC layer, hence proposing the "Self-Adapting MAC Layer" (SAML) design. SAML is composed of two main components: The "Reconfigurable MAC Architecture" (RMA) to switch between the different MAC protocols, and the MAC engine that is used to learn the suitable MAC protocol for the current network conditions. The learning process is performed using the decision tree classier as illustrated in Fig. 13. The learning features of the decision tree are: inter packet interval (IPI) and received signal strength indication (RSSI) statistical parameters (i.e., the mean and the variance), the application QoS requirements (reliability, energy usage,

Table V: Comparison of MAC protocols.

MAC PROTOCOLS	MACHINE LEARNING ALGORITHM(S)	COMPLEXITY	CATEGORY	TYPE	ADAPTIVITY TO CHANGES	SYNCHRONIZATION
Bayesian Statistical Modeling [28]	Bayesian	Moderate	Contention–Based	CSMA	Good	No
Broadcast scheduling [29]	NNs	Low	Contention–Free	TDMA	Weak	Yes
NN-based secure MAC [30]		Moderate	Contention–Based	CSMA/CA	Average	No
RL-MAC [114]	RL	Moderate	Contention–Based	CSMA	Good	Yes
ALOHA-QIR [112]		Low	Contention–Free	Slotted Aloha	Weak	Yes
Adaptive MAC layer [115]	DT	High	Hybrid	Hybrid	Good	Yes

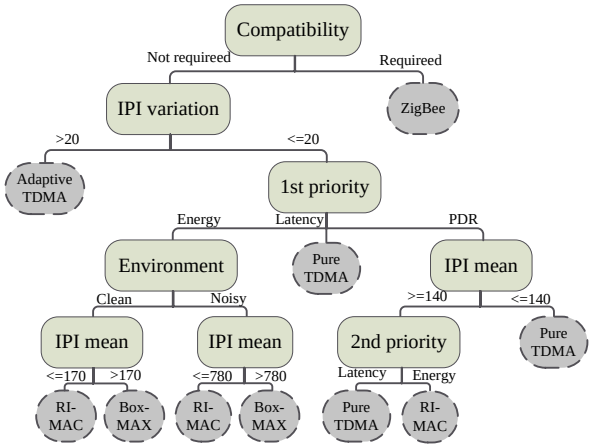


Figure 13: The decision tree classifier used to select the optimal MAC algorithm in the SAML architecture [115].

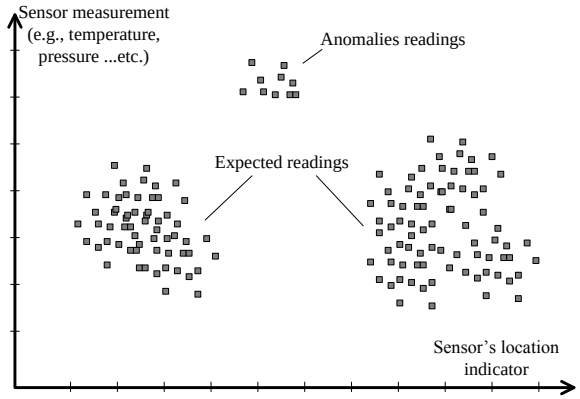


Figure 14: An example of anomaly detection in phenomena monitoring sensor system using machine learning clustering and classification techniques (data set in euclidean space).

and latency), packet delivery rate (PDR), and the traffic pattern. The supported MAC protocols are Pure TDMA [118], Adaptive TDMA [119], Box-MAC [120], RI-MAC [121], and ZigBee [122]. Even though the SAML scheme provides an adaptive MAC solution in dynamic environments, it introduces a level of complexity and additional expense into the designed systems.

IV. NON-FUNCTIONAL CHALLENGES

Non-functional requirements include specifications that are not related to the basic operational behavior of the system. For example, WSN designers may need to ensure that the proposed solution is always capable of providing up-to-date information about the monitored environment. This section provides a comprehensive review of recent machine learning advances that have been adopted to achieve non-functional requirements in WSNs such as security, quality of service, and data integrity. Moreover, this section also highlights some unique efforts in specialized WSN applications. Such studies could inspire researchers to a variety of WSN applications that can be improved using machine learning techniques.

A. Security and Anomaly Intrusion Detection

The major challenge to implement security techniques in WSNs is the limited resource constraints [14]. Moreover, some attack methods aim to produce unexpected, mistaken knowledge, by introducing misleading observations to the network. Figure 14 presents the general concept of anomaly detection in phenomenon monitoring sensor system using machine learning clustering and classification algorithms. In this example, machine learning techniques classify the data into two correct reading regions. Since most observations lie in these two regions, the points that are inconsistent (e.g., from an attack) with these regions are considered as anomalies.

Machine learning algorithms have been employed to detect outlying and misleading measurements. Simultaneously, several attacks could be detected by analyzing well-known malicious activities and vulnerabilities. Basically, WSN security enhancements by adopting machine learning techniques will result in the following earnings:

- Save node's energy and significantly expand WSN lifetime by preventing the transmission of the outlier, misleading data.
- Enhance network reliability by eliminating faulty and malicious readings. In the same way, avoiding the discovery of unexpected knowledge that will be converted to important, and often critical actions.
- Online learning and prevention (without human intervention) of malicious attacks and vulnerabilities.

In this subsection, we explore various machine learning-based algorithms addressing the security issue in WSNs. Table VI summarizes the reviewed methods in this subsection. The column "Predicting missing data" indicates the ability of the proposed solution to provide predictions for any missing sensors' readings.

Table VI: Summary of wireless sensor network outlier detection techniques that adopt machine learning paradigms.

STUDIES	MACHINE LEARNING ALGORITHM(S)	PREDICTING MISSING DATA	DISTRIBUTED / CENTRALIZED	COMPLEXITY	AIM(S)
Outlier detection using Bayesian belief networks [31]	Bayesian	Yes	Distributed	Low	Outlier detection
Outlier detection using k-NN [32]	k-NN	Yes	Distributed	Moderate	Distributed outlier detection
Detecting selective forwarding attacks using SVM [33]	SVM	No	Centralized	Moderate	Detect black hole and selective forwarding attacks
Distributed outlier detection using SVM [34]			Distributed	Low	Outlier detection
Online outlier detection [46]			Centralized	Moderate	Online outlier detection
Intrusion detection system [47]			Centralized	High	Intrusion detection using immune and SVM algorithms
Linear outlier detection [48]			Distributed	Moderate	Adaptive outlier detection
Analyzing attacks with SOM [123]	SOM	No	Distributed	Moderate	Detect anomalous behaviors

1) Outlier detection using Bayesian belief network: Janakiram *et al.* [31] used Bayesian belief networks (BBNs) to develop an outlier detection scheme. Given that the majority of node's neighbors will have similar readings (i.e., temporal and spatial correlations), it is reasonable to use this phenomenon to build conditional dependencies among nodes' readings. BBNs infer the conditional relationships among the observations to discover any potential outliers in the collected data. Furthermore, this method can be used to evaluate missing values.

2) Outlier detection using k-nearest neighbors: Branch *et al.* [32] developed an in-network outlier detection method in WSNs using k-nearest neighbors. Moreover, any missing nodes' readings will be replaced by the average value of the k-nearest nodes. However, such non-parametric, k-NN-based algorithm requires large memory to store every collected readings from the monitored environment.

3) Detecting selective forwarding attacks using support vector machine (SVM): In black hole attacks, malicious nodes send misleading "Routing Reply" (RREP) messages whenever the malicious nodes receive "Route Request" (RREQ) messages, indicating that routes to the destinations are found. Accordingly, source nodes will stop the process of route discovery, and will ignore other RREP messages. Therefore, malicious nodes will drop all network's messages, while the source nodes assume that their packets were delivered to the destination. Kaplantzis *et al.* [33] presented packet dropping attack prevention technique based on one class support vector machine classifier. The proposed scheme is capable of detecting black hole attacks and selective forwarding attacks. Basically, routing information, bandwidth and hop count are used to determine the malicious nodes in the network.

4) Outlier detection using support vector machine (SVM): By using a quarter-sphere centered at the origin, the drawback of high computational requirements of traditional SVM could be alleviated. For instance, Rajasegarar *et al.* [34] introduced a one-class quarter-sphere SVM anomaly recognition technique. The motivation of this distributed scheme is to distinguish anomalies in data while minimizing communication overhead. In [46], Yang *et al.* tackled the design of an online outlier detection method using quarter-sphere SVM. The unsupervised learning method investigates the local data to reduce the computational complexity of traditional SVM-based outlier detection algorithms. This outlier detector is similar to the method introduced in [34].

Artificial immunity algorithm is a computationally intelligent algorithm for problem solving inspired by the biological immunity systems [124]. The biological immunity systems automatically generate the immune body (antibody) against the antigen (e.g., a virus) through the cell fission. In [47], Chen *et al.* extended the basic idea of using SVM for detecting intrusion by combining it with immunity algorithm. In summary, an immune algorithm was introduced as a preprocessing step for the sensor data, that will be used by SVM to detect intruders. Furthermore, Zhang *et al.* [48] also investigated the temporal and spatial correlations of the collected readings using a one-class SVM learning algorithm to develop an outlier detection method. This study adopts an ellipsoidal one-class SVM that can be solved using linear optimization instead of the quadratic optimization problem in traditional SVM methods.

The main advantages of these SVM-based methods are their good performance (efficient learning) and ability to learn non-linear and complex problems. However, they still suffer from a scalability issue to large data set due to their high computational and large memory requirements [45].

5) Analyzing attacks with self-organizing map (SOM): Avram *et al.* [123] addressed the issue of detecting network attacks in wireless ad hoc networks using self-organizing map unsupervised learning. Learning the weights are obtained through statistical analysis of the input data vectors. The main issue of this scheme is the complexity in determining input weights. Moreover, SOM-based algorithms are not suitable for detecting attacks in very large and complex data sets (i.e., large scale sensor network).

B. Quality of Service, Data Integrity and Fault Detection

Quality of service (QoS) guarantees high-priority delivery of real-time events and data. In the context of WSNs, there are potential multi-hop transmissions of data to the end user, in addition to distributing queries from a system controller to the network nodes [125]. WSNs suffer from energy and bandwidth constraints that limit the quantity of information to be transmitted from a source to destination nodes. Furthermore, data aggregation and dissemination in WSNs can be faulty and unreliable [4]. These issues coupled with random network

topologies introduce an important challenge for designing reliable algorithms for such networks. The state of the art and general QoS requirements in WSNs have been reviewed in [126].

In the following, we review the latest efforts of using machine learning techniques to achieve specific QoS and data integrity constraints. In brief, this adoption results in the following advantages:

- Different machine learning classifiers are used to recognize different types of streams, thus eliminating the need for flow-aware management techniques.
- The requirements for QoS guarantee, data integrity and fault detection depend on the network service and application. Machine learning methods are able to handle much of this while ensuring efficient resource utilization, mainly bandwidth and power utilization.

Table VII summarizes the methods that are reviewed in this subsection. The column "Characteristics" indicates features or qualities belonging to each study.

1) QoS estimation using neural network: Recently, there is growing interest in estimating and improving the performance of WSNs. For example, Snow *et al.* [35] introduced a method to estimate a sensor network dependability metric using a neural network method. Dependability is a metric that represents availability, reliability, maintainability, and survivability of a sensor network. Several attributes are used to estimate such a metric including mean time between failure (MTBF) and mean time to repair (MTTR).

Moustapha and Selmic [36] introduced a dynamic fault detection model for WSNs. This model captures the nodes' dynamic behavior and their effects on other nodes. In addition, neural network learning, which is trained using back-propagation method, was used for node identification and fault detection (a similar idea as in [35]). This study results in an effective nonlinear sensor model that suits applications with fault detection requirements.

2) MetricMap (link quality estimation framework): Link quality measurement tools may provide inaccurate and unstable readings across different environments due to different conditions such as signal variations and interference [132]. As a result, Wang *et al.* [37] presented MetricMap, a link quality estimation framework using supervised learning methods. MetricMap enhances the MintRoute [133] protocol by adopting online and offline learning methods, such as decision tree learners, to derive link quality indicators. This framework uses several local features to build the classification tree such as the received signal strength indicator (RSSI), transmission buffer size, channel load, and the forward and backward probabilities. The forward probability $p_f(l)$ is defined as the ratio of the received to the total transmitted packets over the link l, whereas the backward probability $p_b(l)$ is calculated over the reverse path. The local features are preferred over the global one as they can be found without costly communications with far away nodes. Experiments reveal that up to three times improvement in data delivery rate over basic MintRoute method can be achieved.

3) Assessing accuracy and reliability of sensor nodes using multi-output Gaussian processes: Osborne *et al.* [127] presented a real-time algorithm to determine a set of nodes that are capable of handling information processing tasks such as assessing the accuracy of collected sensor readings and predicting the missing readings. This algorithm provides a probabilistic Gaussian process based iterative implementation that is trained to re-use previous experience (i.e., the historical data) and maintain a reasonable training data size. Yet, the posterior distribution of an observed environmental variable x (e.g., sea surface temperature) is calculated using the generalized multivariate Gaussian distribution given by Eq. (5).

$$p\left(x|\mu, K, I\right) \triangleq \frac{1}{\sqrt{\det 2\pi K}} \exp\left(-\frac{1}{2}\left(x-\mu\right)^T K^{-1}\left(x-\mu\right)\right) \tag{5}$$

where μ, K are the prior mean and covariance of the variable x, respectively. Further, I denotes the historical data (a sequence of time-stamped samples) that is updated online to consider the new sequentially collected observations.

4) QoS provisioning using reinforcement learning: Oufer-hat and Mellouk [128] introduced a QoS task scheduler for adaptive multimedia sensor networks based on Q-learning technique. This scheduler significantly enhances the network throughput by reducing the transmission delay. Comparatively, Seah *et al.* [129] considered coverage as a QoS metric in WSNs that represents how efficiently the area of interest will be observed. A Q-learning method was used to develop a distributed learner that is able to find weakly monitored sites. These sites can be resolved in future re-deployment stages.

It is important to note that energy harvesting has not been considered in the above QoS mechanisms. Conversely, Hsu *et al.* [130] introduced a QoS-aware power management scheme for WSNs with energy harvesting capabilities, namely "Reinforcement Learning based QoS-aware Power Management" (RLPM). This scheme is able to adapt to the dynamic levels of nodes' energy in systems with energy harvesting capabilities. QoS-aware RLPM employs reinforcement learning to attain QoS awareness and to manage nodes' duty cycle under the energy restriction. Furthermore, Liang *et al.* [131] designed "Multi-agent Reinforcement Learning based multi-hop mesh Cooperative Communication" (MRL-CC) to be a structure modeling tool for QoS provisioning in WSNs. Basically, MRL-CC is adopted to reliably assess the data in a cooperative manner. Moreover, MRL-CC might be used to examine the impact of traffic load and node mobility on the whole network performance.

C. Miscellaneous Applications

This subsection presents miscellaneous and unique research efforts that are not discussed previously.

1) Resource management through reinforcement learning: Shah and Kumar [134] presented the "Distributed Independent Reinforcement Learning" (DIRL) algorithm that utilizes local information and application constraints to optimize various tasks over time while minimizing energy consumption of the network. Each sensor node learns the minimum required resources to perform its scheduled tasks, and maximizes its future rewards by finding optimal parameters of the intended

Table VII: Summary of quality of service, data integrity and fault detection solutions.

APPROACHES	MACHINE LEARNING ALGORITHM(S)	COMPLEXITY	CHARACTERISTICS
System's dependability [35]	NNs	High	Estimate the dependability metric
Fault detection [36]		Moderate	Dynamic fault detection model
MetricMap [37]	DT	Low	Link quality estimation
Assessing accuracy and reliability metrics [127]	GP	Moderate	Information processing tasks
A QoS scheduler [128]	RL	Low	QoS task scheduler for adaptive multimedia sensor networks
Uncertainty and coverage factors [129]		Moderate	Investigate converge problems
QoS-aware power management [130]		Low	QoS-aware power management in energy harvesting sensor nodes
QoS provisioning [131]		Low	A structure modeling tool for QoS provisioning

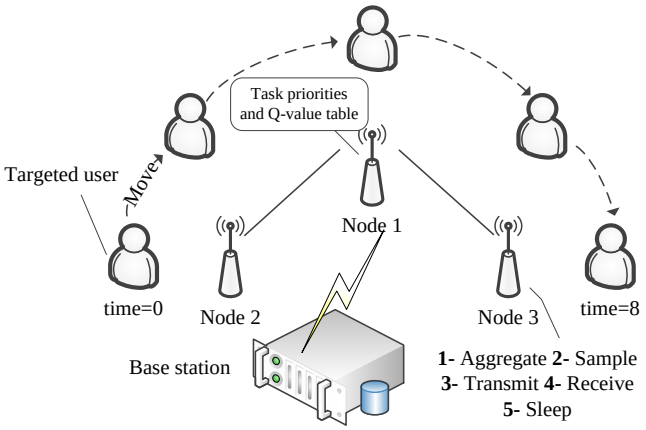


Figure 15: An example of task management using the DIRL middleware algorithm: Object tracking application [134].

application. As a typical case, consider an object recognition application, as shown in Fig. 15, which consists of five fundamental tasks: (a) aggregate two or more readings into a single reading, (b) transmit a message to the next hop, (c) receive incoming messages, (d) sample and take readings, and (e) put the radio into sleep mode. These tasks must be executed in some priority to maximize the network lifetime, where the network do not have such knowledge of priority as there is no static schedule for the events. For example, a node does not have the knowledge of when the object is going to move near to it to start taking samples. Here, the DIRL algorithm can be used to generate the required knowledge of priority by using the Q-learning algorithm and after specifying the set of reward and price value for each task.

2) Decision tree-based animals behavior classification: WSNs were applied in many applications such as environmental and habitat monitoring [135]. As an example, Nadimi *et al.* [136] employed decision tree to accurately classify the behavior of a herd of animals (active or inactive) using parameters such as the pitch angle of the neck and movement velocity. The advantages of the proposed solution for animals behavior classification are the simple implementation and low complexity due to the use of a few critical features.

3) Clock synchronization using self-organizing map: Clock synchronization between sensor nodes is an important process, since most operations of the nodes must be consistent with each other. Moreover, the design of such methods for WSNs has to consider the limited resource constraints. For example, Paladina *et al.* [137] proposed to use SOM to ensure reliable clock synchronization for large scale networks. Nodes predict near-optimal estimation of the current time without having a central timing device and with limited storage and computing resources. However, this method assumes a uniform deployment of the nodes over the monitored area, as well as the same transmission powers for all nodes.

4) Air quality monitoring using neural networks: Postolache *et al.* [138] proposed a neural networks-based method for measuring air pollution levels using inexpensive gas sensor nodes, while eliminating the effects of temperature and humidity on sensor readings. This solution detects the air quality and gas concentration using neural networks implemented using JavaScript (JS). As a result, the solution is able to distribute processing between web server and end user computers (i.e., a combination of client and server side scripts).

5) Intelligent lighting control using neural networks: Gao *et al.* [139] introduced a new standard for lighting control in smart building using the neural network algorithm. A radial basis function (RBF) neural network is used to extract a new mathematical expression, called "Illuminance Matrix" (I-matrix), to measure the degree of illuminance in the lighted area. Fundamentally, in the field of lighting control, converting the collected data from the photosensors to a form that is suitable for digital signal processing is a crucial issue and can highly affect the performance of the developed system. The article shows that using the I-matrix scheme can achieve about 60% more accuracy compared to the standard methods.

V. FUTURE APPLICATIONS OF MACHINE LEARNING IN WIRELESS SENSOR NETWORKS

Although machine learning techniques have been applied to many applications in WSNs, many issues are still open and need further research efforts.

A. Compressive Sensing and Sparse Coding

In practice, a large number of sensor measurements are usually required to maintain a desired detection accuracy. This introduces several challenges to network designers such as network management and communication issues. Given that 80 percent of the nodes' energy is consumed while sending and receiving data [140], data compression and dimensionality

reduction techniques can be used to reduce transmission and hence prolong the network lifetime.

Traditional data compression techniques may result in extra energy consumption due to their high computational and memory requirements. In [141], Barr and Asanović studied the tradeoff between energy consumption in data transmission and compression. This study approximates the efficiency threshold of data compression in WSNs to be 1 bit data reduction using 485-1267 ADD instructions.

Even though compressive sensing can be recast as a linear program, it still not applicable for on-node compression. As a result, it is important to apply and extend the basic concept of compressive sensing to meet the resource constraint of WSNs. For more on the theoretical performance of decentralized compressive sensing, please refer to [142], [143], [144]. Examples of similar emerging techniques include independent component analysis, dictionary learning, non-negative matrix factorization and singular value decomposition.

B. Distributed and Adaptive Machine Learning Techniques for WSNs

Distributed machine learning techniques suit limited resource devices such as WSNs. Compared to centralized learning algorithms, distributed learning methods require less computational power and smaller memory footprint (i.e., they do not need to consider the whole network information). The decentralized learning techniques enable the nodes to rapidly adapt their future behavior and predictions in tune with the current environment conditions. For such reasons, distributed and adaptive learning algorithms are adequate for in-network processing of data while avoiding exhausting the nodes with high computational tasks [145]. Examples of recent online learning algorithms include "Adaptive Regularization of Weights" (AROW) [146], "Improved Ellipsoid Method for Online Learning" (IELLIP) [147] and "Soft Confidence-Weighted" (SCW) [148]. Kotecha *et al.* [149] studied some distributed classification algorithms for WSNs.

C. Resource Management Using Machine Learning

Energy saving is a crucial issue in developing efficient WSNs algorithms and techniques. This design goal can be achieved using two main techniques, namely, by enhancing communication related protocols (e.g., routing and MAC protocols design) and by detecting nonfunctional and energy wasteful activities. The first technique includes physical, MAC and networking layer protocols. As it is discussed in this survey, this technique has been widely studied and enhanced using machine learning algorithms. The second technique focuses on decreasing the consumed energy in minor and nonfunctional requirements. For example, sensor nodes will consume their energy when over-listening to other nodes' transmissions [150]. Accordingly, such operations unnecessarily increase the active time of the nodes (i.e., increase nodes' duty cycle). The nodes that are equipped with machine learning techniques will be able to optimize their resource management and power allocation operations under those circumstances.

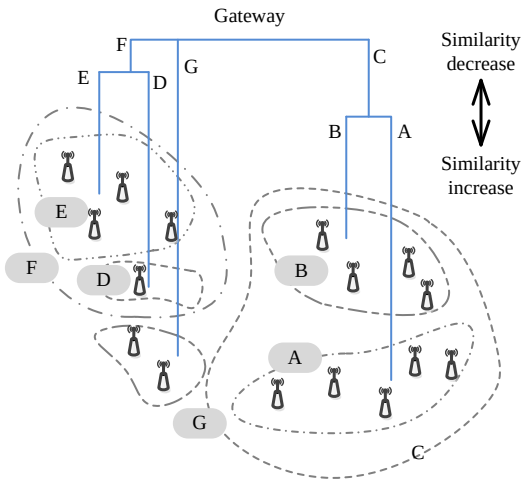


Figure 16: Hierarchical clustering of network's nodes based on data spatial and temporal correlations in a temperature monitoring system.

D. Detecting Data Spatial and Temporal Correlations Using Hierarchical Clustering

Hierarchical clustering is an unsupervised learning algorithm that aims to build a hierarchy of clusters. Basically, hierarchical clustering algorithms generate decomposition of the set of objects, which could be a set of sensor nodes in WSNs. Broadly speaking, hierarchical clustering can provide an emerging clustering technique in WSNs using some clustering criteria such as spatial and temporal correlations of readings. Figure 16 illustrates such hierarchically clustered network based on spatial and temporal correlations of readings in a temperature monitoring system. In this example, Cluster C is formed by combining Clusters A and B, and so on for the rest of the clusters in the network.

The study of data correlation based on hierarchical clustering method will provide simple methods for energy saving. In such formations, only one node from each cluster is activated at a time to cover and monitor the whole cluster area. Typical methods of hierarchical clustering includes "Balanced Iterative Reducing and Clustering using Hierarchies" (BIRCH) [151] and "Clustering Using Representatives" (CURE) [152].

VI. CONCLUSIONS

Wireless sensor networks are different from traditional network in various aspects, thereby necessitating protocols and tools that address unique challenges and limitations. As a consequence, wireless sensor networks require innovative solutions for energy aware and real-time routing, security, scheduling, localization, node clustering, data aggregation, fault detection and data integrity. Machine learning provides a collection of techniques to enhance the ability of wireless sensor network to adapt to the dynamic behavior of its surrounding environment. Table VIII summarizes studies that have adopted machine learning methods to address these challenges from distinct research areas.

From the discussion so far, it became clear that many design challenges in wireless sensor networks have been resolved

Table VIII: Summary of publications resolving various WSN challenges by the adoption of machine learning techniques.

	Functional					Non-functional	
	Routing	Clustering and data aggregation	Localization and objects targeting	Event detection and query processing	MAC	Security and intrusion detection	QoS, data integrity and fault detection
Bayesian statistics			[21], [22]	[91], [94], [92]	[28]	[31]	
k-nearest neighbors				[24], [25]		[32]	
Neural network		[79]	[23], [99], [100]	[26]	[29], [30]		[35], [36]
Support vector machines			[49], [50], [51]			[33], [34], [46], [47], [48]	
Decision tree		[80]	[101], [102]	[27]	[115]		[37]
Gaussian Processes		[81], [82]	[103], [104]				[127]
Self-organizing map	[70]	[54]	[105], [107], [106]			[123]	
Vector quantization		[83]					
Principal component		[55], [56], [57], [58], [84]		[93]			
k-means algorithm		[59], [60]					
Reinforcement learning	[65], [66], [67], [68]	[85], [86]	[108]	[134]	[114], [112]		[128], [129], [130], [131]

using several machine learning methods. In this paper, an extensive literature review over the period 2002-2013 on such studies was presented. In summary, adopting machine learning algorithms in wireless sensor networks has to consider the limited resources of the network, as well as the diversity of learning themes and patterns that will suit the problem at hand. Moreover, numerous issues are still open and need further research efforts such as developing lightweight and distributed message passing techniques, online learning algorithms, hierarchical clustering patterns and adopting machine learning in resource management problem of wireless sensor networks.

References

[1] T. O. Ayodele, "Introduction to machine learning," in *New Advances in Machine Learning*. InTech, 2010.

[2] A. H. Duffy, "The "what" and "how" of learning in design," *IEEE Expert*, vol. 12, no. 3, pp. 71–76, 1997.

[3] P. Langley and H. A. Simon, "Applications of machine learning and rule induction," *Communications of the ACM*, vol. 38, no. 11, pp. 54–64, 1995.

[4] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

[5] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *22nd International Conference on Distributed Computing Systems Workshops*, 2002, pp. 575–578.

[6] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

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

[8] J. Wan, M. Chen, F. Xia, L. Di, and K. Zhou, "From machine-to-machine communications towards cyber-physical systems," *Computer Science and Information Systems*, vol. 10, pp. 1105–1128, 2013.

[9] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[10] A. G. Hoffmann, "General limitations on machine learning," pp. 345–347, 1990.

[11] M. Di and E. M. Joo, "A survey of machine learning in wireless sensor netoworks from networking and application perspectives," in *6th International Conference on Information, Communications Signal Processing*, 2007, pp. 1–5.

[12] A. Forster, "Machine learning techniques applied to wireless ad-hoc networks: Guide and survey," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, pp. 365–370.

[13] A. Förster and M. Amy L, *Machine learning across the WSN layers*. InTech, 2011.

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

[15] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[16] R. Kulkarni, A. Förster, and G. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.

[17] S. Das, A. Abraham, and B. K. Panigrahi, *Computational intelligence: Foundations, perspectives, and recent trends*. John Wiley & Sons, Inc., 2010, pp. 1–37.

[18] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook, 2012.

[19] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-supervised learning*. MIT press Cambridge, 2006, vol. 2.

[20] S. Kulkarni, G. Lugosi, and S. Venkatesh, "Learning pattern classification-a survey," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2178–2206, 1998.

[21] M. Morelande, B. Moran, and M. Brazil, "Bayesian node localisation in wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 2545–2548.

[22] C.-H. Lu and L.-C. Fu, "Robust location-aware activity recognition using wireless sensor network in an attentive home," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 598–609, 2009.

[23] A. Shareef, Y. Zhu, and M. Musavi, "Localization using neural networks in wireless sensor networks," in *Proceedings of the 1st International Conference on Mobile Wireless Middleware, Operating Systems, and Applications*, 2008, pp. 1–7.

[24] J. Winter, Y. Xu, and W.-C. Lee, "Energy efficient processing of k nearest neighbor queries in location-aware sensor networks," in *2nd International Conference on Mobile and Ubiquitous Systems: Networking and Services*. IEEE, 2005, pp. 281–292.

[25] P. P. Jayaraman, A. Zaslavsky, and J. Delsing, "Intelligent processing of k-nearest neighbors queries using mobile data collectors in a location aware 3D wireless sensor network," in *Trends in Applied Intelligent Systems*. Springer, 2010, pp. 260–270.

[26] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in *International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 2, 2005, pp. 1214–1217.

[27] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, and P. J. Havinga, "Distributed event detection in wireless sensor networks for disaster management," in *2nd International Conference on Intelligent Networking and Collaborative Systems*. IEEE, 2010, pp. 507–512.

[28] M. Kim and M.-G. Park, "Bayesian statistical modeling of system energy saving effectiveness for MAC protocols of wireless sensor networks," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2009, vol. 209, pp. 233–245.

[29] Y.-J. Shen and M.-S. Wang, "Broadcast scheduling in wireless sensor networks using fuzzy hopfield neural network," *Expert Systems with Applications*, vol. 34, no. 2, pp. 900 – 907, 2008.

[30] R. V. Kulkarni and G. K. Venayagamoorthy, "Neural network based secure media access control protocol for wireless sensor networks," in *Proceedings of the 2009 International Joint Conference on Neural Networks*, ser. IJCNN'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3437–3444.

[31] D. Janakiram, V. Adi Mallikarjuna Reddy, and A. Phani Kumar, "Outlier detection in wireless sensor networks using Bayesian belief networks," in *1st International Conference on Communication System Software and Middleware*. IEEE, 2006, pp. 1–6.

[32] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Knowledge and information systems*, vol. 34, no. 1, pp. 23–54, 2013.

[33] S. Kaplantzis, A. Shilton, N. Mani, and Y. Sekercioglu, "Detecting selective forwarding attacks in wireless sensor networks using support vector machines," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, pp. 335–340.

[34] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *International Conference on Communications*, 2007, pp. 3864–3869.

[35] A. Snow, P. Rastogi, and G. Weckman, "Assessing dependability of wireless networks using neural networks," in *Military Communications Conference*. IEEE, 2005, pp. 2809–2815 Vol. 5.

[36] A. Moustapha and R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[37] Y. Wang, M. Martonosi, and L.-S. Peh, "Predicting link quality using supervised learning in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 3, pp. 71–83, 2007.

[38] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *Database Theory*. Springer, 1999, pp. 217–235.

[39] T. O. Ayodele, "Types of machine learning algorithms," in *New Advances in Machine Learning*. InTech, 2010.

[40] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[41] R. Lippmann, "An introduction to computing with neural nets," *ASSP Magazine, IEEE*, vol. 4, no. 2, pp. 4–22, 1987.

[42] W. Dargie and C. Poellabauer, *Localization*. John Wiley & Sons, Ltd, 2010, pp. 249–266.

[43] T. Kohonen, *Self-organizing maps*, ser. Springer Series in Information Sciences. Springer Berlin Heidelberg, 2001, vol. 30.

[44] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[45] I. Steinwart and A. Christmann, *Support vector machines*. Springer, 2008.

[46] Z. Yang, N. Meratnia, and P. Havinga, "An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine," in *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, 2008, pp. 151–156.

[47] Y. Chen, Y. Qin, Y. Xiang, J. Zhong, and X. Jiao, "Intrusion detection system based on immune algorithm and support vector machine in wireless sensor network," in *Information and Automation*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2011, vol. 86, pp. 372–376.

[48] Y. Zhang, N. Meratnia, and P. J. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1062–1074, 2013.

[49] W. Kim, J. Park, and H. Kim, "Target localization using ensemble support vector regression in wireless sensor networks," in *Wireless Communications and Networking Conference*, 2010, pp. 1–5.

[50] D. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 981–994, 2008.

[51] B. Yang, J. Yang, J. Xu, and D. Yang, "Area localization algorithm for mobile nodes in wireless sensor networks based on support vector machines," in *Mobile Ad-Hoc and Sensor Networks*. Springer, 2007, pp. 561–571.

[52] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011, vol. 40.

[53] C. E. Rasmussen, "Gaussian processes for machine learning," in *in: Adaptive Computation and Machine Learning*. Citeseer, 2006.

[54] S. Lee and T. Chung, "Data aggregation for wireless sensor networks using self-organizing map," in *Artificial Intelligence and Simulation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, vol. 3397, pp. 508–517.

[55] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data acquisition through joint compressive sensing and principal component analysis," in *Global Telecommunications Conference*. IEEE, 2009, pp. 1–6.

[56] R. Masiero, G. Quer, M. Rossi, and M. Zorzi, "A Bayesian analysis of compressive sensing data recovery in wireless sensor networks," in *International Conference on Ultra Modern Telecommunications Workshops*, 2009, pp. 1–6.

[57] A. Rooshenas, H. Rabiee, A. Movaghar, and M. Naderi, "Reducing the data transmission in wireless sensor networks using the principal component analysis," in *6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, 2010, pp. 133–138.

[58] S. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *11th International Workshop on Signal Processing Advances in Wireless Communications*, 2010, pp. 1–5.

[59] Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, and Y.-Y. Hsieh, "iMouse: An integrated mobile surveillance and wireless sensor system," *Computer*, vol. 40, no. 6, pp. 60–66, 2007.

[60] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[61] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[62] I. T. Jolliffe, *Principal component analysis*. Springer verlag, 2002.

[63] D. Feldman, M. Schmidt, C. Sohler, D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering," in *SODA*, 2013, pp. 1434–1453.

[64] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[65] R. Sun, S. Tatsumi, and G. Zhao, "Q-MAP: A novel multicast routing method in wireless ad hoc networks with multiagent reinforcement learning," in *Region 10 Conference on Computers, Communications, Control and Power Engineering*, vol. 1, 2002, pp. 667–670 vol.1.

[66] S. Dong, P. Agrawal, and K. Sivalingam, "Reinforcement learning based geographic routing protocol for UWB wireless sensor network," in *Global Telecommunications Conference*. IEEE, 2007, pp. 652–656.

[67] A. Förster and A. Murphy, "FROMS: Feedback routing for optimizing multiple sinks in wsn with reinforcement learning," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, pp. 371–376.

[68] R. Arroyo-Valles, R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Q-probabilistic routing in wireless sensor networks," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, pp. 1–6.

[69] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: An efficient framework for modeling sensor network data," in *3rd International Symposium on Information Processing in Sensor Networks*, 2004, pp. 1–10.

[70] J. Barbancho, C. León, F. Molina, and A. Barbancho, "A new QoS routing algorithm based on self-organizing maps for wireless sensor networks," *Telecommunication Systems*, vol. 36, pp. 73–83, 2007.

[71] B. Scholkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA, USA: MIT Press, 2001.

[72] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[73] G. Aiello and G. Rogerson, "Ultra-wideband wireless systems," *IEEE Microwave Magazine*, vol. 4, no. 2, pp. 36–47, 2003.

[74] R. Rajagopalan and P. Varshney, "Data-aggregation techniques in sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[75] G. Crosby, N. Pissinou, and J. Gadze, "A framework for trust-based cluster head election in wireless sensor networks," in *2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, 2006, pp. 10–22.

[76] J.-M. Kim, S.-H. Park, Y.-J. Han, and T.-M. Chung, "CHEF: Cluster head election mechanism using fuzzy logic in wireless sensor networks," in *10th International Conference on Advanced Communication Technology*, vol. 1. IEEE, 2008, pp. 654–659.

[77] S. Soro and W. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *19th IEEE International Parallel and Distributed Processing Symposium*, 2005, pp. 4–8.

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

[79] H. He, Z. Zhu, and E. Makinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 973–982, 2009.

[80] G. Ahmed, N. M. Khan, Z. Khalid, and R. Ramer, "Cluster head selection using decision trees for wireless sensor networks," in *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, 2008, pp. 173–178.

[81] E. Ertin, "Gaussian process models for censored sensor readings," in *14th Workshop on Statistical Signal Processing*. IEEE, 2007, pp. 665–669.

[82] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, pp. 19:1–19:35, 2009.

[83] S. Lin, V. Kalogeraki, D. Gunopulos, and S. Lonardi, "Online information compression in sensor networks," in *IEEE International Conference on Communications*, vol. 7. IEEE, 2006, pp. 3371–3376.

[84] C. Fenxiong, L. Mingming, W. Dianhong, and T. Bo, "Data compression through principal component analysis over wireless sensor networks," *Journal of Computational Information Systems*, vol. 9, no. 5, pp. 1809–1816, 2013.

[85] A. Förster and A. Murphy, "CLIQUE: Role-free clustering with q-learning for wireless sensor networks," in *29th IEEE International Conference on Distributed Computing Systems*, 2009, pp. 441–449.

[86] M. Mihaylov, K. Tuyls, and A. Nowe, "Decentralized learning in wireless sensor networks," in *Adaptive and Learning Agents*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 5924, pp. 60–73.

[87] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[88] M. Duarte and Y. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, 2011.

[89] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[90] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[91] B. Krishnamachari and S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[92] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Wireless Sensor Networks*. Springer, 2008, pp. 17–33.

[93] H. Malik, A. Malik, and C. Roy, "A methodology to optimize query in wireless sensor networks using historical data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, pp. 227–238, 2011.

[94] Q. Chen, K.-Y. Lam, and P. Fan, "Comments on "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks"," *IEEE Transactions on Computers*, vol. 54, no. 9, pp. 1182–1183, 2005.

[95] K. Sha, W. Shi, and O. Watkins, "Using wireless sensor networks for fire rescue applications: Requirements and challenges," in *IEEE International Conference on Electro/information Technology*, 2006, pp. 239–244.

[96] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[97] J. Wang, R. Ghosh, and S. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.

[98] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*. ACM, 2002, pp. 105–111.

[99] S. Yun, J. Lee, W. Chung, E. Kim, and S. Kim, "A soft computing approach to localization in wireless sensor networks," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7552–7561, 2009.

[100] S. Chagas, J. Martins, and L. de Oliveira, "An approach to localization scheme of wireless sensor networks based on artificial neural networks and genetic algorithms," in *10th International Conference on New Circuits and Systems*. IEEE, 2012, pp. 137–140.

[101] Z. Merhi, M. Elgamel, and M. Bayoumi, "A lightweight collaborative fault tolerant target localization system for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, pp. 1690–1704, 2009.

[102] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater survelliance systems," *Ad Hoc Networks*, vol. 4, no. 4, pp. 431–446, 2006.

[103] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *The Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[104] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1279–1290, 2012.

[105] L. Paladina, M. Paone, G. Iellamo, and A. Puliafito, "Self organizing maps for distributed localization in wireless sensor networks," in *12th IEEE Symposium on Computers and Communications*, 2007, pp. 1113–1118.

[106] G. Giorgetti, S. K. S. Gupta, and G. Manes, "Wireless localization using self-organizing maps," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 293–302.

[107] J. Hu and G. Lee, "Distributed localization of wireless sensor networks using self-organizing maps," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008, pp. 284–289.

[108] S. Li, X. Kong, and D. Lowe, "Dynamic path determination of mobile beacons employing reinforcement learning for wireless sensor localization," in *26th International Conference on Advanced Information Networking and Applications Workshops*, 2012, pp. 760–765.

[109] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 247–271.

[110] Y.-X. Wang and Y.-J. Zhang, "Non-negative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[111] H.-P. Tan, R. Diamant, W. K. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Engineering*, vol. 38, no. 14, pp. 1663–1676, 2011.

[112] Y. Chu, P. Mitchell, and D. Grace, "ALOHA and q-learning based medium access control for wireless sensor networks," in *International Symposium on Wireless Communication Systems*, 2012, pp. 511–515.

[113] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[114] Z. Liu and I. Elhanany, "RL-MAC: A reinforcement learning based MAC protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 3, pp. 117–124, 2006.

[115] M. Sha, R. Dor, G. Hackmann, C. Lu, T.-S. Kim, and T. Park, "Self-adapting MAC layer for wireless sensor networks," Technical Report WUCSE-2013-75, Washington University in St. Louis, Tech. Rep., 2013.

[116] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *21st Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2002, pp. 1567–1576 vol.3.

[117] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 171–180.

[118] K. Klues, G. Hackmann, O. Chipara, and C. Lu, "A component-based architecture for power-efficient media access control in wireless sensor networks," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. ACM, 2007, pp. 59–72.

[119] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald, "MultiMAC-an adaptive MAC framework for dynamic radio networking," in *International Symposium on New Frontiers in Dynamic Spectrum Access Networks*. IEEE, 2005, pp. 548–555.

[120] D. Moss and P. Levis, "BoX-MACs: Exploiting physical and link layer boundaries in low-power networking," *Computer Systems Laboratory Stanford University*, 2008.

[121] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 1–14.

[122] Z. Alliance, "Zigbee-2007 specification," *Online: http://www.zigbee.org/Specifications/ZigBee/Overview.aspx*, 2007.

[123] T. Avram, S. Oh, and S. Hariri, "Analyzing attacks in wireless ad hoc network with self-organizing maps," in *5th Annual Conference on Communication Networks and Services Research*, 2007, pp. 166–175.

[124] L. N. De Castro and J. Timmis, *Artificial immune systems: A new computational intelligence approach*. Springer, 2002.

[125] G. J. Pottie and A. Pandya, *Quality of service in wireless sensor networks*. John Wiley & Sons, Inc., 2008, pp. 401–435.

[126] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: A survey," in *International Conference on Wireless Networks*, vol. 233, 2004.

[127] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 2008, pp. 109–120.

[128] N. Ouferhat and A. Mellouk, "A QoS scheduler packets for wireless sensor networks," in *International Conference on Computer Systems and Applications*, 2007, pp. 211–216.

[129] M. Seah, C.-K. Tham, V. Srinivasan, and A. Xin, "Achieving coverage through distributed reinforcement learning in wireless sensor networks," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, pp. 425–430.

[130] R. Hsu, C.-T. Liu, K.-C. Wang, and W.-M. Lee, "QoS-aware power management for energy harvesting wireless sensor network utilizing reinforcement learning," in *International Conference on Computational Science and Engineering*, vol. 2. IEEE, 2009, pp. 537–542.

[131] X. Liang, M. Chen, Y. Xiao, I. Balasingham, and V. C. M. Leung, "A novel cooperative communication protocol for QoS provisioning in wireless sensor networks," in *5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops*, 2009, pp. 1–6.

[132] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 4, p. 34, 2012.

[133] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 14–27.

[134] K. Shah and M. Kumar, "Distributed independent reinforcement learning (DIRL) approach to resource management in wireless sensor networks," in *Internatonal Conference on Mobile Adhoc and Sensor Systems*, 2007, pp. 1–9.

[135] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*. ACM, 2002, pp. 88–97.

[136] E. Nadimi, H. T. Søgaard, and T. Bak, "Zigbee-based wireless sensor networks for classifying the behaviour of a herd of animals using classification trees," *Biosystems Engineering*, vol. 100, no. 2, pp. 167–176, 2008.

[137] L. Paladina, A. Biundo, M. Scarpa, and A. Puliafito, "Self organizing maps for synchronization in wireless sensor networks," in *New Technologies, Mobility and Security*, 2008, pp. 1–6.

[138] O. Postolache, J. Pereira, and P. Girao, "Smart sensors network for air quality monitoring applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3253–3262, 2009.

[139] Y. Gao, Y. Lin, and Y. Sun, "A wireless sensor network based on the novel concept of an I-matrix to achieve high-precision lighting control," *Building and Environment*, vol. 70, pp. 223–231, 2013.

[140] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *International Conference on Information Technology: Coding and Computing*, vol. 2, 2005, pp. 8–13 Vol. 2.

[141] K. C. Barr and K. Asanović, "Energy-aware lossless data compression," *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250–291, Aug. 2006.

[142] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, 2008.

[143] J. Luo, L. Xiang, and C. Rosenberg, "Does compressed sensing improve the throughput of wireless sensor networks?" in *International Conference on Communications*, 2010, pp. 1–6.

[144] S. Feizi, M. Medard, and M. Effros, "Compressive sensing over networks," in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010, pp. 1129–1136.

[145] J. B. Predd, S. Kulkarni, and H. V. Poor, *Distributed learning in wireless sensor networks*. IEEE, 2006, vol. 23, no. 4, pp. 56–69.

[146] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," *Machine Learning*, vol. 91, no. 2, pp. 155–187, 2013.

[147] L. Yang, R. Jin, and J. Ye, "Online learning by ellipsoid method," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 1153–1160.

[148] J. Wang, P. Zhao, and S. C. Hoi, "Exact soft confidence-weighted learning," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 121–128.

[149] J. H. Kotecha, V. Ramachandran, and A. Sayeed, "Distributed multitarget classification in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 703–713, 2005.

[150] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. E. Culler, P. Levis, S. Shenker *et al.*, "An architecture for energy management in wireless sensor networks," *SIGBED Review*, vol. 4, no. 3, pp. 31–36, 2007.

[151] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Record*, vol. 25, no. 2, pp. 103–114, Jun. 1996.

[152] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '98. New York, NY, USA: ACM, 1998, pp. 73–84.