This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

Machine Learning for Advanced Wireless Sensor Networks: A Review

Taeyoung Kim, Luiz Felipe Vecchietti, Kyujin Choi, Sangkeum Lee, and Dongsoo Har, *Senior Member, IEEE*

***Abstract*—Wireless sensor networks (WSNs) are typically used with dynamic conditions of task-related environments for sensing(monitoring) and gathering of raw sensor data for subsequent forwarding to a base station. In order to deploy WSNs in real environments, a variety of technical challenges must be addressed. With traditional techniques developed for a specific task, it is hard to react in dynamic situations beyond the scope of the intended task. As a solution to this problem, machine learning (ML) techniques that are able to handle dynamic situations with successful learning process have been applied lately in WSNs. Particularly, deep learning (DL) techniques, a class of ML techniques characterized by the use of deep neural network, are used for WSNs to extract higher level features from raw sensor data. A range of benefits obtained from ML techniques applied to WSNs can be described as reduced computational complexity, increased feasibility in finding optimal solutions, increased energy efficiency, etc. On the other hand, it is found from our survey that large training time and large dataset to get acceptable performance are accompanied with large energy consumption which is not favorable for resource-restrained WSNs. Reviews on the applications of ML techniques in WSNs appeared in the literature. However, few reviews have dealt with the applications of DL techniques in WSNs. In this review, recent developments of ML techniques for WSNs are presented with much emphasis on DL techniques. The DL techniques developed for various applications in WSNs are addressed together with their respective deep neural network architectures.**

***Index Terms*—Wireless Sensor Networks, Machine Learning, Deep Learning**

I. Introduction

WIRELESS sensor network (WSN) composed of sensor nodes is deployed in the target area where a phenomenon of interest occurs [1]. The typical goal of a WSN is to monitor and collect information from the target area and transfer the detected information to a base station or a sink node [2]. While fulfilling a task to achieve the goal is critical for WSNs in general, cost-efficiency and energy-efficiency are increasingly important for larger scale WSNs [3]. A large number of sensors deployed in a large coverage area cause many technical issues that should be addressed while developing algorithms to achieve the goal. These technical issues include localization [4]–[8], coverage [9]–[13], anomaly detection [14]–[18], fault detection [19]–[23], routing [24]–[28], data aggregation [29]–[33], synchronization [34]–[38], congestion control [39]–[43], event detection [44]–[48], energy harvesting [49]–[54], and security [55]–[59]. Most solutions to resolve these technical issues rely on a statistical or mathematical framework establishing relationships between raw sensor data obtained in the first stage and final decision made in the last stage. The use of a well-defined statistical or mathematical framework often leads to the successful operation of WSNs when the dynamic variation of operating conditions of WSNs is within tolerable level. However, when the dynamic variation is beyond the tolerable level, these frameworks fail in providing appropriate solutions. Furthermore, most of these frameworks used for modeling technical issues in WSNs only provide the instantaneous relationship between raw sensor data and the final decision, rather than a trend-based relationship between them. From these perspectives, machine learning (ML) techniques that could address the systematic correlation between the two have been applied to applications of WSNs lately.

The ML technique which automatically learns tasks using example data without being specifically programmed is a class of artificial intelligence (AI) algorithm [60], [61]. ML technique facilitates the development of complex models solely based on data, without the need of specialized human intervention. Thanks to this intervention-free property, the solutions obtained from ML techniques are more efficient, cheaper, and more flexible. ML techniques are usually categorized into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (RL). Over the past decades, ML techniques have been applied to diverse applications including social media, medical systems, transportation, computer vision, and wireless communication. The ML techniques have been applied successfully to many WSN applications such as classification, clustering, dimensionality reduction, feature extraction, and forecasting [62], [63]. The advantages of applying ML techniques to WSNs can be enumerated as finding an optimal solution, e.g., optimal location for the placement of sensors, reducing computational complexity, e.g., reducing the required bandwidth to transmit collected sensor data, and flexibility, e.g., capability to react to dynamic inputs. Some ML techniques providing successful results for diverse applications of WSNs can be described by explicit mathematical expressions. However, other ML techniques like deep learning (DL) techniques cannot be understood by explicit mathematical expressions.

The DL technique that has attracted increased attention lately due to achieving breakthrough results in important areas such as natural language processing, image classification [64],

Manuscript received Month XX, 20XX.

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korea Government (MSIT) (Development of Artificial Intelligence Technology that Continuously Improves Itself as the Situation Changes in the Real World) under Grant 2020-0-00440.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

and bio-informatics [65] is a subset of ML technique [66]. The DL technique is able to learn hierarchical representations, using artificial neural networks (ANNs) with multiple hidden layers of nonlinear processing units. The DL architectures have also gained attention because of their flexibility that allows them to be employed in diverse applications. Recently, DL techniques have been applied to applications of WSNs and are expected to be a crucial part of WSNs in the near future.

Detailed surveys on the employment of ML algorithms for WSNs have been covered by Alsheikh et al. [62], for the period of 2002-2013, and by Kumar et al. [63], for the period of 2014-March 2018, in which various applications of WSNs are discussed. In this review, an updated survey with recent research works for ML-based and DL-based algorithms used for WSNs is presented. This review complements the review in [63] by covering the period of 2018-2020, and adding DL-based works for WSNs.

The rest of this paper is organized as follows. Section II presents background of the ML techniques. In Section III, a review of design issues on WSNs with ML techniques is presented. Section IV describes architectures and applications of DL techniques in WSNs. Section V presents impact and limitations of using ML/DL techniques instead of conventional approaches. Section VI concludes this review paper.

II. Machine Learning background

In this section, an introduction to ML is presented. Various ML techniques that are applied to WSNs are introduced. Depending on how the learning is performed, ML techniques can be categorized into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Despite being a subset of ML technique, DL technique based on deep neural networks is presented and described in details in Section IV.

A. Supervised Learning

Figure 1 presents a classification of ML techniques. The majority of the ML techniques fall into the category of supervised learning. In supervised learning, there exists a dataset that is a collection of labeled examples $((x_1, y_1), ..., (x_n, y_n))$ where y_i is the label of input x_i. The goal of a supervised learning algorithm is to learn a mapping function (model) that can predict the label information for the given input. This class of algorithms is called supervised learning because the correct label for each feature vector is known a priori. Depending on the type of the output label, supervised learning algorithms can be further grouped into regression and classification. If the output label is a real number, the problem can be solved by a regression algorithm. If the output label is a finite set of classes, the problem can be solved by a classification algorithm. Supervised learning algorithms are presented in [67], [68] to address localization and in [69], [70] to solve target tracking and in [71]–[73] to deal with anomaly detection. As other applications, they are used for data aggregation [74]–[76], congestion control [77], [78], energy harvesting [79] and security [80], [81].

1) k-Nearest Neighbors: The k-nearest neighbors (k-NN) algorithm [82] is a type of instance-based learning used for regression and classification. The k-NN algorithm calculates distances between the input feature vector and feature vectors of the examples in the dataset. The examples associated with the nearest feature vectors are defined as the k nearest neighbors. Various distance metrics can be used to obtain the nearest feature vectors, such as Euclidean distance, Hamming distance, Minkowski distance, among others, and can affect the final performance of the algorithm. When the k-NN algorithm is used for classification, the object is assigned to the most common class among its k nearest neighbors. The size of the input dataset affects the performance of the k-NN algorithm. The k-NN is applied to solve localization [68], data aggregation [74] and security [80] in WSNs.

2) Decision Trees: A decision tree (DT) [83], as the name suggests, is a classifier with the structure of a tree that can be learned automatically from data and can be used to make decisions, e.g., classification of an object according to the given feature vector. The DT algorithm creates leaf nodes (final outcomes) and decision nodes (choices) that implement if-then rules evaluating the feature vector [84]. The quality of a leaf node created is evaluated based on a criterion called entropy. The DT algorithm stops, representing that the final tree is created, when it is not possible to create a leaf node that minimizes the entropy or if the tree reaches its maximum depth. As the leaf nodes implement an if-then rule that can be interpreted, DT algorithms offer a good solution when a comprehensive analysis of the final classification is needed. The DT has been applied to solve data aggregation [74] and congestion control [77] in WSNs.

3) Random Forest: The random forest (RF) algorithm [85] is developed as an evolution of the DT algorithm. In the RF algorithm, multiple decision trees are trained in parallel with bootstrapping to create an ensemble method that improves the generalization of the model. The DT trained by the RF algorithm is unique and the decisions of the DT are aggregated to get the final decision. Because the RF uses an ensemble of decision trees, the overfitting problem is often alleviated, as compared to other classifiers using only one DT. As a result, the RF algorithm combines a large number of weak individual DT classifiers to create a strong classifier. The RF is adopted to solve anomaly detection [86] and security [80] for WSNs.

4) Naive Bayes: The Naive Bayes model [87] is a classification algorithm that relies on Bayes' theorem. The Naive Bayes models are simple and fast algorithms especially suitable for multi-dimensional feature vectors. The Naive Bayes classifiers try to calculate the probability of a label for a given feature vector. These probabilities are calculated using Bayes' theorem, in which the probabilities needed to make the inference are calculated with the examples contained in the dataset. In Naive Bayes classifiers, it is assumed that the features are independent when calculating the probabilities of features. The Naive Bayes model is used in [70] and [80] for target tracking and security.

5) Support Vector Machine: The support vector machine (SVM) [88] is an algorithm used for problems of classification into two groups, e.g., classification of an email into 'spam

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

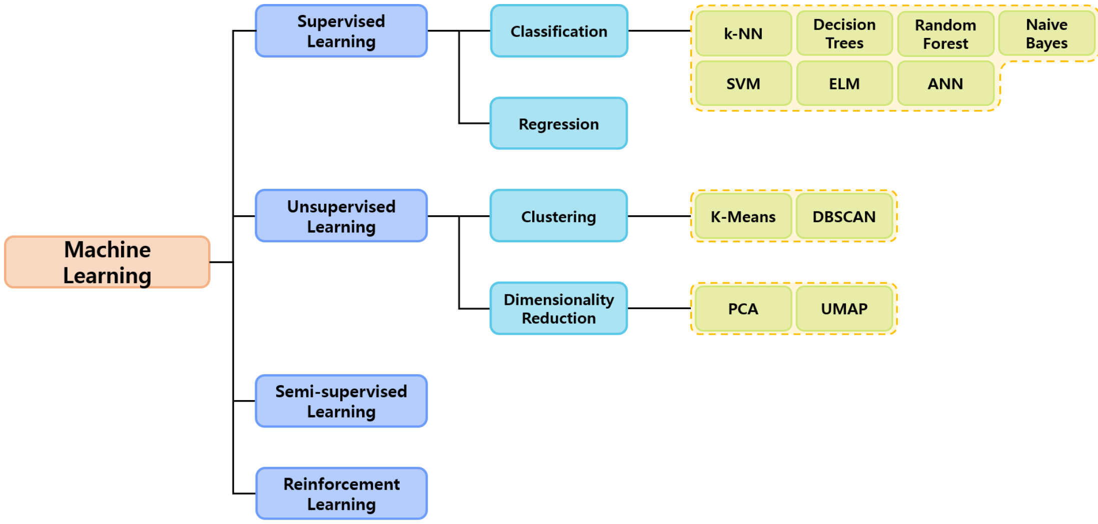

Fig. 1. Classification of ML techniques.

mails' or not. If the label using the SVM algorithm is +1 (one) for one class, then the label is -1 (minus one) for the other class. The SVM algorithm sees the feature vector as a point in a multi-dimensional space and tries to create a decision boundary (hyperplane) in the space that successfully separates both classes. The best decision boundary is the one that has the largest distance between the closest examples of the two classes, a.k.a. the largest margin. The SVM algorithms are powerful when a linear hyperplane (model) can separate both classes in the feature vector space. When the classes are not linearly separable in the feature vector space, non-linear kernels can be used to make the decision boundary in different non-linear patterns. The SVM algorithm can be applied to solve various issues in WSNs such as target tracking [69], data aggregation [74], congestion control [78], and security [81].

6) Extreme Learning Machine: The extreme learning machine (ELM) algorithm has been proposed in [89] for training a feed-forward neural network with a single hidden layer. The ELM algorithm is presented in this subsection related to ML, and not together with other deep neural network architectures in the DL section, mainly because it does not use gradient-based backpropagation for learning. The weights are learned using the Moore-Penrose generalized inverse. As compared to traditional feed-forward neural networks trained with backpropagation, the ELM is remarkably efficient and tends to reach a global optimum. Despite not being as accurate as the network architectures often used in DL, the ELMs can be used in dynamic problems that require real-time retraining of the network. In WSNs, the ELM is utilized for localization [67], anomaly detection [72], [73], and data aggregation [75], [76].

7) Artificial Neural Networks: The ANNs are designed to simulate a human brain and its capabilities [90] with neurons connected hierarchically in layers. The simplest ANN architecture consists of 3 types of layers: input layer, hidden layer, and output layer. The input vector applied to the input layer is responsible for the feature vector. The hidden layer is where the mapping function is modeled, usually containing the hidden neurons with an activation function. The output layer is where the system outputs the label based on the feature vector. The mapping function, represented by the weights that connect the neurons, is learned automatically using, for example, the backpropagation algorithm. The ANN can be used to solve anomaly detection [71] and energy harvesting [79]. It is noted in this review that the term DL is for the ANN architectures that contain at least two hidden layers using non-linear activation functions.

B. *Unsupervised Learning*

In unsupervised learning, the dataset consists of a collection of unlabeled examples. Contrarily to supervised learning, each example contains only the feature vector (input) in an unsupervised learning dataset. The goal of an unsupervised learning algorithm is to create a model that learns by itself to extract features on the input to either transform it into another vector or into a value that can be used to solve a practical problem. The result of unsupervised learning is a model with a priori probability distribution $p_X(x)$ over input data x, whereas supervised learning attempts to find a conditional probability distribution $p_X(x|y)$ conditioned on the label y. Unsupervised learning algorithms are usually used to solve problems associated with clustering, dimensionality reduction, and outlier detection. In the clustering, the objective is to group feature vectors according to their characteristics. In dimensionality reduction, the output vector is a feature vector with fewer dimensions than the input vector. In the outlier detection, the model should predict how different a feature vector is from the average example in the dataset. This system can be used for fraud detection in credit card purchases. Unsupervised learning algorithms used in WSNs are briefly described as follows.

1) K-Means: The K-Means clustering algorithm [91] is used to assign the unlabeled examples to k different clusters. Each cluster can be associated with a label. At the beginning of the learning, k centroids are defined randomly in the feature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

space. The distance between each example and each centroid is defined according to a distance metric like Euclidean distance. Each example is assigned to its closest centroid. The positions of the centroids in the feature space are updated iteratively based on the examples assigned to them. The K-Means algorithm is stopped when the examples are always assigned to the same centroid. It is noted that the initial random positions of the centroids affect the final model. Also, the number of labels (centroids) k should be tuned by the data analyst to gain better performance.

2) Density-based Spatial Clustering of Applications with Noise: While the K-Means algorithm is centroid-based, the density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering algorithm [92]. In the DBSCAN algorithm, two hyperparameters ϵ and n should be defined by the data analyst. For each example assigned to a cluster, the number of other examples (neighbors) with a distance less or equal to ϵ is calculated. If this number is equal to or greater than n, they are assigned to the same cluster. The clusters are expanded using the same method until there are no more examples assigned to it. All examples in the dataset are examined to form the final clusters. The DBSCAN is useful for outlier detection problems. If an example has less than n neighbors with a distance less or equal to ϵ it is considered an outlier. Recently, anomaly detection solved by the DBSCAN was presented for WSNs in [86].

3) Principal Component Analysis: The principal component analysis (PCA) [93], [94] is one of the most popular algorithms used for dimensionality reduction. The PCA defines a new coordinate system based on the variance of the data. The first axis is defined in the direction of the largest variance in the data, the second axis is defined orthogonally to the first one in the direction of the second largest variance in the data. This process of finding an additional axis is continued until all the relevant axes are found. To apply the PCA for dimensionality reduction, the most relevant axes (largest principal components) are chosen and the data are then projected in this space. For example, a 3-Dimensional space can be reduced to a 2-Dimensional space if the two largest principal components are kept.

4) Uniform Manifold Approximation and Projection: The concept of the uniform manifold approximation and projection (UMAP) algorithm [95] is also used for dimensionality reduction, similar to the PCA. In the UMAP algorithm, a similarity metric is defined to compare two examples of the dataset. This similarity metric can take into consideration the distance between the examples and the density around them. The similarity metric is used to construct a high-dimensional graph that represents the data. The UMAP algorithm tries to construct a low-dimensional graph that tries to be as similar as possible to the structure of the high-dimensional graph. This process is executed, using fuzzy sets and gradient descent optimization.

C. Semi-supervised Learning

To create a big dataset of labeled examples is a time-consuming and computationally expensive task. In most real world applications, the dataset is a combination of labeled and unlabeled examples. When the dataset contains both labeled and unlabeled examples, $((x_1, y_1), ..., (x_l, y_l), x_{l+1}, ..., x_{l+u})$, where y_i is the label of input x_i, a semi-supervised learning algorithm is used. Usually, the number of labeled examples is much smaller than the number of unlabeled examples. These algorithms contain characteristics of both supervised and unsupervised algorithms. In most applications, the goal of a semi-supervised learning algorithm is to learn a model that can predict labels based on a feature vector, similarly to supervised learning. On the other hand, unlike supervised learning, the information extracted from unlabeled examples is also used in semi-supervised learning to improve the final performance of the trained model. As a subgoal to learn the final model, it is desired to predict the label of the unlabeled examples in the dataset. Semi-supervised learning algorithms are gaining attention lately, especially for data augmentation [96], [97], and data privacy [98] . A semi-supervised algorithm is also used for WSNs to address the localization [99].

D. Reinforcement Learning

The RL algorithms are used for problems in which the model has the ability to make sequential decisions. These problems can usually be modeled as a Markov decision process (MDP). The MDP is completely identified with a tuple $(S, A, P^a_{s,s'}, R_a)$, where S is a set of states, A is a set of actions, $P^a_{s,s'}$, is the matrix of transition probability from current state s to next state $s^{'}$, and R_a is the immediate reward to an action a. In an RL framework modeled as an MDP, an agent can interact with an environment by gathering information (observing the environment state) and by making decisions (taking actions) that affect the environment. Good decisions taken by the agent receive positive feedback (reward) from the environment. The resemblance between the RL framework and human learning makes it flexible to be used in a variety of applications. The goal of RL is to learn good actions by trial-and-error method in order to maximize rewards received in the future. A classical example of a problem in which RL algorithms are well suited is game environments. In a game environment, the agent(in this case the human) is trying to take actions, usually using a video game controller, to maximize the final score in an episode or to win against an opponent. An RL algorithm tries to learn a mapping function, called policy $\pi(a, s)$, to choose the desired action a based on an environment state s. Lately, to handle environments with large state spaces, the ANNs are used by RL algorithms to learn the mapping function between state and action, which is called deep RL (DRL). The DRL algorithms that achieve breakthrough results lately are presented in Section IV. The RL solves various challenges in WSNs such as coverage [100], anomaly detection [101], routing [102]–[104], and mobile sink [105].

III. Design of Wireless Sensor Networks with Machine Learning

In this section, the use of ML techniques in WSNs is explored. In each subsection, the advantages of selecting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

specific ML techniques to address the issues in WSNs are discussed. The features of existing ML techniques fit to each application of WSNs are presented in TABLE I.

A. Localization and Target Tracking

Localization is to determine the locations of sensor nodes in the environment of interest [106]. In [67], a node localization algorithm is proposed, using kernel ELM based on hop-count quantization. The kernel ELM is trained with the inputs of hop-counts between unknown nodes and anchor nodes. The locations of anchor nodes are known as reference information. The proposed algorithm improves the accuracy of node localization as compared to localization based on the SVM. A cascaded two-stage ML approach has been developed using the k-NN algorithm to improve indoor localization accuracy [68], targeting internet-of-things (IoT) applications. In the first stage, the type of environment is identified by using the k-NN algorithm. In the second stage, appropriate radio frequency features are selected by the k-NN algorithm, considering the type of environment. Improved accuracy of localization is verified with experimental results. In [99], a semi-supervised polynomial manifold learning has been introduced for localization in WSNs. The physical locations of unknown nodes can be obtained directly by using multi-dimensional regression without coordinate translation. As a result, reduced computational complexity and higher localization accuracy are obtained as compared to existing methods using range-based manifold localization.

Target tracking is the process of sensor node(s) to detect the future location of a dynamic object. Target tracking involves single or multiple sensors. Tracking with a single sensor node consumes less energy, but it is less accurate. Tracking with multiple sensors is more accurate at the cost of increased computational overhead. To reduce the computational overhead and track the target in dynamic situations, ML techniques have been used. A target tracking algorithm using the SVM with Kalman filter (KF) is presented in [69]. The SVM is utilized to obtain an initial position estimate of the target based on the received signal strength. This estimated position is modified by the KF. As compared to existing algorithms, the proposed target tracking algorithm improves tracking accuracy and stability. In [70], a Bayesian algorithm for localization and tracking of a moving target in WSNs is proposed. In the proposed method, an improved Bayesian algorithm is trained to obtain probability information on the predictive location of the target, which is used to update the weight of measurement.

B. Coverage and Connectivity

Coverage represents how efficiently a sensor network monitors the area of interest with proper node deployment [107]. Coverage is an issue that needs to be addressed before deployment because it is the basic step for subsequent routing and data transmission. To increase coverage, methods to reduce energy consumption and increase transmission power in WSNs are critical. Connectivity represents the ability of sensor nodes to communicate with each other in WSNs. This means that every node in a WSN can receive sensing data from neighbor nodes and send sensing data to another neighbor node or sink node. Increased connectivity with high energy efficiency can be achieved by selecting when sensor nodes are active (awake) or inactive (sleeping) [108] and by using an artificial "bee" colony algorithm [109]. The use of ML techniques can allow the proper determination of the number of sensor nodes necessary to cover an area of interest or the proper classification of nodes that should be connected and disconnected.

To recover coverage holes, a new game theory approach based on RL is proposed in [100]. In this game theory, each sensor node takes an action of changing the node location and transmission power. The action is taken based on the state of the network to reduce coverage gaps.

C. Anomaly detection

Anomaly detection is to determine inconsistencies in the sensing data. Anomaly detection can be defined as detecting anomalies caused by attacks, faults due to failures, and unusual events [110]. In WSNs, there can be data loss caused by abnormal attacks while the sensor nodes are transmitting the sensing data to the desired destination, e.g. the sink node. The possible types of attacks are misdirection attack [111], blackhole attack [112] , wormhole attack [113], sinkhole attack [114], and hybrid anomaly [115]. ML approaches for anomaly detection can improve the accuracy of detection and minimize communication overhead and complexity. Some WSNs can be deployed in hostile, uncontrolled, or unattended environments, which leads to additional failures of data communication, battery operation, or system operation [116].

In [71], an ANN-based forecast model for outlier detection is proposed. In this model, the temperature data is considered as an outlier when the error of the predicted temperature data is much greater than the expected value. The proposed model provides a prediction of the anomaly occurrence with high accuracy. A hybrid intrusion detection system with the cross-layer rules and ELM algorithm is implemented in [72] for WSNs. In the proposed system, the cross-layer rules and ELM are used at sensor nodes and the base station, respectively, to detect the attacks. In [86], a hybrid intrusion detection system using supervised and unsupervised ML techniques is investigated. The proposed algorithm solves the intrusion problem by detecting known and unknown intrusions with misuse detection and anomaly detection subsystems using the random forest and enhanced-DBSCAN clustering algorithms, respectively. In [101], an RL algorithm for detection and prevention of misdirection attacks in WSNs is studied. Each node considered as an agent takes an action related to the packet transmission based on the node state to receive a reward which gets bigger when the distance from the sink node is reduced. In [73], a hybrid technique using the ELM and KF for predictive classification of faulty data is proposed. Using this hybrid technique, the faulty data is classified by the ELM classifier. The features of each data in the entire dataset are extracted by the KF.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

TABLE I. ML techniques for WSNs

ML Technique	Studies	Design Issue	Control Scheme	Mobility	Remarks
Regression, ANN	[79]	Energy Harvesting	Distributed	Static	Improved prediction accuracy
ANN	[71]	Anomaly Detection	Distributed	Static	Improved detection accuracy
Bayesian	[70]	Target Tracking	Centralized	Static	Reduced computational complexity
ELM	[72]	Anomaly Detection	Centralized	Static	Improved detection rate
	[73]	Anomaly Detection	Centralized	Static	Improved detection accuracy
	[75]	Date Aggregation	Centralized	Mobile	Improved network lifetime
	[76]	Date Aggregation	Distributed	Static	Improved calibration accuracy
	[67]	Localization	Distributed	Static	Improved localization accuracy
Decision Tree	[77]	Congestion Control	Distributed	Static	Improved prediction accuracy
k-NN	[68]	Localization	Centralized	Static	Improved localization accuracy
k-NN, Random Forest, Naive Bayes	[80]	Security	Centralized	Static	Reduced computational complexity
Decision Tree, k-NN, SVM	[74]	Date Aggregation	Distributed	Static or Mobile	Improved network lifetime
SVM	[78]	Congestion Control	Distributed	Static	Improved classification accuracy
	[81]	Security	Distributed	Static and Mobile	Reduced computational complexity
	[69]	Target Tracking	Centralized	Static	Improved tracking accuracy and stability
Random Forest + DBSCAN	[86]	Anomaly Detection	Centralized	Mobile	Improved detection accuracy and rate
Semi-supervised	[99]	Localization	Centralized	Static	Reduced computational complexity
RL	[100]	Coverage	Distributed	Mobile	Energy-efficient
	[101]	Anomaly Detection	Distributed	Static	Improved network lifetime
	[102]	Routing	Distributed	Static	Improved network lifetime
	[104]	Routing	Distributed	Static	Energy-efficient
	[103]	Routing	Distributed	Mobile	Energy-efficient
	[105]	Mobile Sink	Centralized	Mobile	Improved network lifetime

D. Routing and Congestion Control

In WSNs consisting of randomly deployed sensor nodes, the transmission of the sensing data to the sink node requires routing through relay sensor nodes. Routing of sensor data through relay nodes is called multi-hop transmission. Routing is the process of determining a path for data transmission in WSNs. The purpose of a routing protocol is to save the operational energy of sensors and consequently prolong the network lifetime [117]. ML techniques can be used to select optimal cluster heads and the optimal route without reprogramming in response to the changes in the environment and can help reduce the energy consumption of sensor nodes. Congestion in WSNs indicates that the amount of data to be transmitted exceeds the transmission capacity of sensor nodes [118]. It occurs at node level due to the high packet arrival rate and link-level because of the lower transmission rate. Since it affects the performance of WSNs, congestion control is important in WSNs. To this end, ML techniques predicting congestion situations and finding optimal paths are used for congestion control.

An RL technique named state–action–reward–state–action (SARSA) learning is introduced for intelligent routing in [102]. Clustering combined with the SARSA is adopted for intelligent routing to balance energy consumption. The sensor nodes in the WSN are trained in order to decide how many route requests to accept by considering the drain rate and residual energy. The proposed technique achieves lower energy consumption, higher stability, and prolonged network lifetime. An efficient routing protocol based on distributed multi-agent RL (MARL) is developed in [103] for underwater optical WSNs. In this work, the energy consumption of nodes, link stability, and quality of data transmission are considered in the reward function. With this reward function, the proposed routing protocol selects the route to the sink node. The proposed routing protocol reduces the energy consumption of sensor nodes and improves the packet delivery ratio in WSNs. A congestion-avoiding routing protocol based on the RL is reported in [104]. This protocol uses an RL technique to reduce the end-to-end delay and energy consumption of sensor nodes. This protocol finds the optimal route to the sink node by exploring lots of hop-by-hop combinations with a reward function to reduce congestion and energy consumption.

In [77], the solution of the congestion control problem is addressed with a DT for 5G IoT environments. The DT based model is trained to classify the dataset and predict the optimal alternative node. As a result, the proposed model can find the optimal path to the sink node. An algorithm for transmission rate control is proposed in [78], using the SVM to address the issue on congestion in WSNs. The SVM parameters are tuned by optimization algorithms like differential evolution and grey wolf optimization to reduce mis-classification error.

E. Data Aggregation

Data aggregation is the process of aggregating the data obtained from multiple sensor nodes. The goal of data aggregation is to reduce redundant transmission and communication overhead. Efficient data aggregation reduces the energy consumption of sensor nodes and prolongs the network lifetime. Data aggregation techniques can be divided into flat network-based, cluster-based, tree-based, and grid-based ones [119]. Utilizing ML techniques for data aggregation is helpful for selecting cluster heads and for dimensionality reduction of the sensor data.

A method achieving energy-focused clustering and data reduction is proposed in [74], using fuzzy logic and ML classifiers. In this method, ML classifiers such as the DT, k-NN, and SVM are applied to sensor nodes grouped in clusters to identify the similarity of sensor data and classify the data according to this similarity. The sensor nodes are clustered by fuzzy logic. The number of data transmissions from nodes to cluster head nodes is reduced by utilizing the classified data. The proposed method balances the energy consumption of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

sensor nodes and improves the network lifetime. The ELM for data fusion in the mobile heterogeneous WSN is used in [75] to reduce redundant data transmissions and improve the efficiency in data fusion. The ELM extracts the features of collected data and combines them with the clustering route to reduce data transmissions. In [76], a constrained ELM with KF is presented to calibrate drift in the sensor data. The constrained ELM is trained to predict the data of the target sensor by using the sensor measurements obtained from its neighbor nodes. This predicted data and de-noised data are fed to the KF to adjust the drift.

F. Mobile Sink

In a WSN, nodes close to the sink node consume much more energy due to more data transmissions to the sink node, as compared to other sensor nodes. To alleviate this problem, the use of a mobile sink is introduced. The basic concept of mobile sink is that a mobile sink visits a few sensor nodes or points called rendezvous points in the sensor network to gather data [120]. Multiple mobile sinks can be used to avoid time delays that occur with a single mobile sink. In large scale WSNs, scheduling the route of mobile sink nodes is important to reduce time delay and energy consumption of sensor nodes. Using ML techniques to schedule mobile sink nodes can provide benefits in finding optimal rendezvous points and the optimal path of each mobile sink node.

In [105], genetic algorithm and MARL technique are used to establish the shortest tour length of a mobile sink in an environment with multiple obstacles. In this way, an efficient algorithm for data gathering in WSNs in an environment with obstacles is proposed. The proposed approach combining genetic algorithm and MARL allows the construction of the tour in two phases. Several short tours are constructed in the first phase, using the MARL. In the second phase, the genetic algorithm is used to determine the optimal tour based on the short tours found in the first phase. The proposed algorithm can improve the performance in terms of energy consumption of sensor nodes and network lifetime.

G. Energy Harvesting

In WSNs, the network lifetime depends on the battery capacity of sensor nodes. To prolong the lifetime of a sensor node, energy harvesting techniques are introduced. Energy harvesting is the process of converting nature energy such as radio frequency (RF), wind, and solar energy to electrical energy for sensor nodes in WSNs [121]. Energy harvesting can be carried out with and without energy storage. Using ML techniques for energy harvesting is helpful for predicting the amount of harvested energy and balancing the energy consumption of sensor nodes.

In [79], an ANN and a linear forecaster with an enhancer based on linear regression are used to determine the optimal scheduling of RF energy harvesting. With these ML algorithms, sensor nodes can reliably harvest RF energy from both intended and unintended sources in a dynamically varying environment.

H. Security

Some WSNs deal with security-sensitive data in hostile environments with an unattended fashion. In such situations, it is important to adopt security techniques for the WSNs. The security techniques can be used for data authentication, data confidentiality, data integrity, and data freshness [122]. However, traditional techniques for network security, such as user authorization, are not suitable for these applications due to limited resource and computing constraints of the WSNs [123].

In [80], ML classification techniques, such as Random Forest, k-NN, and Naive Bayes, are used for the access gateway to detect IoT malware network activity. Results of performance evaluation with these techniques show that the highest classification accuracy can be obtained with the k-NN technique. A privacy-preserving SVM training scheme is presented in [81] for IoT data, requiring only two interactions in one iteration without the need of a trusted third-party. This approach significantly reduces the computational complexity and communication overhead as compared to conventional SVM.

IV. Deep Learning Architectures for Wireless Sensor Networks

In this section, an overview of DL techniques that appeared in the literature in recent years is given. In TABLE II, DL techniques fit to each application for WSNs are summarized.

The definition of DL is made in relation to ANNs. The neural networks that consist of an input layer, more than one hidden layer with non-linear activation function, and an output layer are typically considered as deep neural networks (DNNs). Each layer includes several units called neurons. A neuron receives several inputs and performs a weighted summation over its inputs. The resulting sum goes through an activation function to produce an output. The weights associated with each neuron are optimized during the training process. These weights are usually trained by backpropagation in order to optimize a loss function. The process of training DNNs is called DL, even though the term is used in a broad way nowadays. It is important to mention that the non-linear activation functions are really important for the DNNs to be able to solve complex non-linear problems. Without the non-linear activation functions, even with multiple hidden layers, the mapping function would still be linear, and there would not be any difference between using an ANN and an SVM with a linear kernel.

A. Convolutional Neural Networks

When the input of the network is an image, traditional ANNs with feed-forward layers have problems in learning spatial features. To specific tasks, such as image recognition and image classification, convolutional neural networks (CNNs) have been proven very effective. Though the concept of CNNs is old, considering the works that date back to the 70s [143], the breakthrough results in this area were obtained lately by the AlexNet architecture in 2012 [144]. A CNN architecture similar to the AlexNet architecture is shown in

TABLE II. DL techniques for WSNs

DL Technique	Studies	Design Issue	Control Scheme	Mobility	Remarks
CNN	[124]	Routing	Centralized	Static	Improved network lifetime
	[125]	Routing	Centralized	Static	Improved network lifetime
	[126]	Data Aggregation	Centralized	Static	Energy-efficient
	[127]	Anomaly Detection	Distributed	Static	Reduced computational complexity
CNN+LSTM	[128]	Data Aggregation	Distributed	Static	Improved prediction accuracy
LSTM	[129]	Data Aggregation	Distributed	Static	Improved prediction accuracy
	[130]	Data Aggregation	Distributed	Static	Improved imputation accuracy
	[131]	Data Aggregation	Distributed	Static	Reduced communication overhead
AE	[132]	Localization	Centralized	Static	Improved localization accuracy
	[133]	Routing	Centralized	Static	Improved estimation accuracy
	[134]	Data Aggregation	Distributed	Static	Energy-efficient
GAN	[135]	Security	Centralized	Static and Mobile	Improved security
DQN	[136]	Routing	Centralized	Static and Mobile	Reduced outage probability
	[137]	Routing	Distributed	Static	Improved network lifetime
	[138]	Spectrum Access	Distributed	Static	Improved channel utilization
	[139]	Mobile Sink	Centralized	Mobile	Energy-efficient
	[140]	Mobile Sink	Centralized	Mobile	Energy-efficient
DDPG	[141]	Target Tracking	Distributed	Static	Improved tracking accuracy
	[142]	Energy Harvesting	Distributed	Static	Improved net bit rate

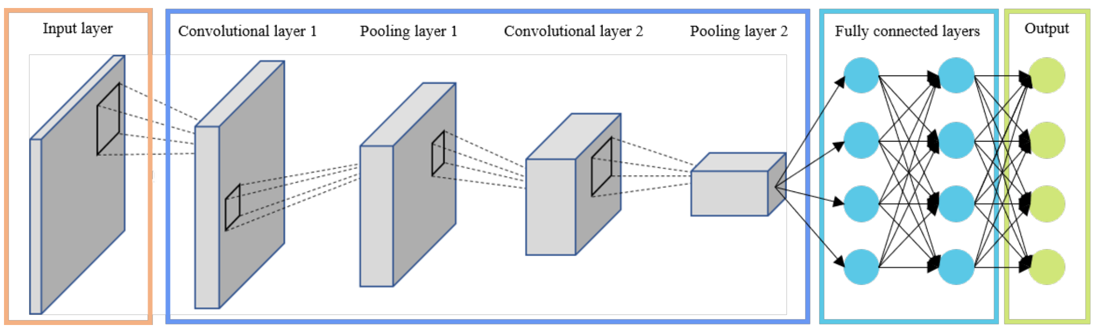

Fig. 2. Architecture of convolutional neural network.

Fig. 2. The idea behind the CNN architecture is to use filters that are responsible for the implementation of ”convolution” operations. The filters are convolutional layers to handle the 2-Dimensional (grayscale image) or 3-Dimensional (color image) information of the input image. The primary purpose of the adoption of convolutional layers is to extract features that preserve the spatial relationship between pixels of the input image. The filters slide over the whole image to produce a feature map that is learned automatically during training. If multiple convolutional layers are connected subsequently, the first set of convolutional layers usually learn low-level feature maps, e.g., borders, while the last set of convolutional layers learn high-level feature maps, e.g., objects.

Two additional operations that are important for the performance of CNNs occur with the non-linear activation function, usually a rectified linear unit (ReLU), and pooling layers. The non-linearity of the ReLU is used for every convolution operation. In this way, feature maps that represent non-linear mapping of convolutional layers can be learned. Pooling layers are used for reducing the dimensionality (down-sampling) of each feature map. The pooling layer retains only the most important spatial information in the feature map. For example, it can be implemented by only getting the maximum or the average of values in a specific region. As stated before, CNNs are really powerful when the application needs to handle image data. In the following, research works dealing with the CNNs for WSNs are briefly reviewed.

The ability to successfully cluster sensor nodes to implement cluster-based routing can reduce energy consumption in WSNs. Three techniques are proposed using traditional ML to address this issue. In the low energy adaptive clustering hierarchy (LEACH) method [145], clusters are formed based on the distance between the sensor node and cluster head. In the fuzzy logic based cluster formation protocol (FLCFP) method introduced in [146], fuzzy logic is used to cluster nodes in a network. In the hybrid energy-efficient and distributed clustering (HEED) method [147], the cluster is formed by using a probabilistic model. However, the performance of these techniques needs to be enhanced in terms of energy optimization and improved accuracy. To improve the performance in cluster-based routing, a neuro-fuzzy rule-based cluster formation protocol (FBCFP) using CNNs is proposed in [124]. With this protocol, the neural network is trained with trace data obtained from past and current transmissions. Weights are adjusted by using the current data and by applying fuzzy rules. The proposed protocol extends network lifetime compared with the LEACH, FLCFP, and HEED methods and also reduces the total energy consumption in WSNs.

Resilient routing protocols can address the problem of rerouting after attacks in WSNs. In [148], a distributed routing algorithm for networks fabricated on a chip is presented. In [149], a local strategy is designed, which enables networks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

with random topologies to automatically re-route in the situation of failures in sensor nodes. For software-defined networks, a dynamic attack-resilient routing algorithm to optimize multipath routing is developed in [150]. Despite the importance of link reliability prediction in routing design, few studies have dealt with the use of link prediction for resilient routing in WSNs. In [125], resilient routing algorithms for WSNs are proposed, using link reliability. This algorithm includes a DL based link prediction model that utilizes the Weisfeiler-Lehman kernel, a dual CNN to extract features, and label subgraph. This algorithm enhances the network's survivability and resilience with a significantly extended life cycle.

Excessively redundant data in sensor nodes that are unevenly distributed increases transmission delay and energy consumption of sensor nodes. In order to solve this problem, data fusion techniques for WSNs are proposed. The traditional algorithms for data fusion use neural networks trained with backpropagation and self-organizing feature map [151], [152]. Since these traditional algorithms have the problems of overfitting and local optima, recent DL architectures have been applied for data fusion in WSNs. To improve the performance of data fusion, set theory integrated with an improved CNN architecture is proposed in [126] as a novel information aggregation algorithm. The data features, which are extracted from the CNN, are sent to the sink node through cluster heads, so as to reduce the quantity of data transmission and extend the network lifetime. The proposed algorithm decreases the energy consumption of sensor nodes and improves the efficiency of data aggregation.

The CNN architecture is also applied to event detection which is a basic function of WSNs. In [127], a novel application using 1-D CNNs is presented for WSNs to detect structural damage. This work involves training an individual 1-D CNN for each wireless sensor node in WSN without the need for data transmission and synchronization. The proposed method for damage detection operates directly on the raw ambient vibration condition signals without any filtering or preprocessing.

B. Recurrent Neural Networks

In many tasks, prediction of future data is dependent on samples acquired in previous timesteps so that analysis of a sequence of inputs is needed for efficient prediction. In such applications, a feed-forward neural network is not applicable, since it does not assume any dependency on previous inputs. Recurrent neural network (RNN) shown in Fig. 3 is developed to handle sequential data, e.g. speech or text, or time-series data, e.g. sensor data, with various lengths. The input to an RNN consists of both the current sample(Input in Fig. 3(a)) and a vector related to the previously observed samples(h_1 in Fig. 3(a)). In other words, the output of an RNN at timestep $t-1$ affects the output at a timestep t. Each neuron or cell is equipped with a feedback loop that returns the current output as an input for the next step. This structure allows each neuron to have an internal memory that keeps the information of the computations from the previous input. Figure 3(d) presents the unrolled version of RNN. To train the

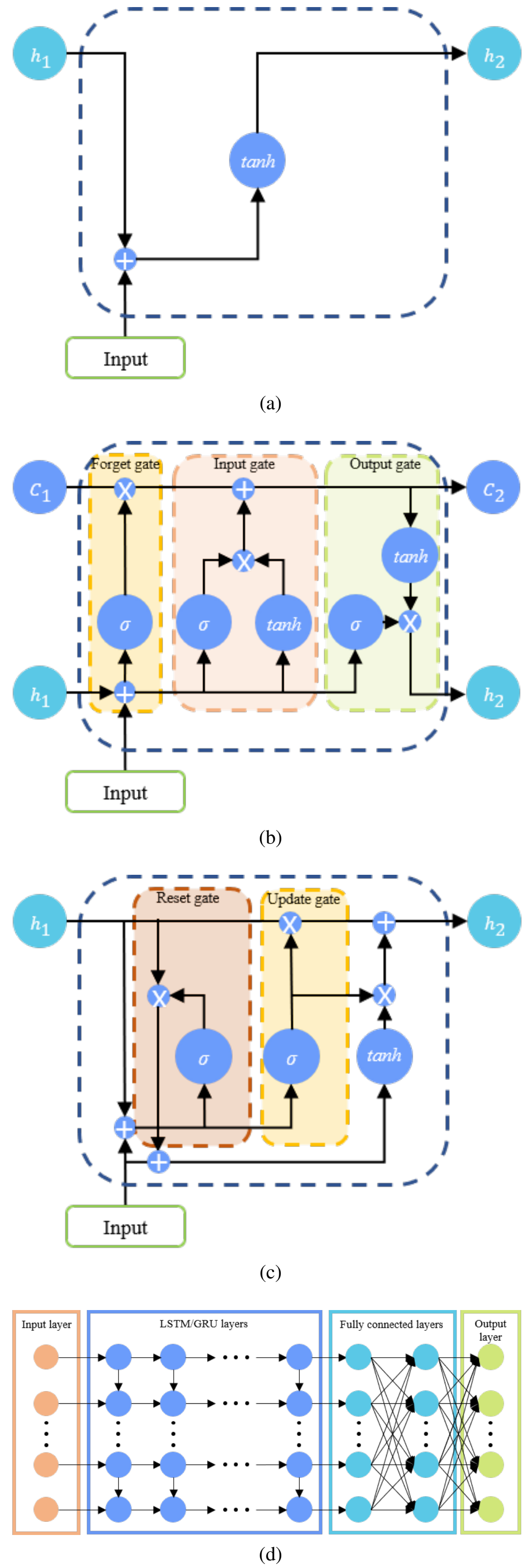

Fig. 3. RNN/LSTM/GRU cell and architecture of RNN/LSTM network/GRU network:(a) RNN cell;(b) LSTM cell;(c) GRU cell;(d) architecture of RNN/LSTM/GRU.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

RNN, an extension of the backpropagation algorithm, called backpropagation through time (BPTT) [153], is used. The error in the current time step using the BPTT algorithm is propagated back to the RNN state in previous time steps.

1) Long Short-Term Memory: The long short-term memory (LSTM) architecture(network) is an advancement of the basic RNN architecture for long term prediction (output dependent on an input that happened ahead of multiple time steps). The LSTM architecture introduced by [154] is presented in Fig. 3(d). The LSTM architecture uses different types of gates in a memory cell(LSTM cell in Fig. 3(b)) to model the mapping function and a memory to store relevant data. In addition to a feedback loop similar to the RNN architecture, each neuron in the LSTM architecture, also called a memory cell, consists of three gates: forget gate, input gate, and output gate. These gates control the memory cell and implement the process that makes the LSTM architecture remember relevant inputs that appeared in the past while forgetting irrelevant inputs that also appeared in the past. The forget gate is used to erase old data that are not relevant anymore. When the forget gate is activated by multiplying the data by a value 0, the cell forgets its last content. When the output gate is set to 1, other connected cells can write new data to that neuron. If the input gate is set to 1, the connected neurons can read the content of the neuron.

An important difference of the LSTM architecture as compared to the basic RNN architecture is that LSTM cells utilize forget gates to actively control the cell states. The gates can use sigmoid or tanh as their activation functions. In fact, these activation functions may cause the problem of vanishing gradient during the BPTT in the training phase with a long temporal sequence of inputs. When the input data are characterized by a long time dependency, the LSTM architecture performs better than basic RNN.

In WSNs, the problem of missing data is common. To avoid this, algorithms for data imputation have been developed [155]. In recovering missing sensor data, DL techniques such as autoencoder [156] and deep stacking network [157] have been used. These DL techniques require huge training data and are not effective for systems that collect only time-series data. To overcome this problem, a new sequence-to-sequence imputation model employing an LSTM architecture and a sliding window is proposed in [130] for recovering missing data in WSNs. The proposed model can provide stable data imputation with improved accuracy.

In the presence of cooperation between nodes, unnecessary data transmission causes data congestion and data loss. To avoid this situation, data prediction methods can be used. In [158], a data suppression mechanism has been proposed to predict data. To conserve the energy of sensor nodes, an aggregation model based on temporal data prediction is considered in [159]. In [160], a method based on the ELM to predict micro-climate data is proposed. Unlike these methods, a new prediction method named multi-node multi-feature utilizing the spatial-temporal correlation with bi-directional LSTM network is investigated in [129]. The bi-directional LSTM network extracts and learns the prediction features of sensor data. The proposed model achieves high prediction accuracy and reasonable prediction bias. In order to predict multi-step sensor data in WSNs, authors in [128] utilize bi-directional LSTM network and 1-D CNN. Parallel networks based on DL architectures are used to extract the features of sensor data for multi-step prediction.

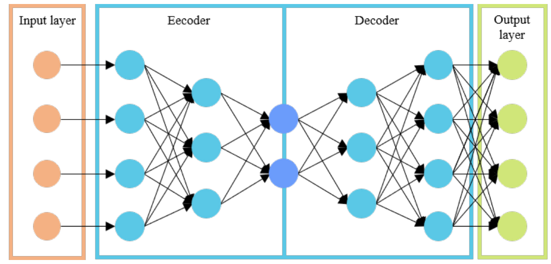

Fig. 4. Architecture of autoencoder.

In [131], a distributed data mining model based on DL is presented to achieve energy efficiency and optimally balanced loading at the fusion center or sink node of the WSN. The proposed model includes an LSTM network and reduces the overhead at the fusion center along with reduced data transmissions. The proposed model based on the LSTM network is tested with a large set of experimental data while changing the number of nodes in hidden layers and signaling intervals. The amount of energy needed to transmit data by the proposed model is considerably lower than the energy needed to transmit actual data.

2) Gated Recurrent Unit: The gated recurrent unit (GRU) architecture consisting of GRU cell shown in Fig. 3(c) is proposed in [161] as another variant of the RNN. The GRU architecture is also used to forecast sensor data, taking into account the randomness of the sensor data. The GRU architecture is developed to reduce computational time and cost as compared to the LSTM architecture. Similarly to the LSTM cell in the LSTM architecture, the GRU cell in the GRU architecture also has memory capabilities and controls the information flow without using a memory unit. The number of GRU layers and the maximum length of input data to the input GRU layer are typically determined by trial-and-error method. When the number of GRU layers is not properly set, inherent parameters of the GRU network grow or shrink exponentially and eventually become improper for the training data. The number of inputs and outputs of the GRU network can be determined through the evaluation of root-mean-squared-error. Replacing the LSTM architecture with the GRU architecture in WSN applications might increase the network lifetime due to reduced computational time.

C. Autoencoder

An autoencoder (AE) is presented in Fig. 4. The AE aims to encode the input into a new representation (latent variable) by reducing its dimensionality and then reconstructing the input from the latent variable while keeping its most important characteristics [162]. The AE consists of two main components: an encoder and a decoder. The encoder receives the input and transforms it into the new representation. The decoder receives the output of the encoder and transforms it into a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

reconstruction of the original input. The AE is trained with the objective of minimizing the reconstruction error, i.e., how similar the input and output are.

There are several variations and extensions of the AEs. Denoising AE, contractive AE, stacked AE, sparse AE, and variational AE are some examples. The AEs are used particularly for problems that can be solved by unsupervised learning such as dimensionality reduction, image compression, image denoising, feature extraction, and image generation. The AEs have been also applied to problems that can be resolved by transfer learning.

As the demand for location-based services in indoor spaces increases, much attention is being paid to fingerprint based indoor localization. For fingerprint based indoor localization, the ELM is utilized due to its high learning speed [163]. In [164], two kinds of ELMs are used for indoor localization. Authors in [165] presented semi-supervised deep ELM, which uses a large number of unlabeled data. These ELM methods use random weights and bias in their hidden nodes. To address the indoor localization, an autoencoder relying on deep ELM is presented in [132]. This algorithm takes advantage of the DL, ELM, and high level extracted features obtained by AEs. The extracted features by the AEs are utilized instead of random weight to improve localization performance.

In WSNs, effective link quality estimation can help the routing protocol and topology management. To estimate the link quality of WSNs, link characteristics [166], [167] and statistics of link data [168], [169] are considered. Information loss originated from erroneous link quality estimation and slow updating of link quality due to large statistical data can be critical for link quality estimation. A link quality estimator making use of stacked AE is presented in [133]. This estimator uses four stacked AE (SAE) models. The SAE1, SAE2, and SAE3 extract the asymmetric feature of signal-to-noise ratio, link quality indicator, and received signal strength indicator, respectively, for both uplink and downlink. The SAE4 integrates the asymmetric information obtained from the three SAEs.

The limited computing capacity of WSNs makes data collection difficult. To mitigate this issue, energy-efficient data collection algorithms suitable for big data are designed. For data collection in WSNs, compressed sensing theory taking sparsity as a key condition is employed due to its computational asymmetry and compressibility [170], [171]. However, the group of transform bases used in compressed sensing, such as discrete Fourier transform, discrete cosine transform, and discrete wavelet transform, are usually empirical and thus cannot always transform the signal into the sparsest one. In order to solve this problem, a data collection model based on denoising AE has been proposed in [134]. The denoising AE is trained by using the historical sensor data. By utilizing the trained weight matrices, each sensor node can compress the sensor data before transmitting it to the sink node. Then, the compressed data received in the sink node is reconstructed to the original sensor data. This method reduces the energy consumption of sensor nodes and increases the data compression rate as well as data reconstruction speed and accuracy.

D. Generative Adversarial Networks

Figure 5 depicts the concept of the generative adversarial network (GAN). The GAN, introduced in [172], is a neural network architecture used to produce high-quality synthetic data. The GAN architecture is divided into a generator network and a discriminator network. The generator network produces synthetic examples, e.g. artificial images, from random noise using the data distribution learned from the training dataset. The discriminator network can be interpreted as an adversary to the generator network and thus it tries to distinguish real data from fake input data coming from the generator network. In other words, the generative network is competing with the discriminator network. Both the generator and discriminator networks are optimized to improve their respective abilities. The goal of the generator network is to produce artificial examples close to the real examples so that the discriminator network is unable to detect the difference between them.

The GANs are trained in a way similar to unsupervised learning. This means that they do not have labeled examples to learn from. The objective function in GANs is based on minimax games, in which one network tries to maximize the value function while the other tries to minimize it. The generator plays by producing artificial data from random noise, trying to fool the discriminator. The discriminator receives both real data and artificial data and tries to discriminate one from the other. The discriminator network is important for the training of the generator network, because when it identifies an artificial example, it instructs the generator how the artificial example should be tweaked to look closer to the real example. The GANs have been used for diverse applications like drug discovery, data manipulation, and data security.

With WSNs being more extensively deployed recently, data security is becoming more important. To address data security issues, middleware, which is a bridge between WSNs and the end-user, is used in diverse WSN applications [173], [174]. However, most existing algorithms using middleware cannot ensure secure communication. In [135], a secure middleware for WSNs, featured by the utilization of GANs to deceive attackers with fake data, is presented. A generator network creates various attacks (fake) data and a discriminator network differentiates real data from the fake data to achieve strong security.

E. Networks for Deep Reinforcement Learning

The RL that learns by trial-and-error method is characterized by 3 main components, state, action, and reward. The main objective of the RL framework is to define which action to choose in a specific state. This could be achieved by a table, for example, that lists combinations of state, action, and Q-values. The Q-value represents accumulated rewards with appropriate discounts for future rewards. As alternative solution for replacing a table, a deep neural network can be applied to model the mapping function between state and action. Recent breakthrough results in video game environments [175] have brought attention to DRL algorithms, especially because of their flexibility and ability to handle enormous state space. In the next subsections, two DRL algorithms, deep Q-networks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

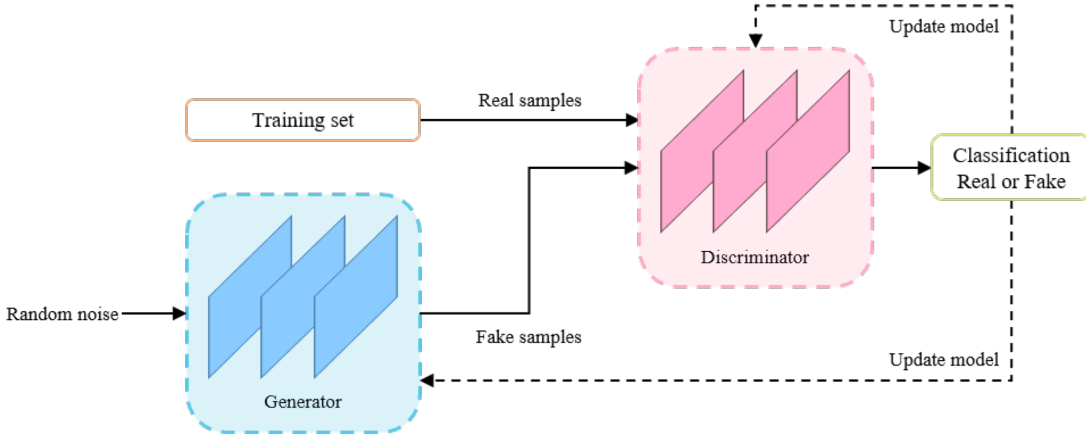

Fig. 5. Architecture of generative adversarial network.

and deep deterministic policy gradients (DDPG), which have been applied in WSNs, are presented.

1) Deep Q-Networks: Figure 6(a) shows the architecture of Deep Q-Network (DQN). The DQN proposed in 2015 [175] is a model-free RL algorithm for discrete action spaces (finite number of actions). The DQN algorithm maps a state to a Q-value for each possible action of the agent. In this way, the input of the network is the state of the environment and the output is the Q-values for each action. Typically, the action to be chosen in a specific state is the one with the maximum Q-value.

For training of DQN, episodes are generated using an ϵ-greedy policy. According to the ϵ-greedy policy, random actions are taken with a higher probability at the beginning of training to explore different strategies. At the end of the training, more actions are chosen using the neural network, as is expected that the agent is starting to be competent at the desired task such as playing a video game. To improve training performance, the experiences collected by the agent are stored in a replay buffer, so that it can be used multiple times during training. For improved use of experiences for training, sample presentation techniques such as batch prioritization [176] can be adopted. The network is trained to optimize a loss that encourages the approximated Q-values to satisfy the Bellman equation [177] and to maximize future rewards.

Cooperative communication is a key technology that can improve the performance of WSNs [178]. Since it utilizes cooperation between multiple relay nodes to share the transmission route, the relay selection problem is important in cooperative communication. In some research works, the RL has been applied to address the relay selection problem. To select an optimal relay, the Q-learning [179] and MARL [180] have been used. However, these algorithms do not consider the computational complexity and convergence speed. To increase the convergence speed, authors in [136] have proposed a relay selection scheme using the DQN architecture. The DQN architecture is trained to learn how to select a relay node, as an action, based on the state defined by channel state and mutual information at each time slot. This scheme allows better performance than traditional Q-learning based schemes in terms of outage probability, system capacity, energy consumption, and convergence time.

In underwater acoustic sensor networks (UASNs), the efficient routing protocol is primarily important. Nonetheless, the routing protocols effective to wireless signals in typical terrestrial WSNs cannot be applied to UASNs directly. Research works on routing protocol in UASNs appeared in the literature to improve energy efficiency [181], end-to-end latency [182], and network lifetime [183]. To improve the performance of UASNs, the DQN architecture is used in [137] to develop energy- and latency-aware routing protocol. The DQN model is trained to learn whether to forward a packet or not, as an action, based on the state of the current node information consisting of residual energy, depth, and neighboring nodes with a reward obtained when the transmission of the packet is done.

With spectrum scarcity, the development of dynamic spectrum access has become important for WSNs. In [184], a channel access scheme based on DRL for a single user is proposed. In [185], non-cooperative spectrum access addressed with a MARL framework has been addressed. To solve the dynamic spectrum access with a DQN of lower complexity, a deep Q-learning technique for spectrum access with multi-user is developed in [138]. In this work, the action is defined as the transmission of a packet through a channel, and reward is defined as an achievable data rate on the selected channel. The state can be described by current capacities of channels, selected channels in the previous time slot, and distribution of ACK signals over entire channels. At each time slot, each user selects the channel by a DQN in the manner that for the current state achieved data rate on the selected channel is maximized. In the proposed algorithm, the LSTM network is also used to give the ability to estimate the true state using the history of channel access.

In asynchronous WSNs, scheduling the mobile sink is challenging because of the changing number of active nodes over time. In an asynchronous and clustered network, scheduling with fixed halt-time, which is the time duration over which the mobile sink stays without moving, can cause a data loss, buffer overflow, and lower quality of services. Therefore, a new scheduling scheme for the mobile sink is required. In [139], joint mobile sink scheduling and data aggregation for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

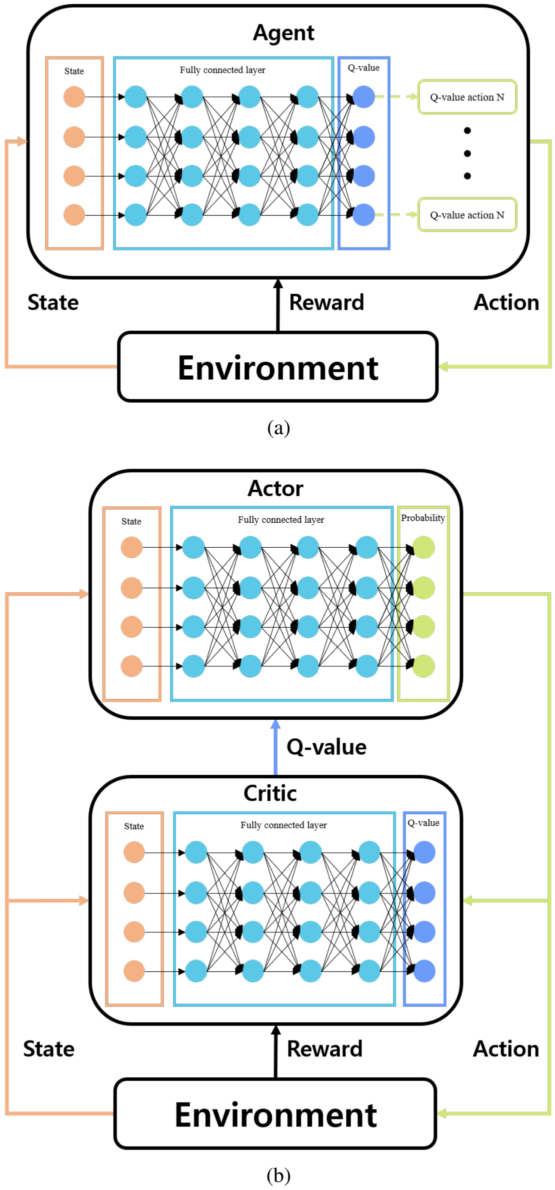

Fig. 6. Architecture of DQN and actor/critic networks of DDPG algorithm:(a)architecture of DQN;(b)architecture of actor/critic networks of DDPG algorithm

asynchronous WSNs is proposed by using DQN. The DQN is trained to learn how to determine whether to move or stay based on the current state. The current state is defined with current cluster data like the number of active nodes and the information about residual energy of the mobile sink. The proposed method minimizes the information loss and enables the mobile sink to perform the data gathering operation with limited energy consumption while maximizing network lifetime.

To improve the energy-efficiency of WSNs, network clustering, presence of mobile sink, and the variable rate of sensing can be considered [186]–[188]. However, independent developments of these techniques can introduce some network issues. An energy-efficient method considering node clustering and mobile sink scheduling with buffer management in the dynamic sensing rate scenario is proposed in [140] with the help of DQN. The DQN framework takes an action, move or not, based on the state, which includes the number of active nodes, information loss, and expected information loss, to minimize the data loss due to buffer overflow. In this work, minimization of the data loss can be considered when a reward is defined.

2) Deep Deterministic Policy Gradient: Figure 6(b) shows the architecture for the DDPG algorithm. The DDPG is a model-free RL algorithm for continuous action spaces (infinite number of actions), unlike the DQN algorithm that only can handle discrete action spaces. To handle continuous actions, the DDPG algorithm employs two networks: actor network and critic network. The actor network maps the state into action and the critic network maps state-action pairs into Q-values.

The DDPG algorithm works similarly to the DQN algorithm for continuous action spaces. However, the training procedure is different, because in the DDPG algorithm two networks should be trained. The critic network is trained to estimate the Q-value based on the received rewards. The training of the actor network is dependent on the training of the critic network. The gradients calculated in the critic network in relation to the input action are used to run backpropagation algorithm and train the actor network.

The use of sensor nodes in WSNs is effective for tracking objects in various environments. The existing decentralized tracking methods introduce unnecessary energy consumption due to the proactive sensing of sensor nodes. In [141], a dynamic adjustment of the activation area is presented using DRL. The rewards of the DRL algorithm are tracking accuracy and energy consumption. To maximize these rewards, two schemes employing DQN and DDPG algorithm are proposed. The DQN and actor/critic networks learn how to select the radius of the activation area, as an action, based on the state of the estimated motion vector of the vehicle and current radius of the activation area.

Frequent replacement of the battery of sensor nodes in remote environments limits the deployment of WSNs. A solution to this problem is the energy harvesting. Various strategies such as water-filling algorithm [189], MDP [190], and Lyapunov optimization [191] have been applied for energy harvesting. In [142], energy management rooted in energy harvesting is proposed using the DDPG algorithm. The objective of the energy management is maximizing the bit rate that can be taken as a reward. In the DDPG algorithm, the networks are trained to determine how much energy a transmitter node will consume, corresponding to the action, based on the state of bit error rate, maximum available energy of the node, and energy collected by energy harvesting. The proposed algorithm improves performance in terms of long-term average bit rate.

V. Impact and Limitations of using ML/DL Techniques instead of Conventional Approaches

The ML techniques, including DL techniques, have been successfully used for WSN applications such as classification, clustering, dimensionality reduction, feature extraction, and forecasting. Type of benefit obtained from the use of ML techniques for WSNs can be enumerated as reduced computational

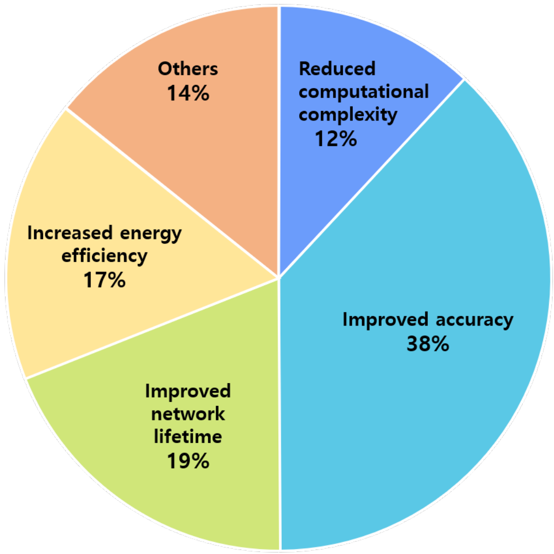

Fig. 7. Impact of using ML-based algorithms on performances of WSNs.

complexity, improved accuracy, improved(prolonged) network lifetime, increased energy efficiency, and so on. Figure 7 shows the result of survey with 42 reference papers dealing with ML techniques referred in this paper. Result of survey is presented in Figure 7 in terms of type of benefit and its proportion. Improved accuracy and prolonged network lifetime seem to be the two representative types of benefit obtainable from the use of ML techniques.

As drawbacks or limitations of ML techniques, large training time, sometimes excessively large, and large dataset to get acceptable performance are often mentioned in the 42 papers. Since large training time with large dataset is accompanied with large energy consumption, a trade-off between energy consumption and accuracy is required for resource-restrained WSNs. Other type of drawback mentioned in these papers is difficulty in selecting a proper ML technique for given WSNs.

VI. Conclusion

Advances in technologies related to the design, implementation, and deployment of WSNs enable more sophisticated operation of WSNs. In the case of traditional WSNs developed for specific tasks, it is hard to react in dynamically changing environments. To overcome this problem, ML techniques have been used lately in WSNs to deal with time-varying operating conditions. Among ML techniques available for WSNs, DL techniques, a subclass of ML techniques, have been adopted to efficiently address the implicit and long-term relationship between raw sensor data and processed data, rather than the instantaneous relationship between them. A range of benefits obtainable with ML techniques applied to WSNs can be described as reduced computational complexity, increased feasibility in finding optimal solutions, increased energy efficiency, and so on. On the other hand, it is found from our survey that large training time and large dataset to get acceptable performance are accompanied with large energy consumption which is not suitable for resource-restrained WSNs. Particularly, the applications of DL techniques are reviewed in details for WSNs. In this review, recent developments of DL techniques used to achieve enhanced performances in WSNs are presented. Combining DL with RL algorithms defines the class of DRL techniques. The use of DRL techniques that maps states into actions using deep neural networks is powerful and flexible enough to apply in various applications of WSNs. It is popularly being adopted in diverse applications and will be applied to virtually every application requiring multi-agent coordination and cooperation between sensor nodes to achieve enhanced performance in WSNs.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] I. Stojmenovic, *Handbook of sensor networks: algorithms and architectures*. John Wiley & Sons, 2005, vol. 49.

[4] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *2006 3rd annual IEEE communications society on sensor and ad hoc communications and networks*, vol. 1. IEEE, 2006, pp. 374–382.

[5] W. Yu and H. Li, "An improved dv-hop localization method in wireless sensor networks," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 3. IEEE, 2012, pp. 199–202.

[6] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2017.

[7] X. Liu, J. Yin, S. Zhang, B. Ding, S. Guo, and K. Wang, "Range-based localization for sparse 3-d sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 753–764, 2018.

[8] I. Ullah, Y. Shen, X. Su, C. Esposito, and C. Choi, "A localization based on unscented kalman filter and particle filter localization algorithms," *IEEE Access*, vol. 8, pp. 2233–2246, 2019.

[9] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, vol. 3. IEEE, 2003, pp. 1609–1614.

[10] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, "Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 836–848, 2009.

[11] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE sensors journal*, vol. 14, no. 3, pp. 636–644, 2013.

[12] Z. Sun, L. Wei, C. Xu, and Z. Lv, "An event-driven mechanism coverage algorithm based on sensing-cloud-computing in sensor networks," *IEEE Access*, vol. 7, pp. 84 668–84 679, 2019.

[13] B. Khalifa, A. M. Khedr, and Z. Al Aghbari, "A coverage maintenance algorithm for mobile wsns with adjustable sensing range," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1582–1591, 2019.

[14] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *International Conference on Dependable Systems and Networks, 2004*. IEEE, 2004, pp. 637–646.

[15] H.-b. Wang, Z. Yuan, and C.-d. Wang, "Intrusion detection for wireless sensor networks based on multi-agent and refined clustering," in *2009 WRI International Conference on Communications and Mobile Computing*, vol. 3. IEEE, 2009, pp. 450–454.

[16] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, and W.-C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[17] H. Huang, T. Gong, R. Zhang, L.-L. Yang, J. Zhang, and F. Xiao, "Intrusion detection based on k-coverage in mobile sensor networks with empowered intruders," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 109–12 123, 2018.

[18] W. Wang, H. Huang, Q. Li, F. He, and C. Sha, "Generalized intrusion detection mechanism for empowered intruders in wireless sensor networks," *IEEE Access*, vol. 8, pp. 25 170–25 183, 2020.

[19] J. W. Barron, A. I. Moustapha, and R. R. Selmic, "Real-time implementation of fault detection in wireless sensor networks using neural networks," in *Fifth International Conference on Information Technology: New Generations (itng 2008)*. IEEE, 2008, pp. 378–383.

[20] A. I. Moustapha and R. 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.

[21] C. Lo, J. P. Lynch, and M. Liu, "Distributed reference-free fault detection method for autonomous wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 2009–2019, 2013.

[22] Y. Peng, W. Qiao, L. Qu, and J. Wang, "Sensor fault detection and isolation for a wireless sensor network-based remote wind turbine condition monitoring system," *IEEE Transactions on Industry Applications*, vol. 54, no. 2, pp. 1072–1079, 2017.

[23] Y. Gao, F. Xiao, J. Liu, and R. Wang, "Distributed soft fault detection for interval type-2 fuzzy-model-based stochastic systems with wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 334–347, 2018.

[24] C. Schurgers and M. B. Srivastava, "Energy efficient routing in wireless sensor networks," in *2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No. 01CH37277)*, vol. 1. IEEE, 2001, pp. 357–361.

[25] J.-F. Yan and Y.-L. Liu, "Improved leach routing protocol for large scale wireless sensor networks routing," in *2011 international conference on electronics, communications and control (ICECC)*. IEEE, 2011, pp. 3754–3757.

[26] D. Zhang, G. Li, K. Zheng, X. Ming, and Z.-H. Pan, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 10, no. 1, pp. 766–773, 2013.

[27] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Ring routing: An energy-efficient routing protocol for wireless sensor networks with a mobile sink," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1947–1960, 2014.

[28] Y. Sun, W. Dong, and Y. Chen, "An improved routing algorithm based on ant colony optimization in wireless sensor networks," *IEEE communications Letters*, vol. 21, no. 6, pp. 1317–1320, 2017.

[29] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang, "A delay-efficient algorithm for data aggregation in multihop wireless sensor networks," *IEEE transactions on parallel and distributed systems*, vol. 22, no. 1, pp. 163–175, 2010.

[30] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *2011 8th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*. IEEE, 2011, pp. 46–54.

[31] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure data aggregation in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1040–1052, 2012.

[32] C. Zhao, W. Zhang, Y. Yang, and S. Yao, "Treelet-based clustered compressive data aggregation for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4257–4267, 2014.

[33] S. Boubiche, D. E. Boubiche, A. Bilami, and H. Toral-Cruz, "Big data challenges and data aggregation strategies in wireless sensor networks," *IEEE Access*, vol. 6, pp. 20 558–20 571, 2018.

[34] J. Chen, Q. Yu, Y. Zhang, H.-H. Chen, and Y. Sun, "Feedback-based clock synchronization in wireless sensor networks: A control theoretic approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2963–2973, 2010.

[35] M. Leng and Y.-C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5404–5414, 2011.

[36] K. S. Yildirim and A. Kantarci, "Time synchronization based on slow-flooding in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 244–253, 2013.

[37] F. Lamonaca, A. Gasparri, E. Garone, and D. Grimaldi, "Clock synchronization in wireless sensor network with selective convergence rate for event driven measurement applications," *IEEE transactions on instrumentation and measurement*, vol. 63, no. 9, pp. 2279–2287, 2014.

[38] J. Wu, L. Zhang, Y. Bai, and Y. Sun, "Cluster-based consensus time synchronization for wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1404–1413, 2014.

[39] C. Wang, K. Sohraby, Y. Hu, B. Li, and W. Tang, "Issues of transport control protocols for wireless sensor networks," in *Proceedings. 2005 International Conference on Communications, Circuits and Systems, 2005.*, vol. 1. IEEE, 2005, pp. 422–426.

[40] M. Zawodniok and S. Jagannathan, "Predictive congestion control protocol for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 3955–3963, 2007.

[41] D. Lee and K. Chung, "Adaptive duty-cycle based congestion control for home automation networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 42–47, 2010.

[42] N. Aslam, K. Xia, A. Ali, and S. Ullah, "Adaptive tcp-iccw congestion control mechanism for qos in renewable wireless sensor networks," *IEEE sensors letters*, vol. 1, no. 6, pp. 1–4, 2017.

[43] J. Tan, W. Liu, T. Wang, S. Zhang, A. Liu, M. Xie, M. Ma, and M. Zhao, "An efficient information maximization based adaptive congestion control scheme in wireless sensor network," *IEEE access*, vol. 7, pp. 64 878–64 896, 2019.

[44] J. Meng, H. Li, and Z. Han, "Sparse event detection in wireless sensor networks using compressive sensing," in *2009 43rd Annual Conference on Information Sciences and Systems*. IEEE, 2009, pp. 181–185.

[45] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, and P. J. Havinga, "Distributed event detection in wireless sensor networks for disaster management," in *2010 international conference on intelligent networking and collaborative systems*. IEEE, 2010, pp. 507–512.

[46] Y. Liu, X. Zhu, C. Ma, and L. Zhang, "Multiple event detection in wireless sensor networks using compressed sensing," in *2011 18th International Conference on Telecommunications*. IEEE, 2011, pp. 27–32.

[47] P. Zhang, I. Nevat, G. W. Peters, G. Xiao, and H.-P. Tan, "Event detection in wireless sensor networks in random spatial sensors deployments," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6122–6135, 2015.

[48] W. Zhu, J. Cao, and M. Raynal, "Energy-efficient composite event detection in wireless sensor networks," *IEEE Communications Letters*, vol. 22, no. 1, pp. 177–180, 2017.

[49] M. A. Weimer, T. S. Paing, and R. A. Zane, "Remote area wind energy harvesting for low-power autonomous sensors," in *2006 37th IEEE Power Electronics Specialists Conference*. IEEE, 2006, pp. 1–5.

[50] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2007, pp. 21–30.

[51] K. V. Naveen and S. Manjunath, "A reliable ultracapacitor based solar energy harvesting system for wireless sensor network enabled intelligent buildings," in *2011 2nd International Conference on Intelligent Agent & Multi-Agent Systems*. IEEE, 2011, pp. 20–25.

[52] C. Moraes and D. Har, "Charging distributed sensor nodes exploiting clustering and energy trading," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 546–555, 2016.

[53] T. Ruan, Z. J. Chew, and M. Zhu, "Energy-aware approaches for energy harvesting powered wireless sensor nodes," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2165–2173, 2017.

[54] C. Wang, J. Li, Y. Yang, and F. Ye, "Combining solar energy harvesting with wireless charging for hybrid wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 560–576, 2017.

[55] A. Boukerch, L. Xu, and K. El-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2413–2427, 2007.

[56] K. Ren, W. Lou, and Y. Zhang, "Leds: Providing location-aware end-to-end data security in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 585–598, 2008.

[57] A. Rasheed and R. N. Mahapatra, "The three-tier security scheme in wireless sensor networks with mobile sinks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 958–965, 2010.

[58] H. Fouchal, J. Biesa, E. Romero, A. Araujo, and O. N. Taladrez, "A security scheme for wireless sensor networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–5.

[59] C. Miranda, G. Kaddoum, E. Bou-Harb, S. Garg, and K. Kaur, "A collaborative security framework for software-defined wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2602–2615, 2020.

[60] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[61] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[62] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[63] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, vol. 49, pp. 1–25, 2019.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[65] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017.

[66] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[67] L. Wang, M. J. Er, and S. Zhang, "A kernel extreme learning machines algorithm for node localization in wireless sensor networks," *IEEE Communications Letters*, 2020.

[68] M. I. AlHajri, N. T. Ali, and R. M. Shubair, "Indoor localization for iot using adaptive feature selection: a cascaded machine learning approach," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 11, pp. 2306–2310, 2019.

[69] X. Wang, X. Liu, Z. Wang, R. Li, and Y. Wu, "Svm+ kf target tracking strategy using the signal strength in wireless sensor networks," *Sensors*, vol. 20, no. 14, p. 3832, 2020.

[70] T. Wang, X. Wang, W. Shi, Z. Zhao, Z. He, and T. Xia, "Target localization and tracking based on improved bayesian enhanced least-squares algorithm in wireless sensor networks," *Computer Networks*, vol. 167, p. 106968, 2020.

[71] K. Zhang, K. Yang, S. Li, D. Jing, and H.-B. Chen, "Ann-based outlier detection for wireless sensor networks in smart buildings," *IEEE Access*, vol. 7, pp. 95 987–95 997, 2019.

[72] S. Duraisamy, G. K. Pugalendhi, and P. Balaji, "Reducing energy consumption of wireless sensor networks using rules and extreme learning machine algorithm," *The Journal of Engineering*, vol. 2019, no. 9, pp. 5443–5448, 2019.

[73] P. Biswas, R. Charitha, S. Gavel, and A. S. Raghuvanshi, "Fault detection using hybrid of kf-elm for wireless sensor networks," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019, pp. 746–750.

[74] S. Radhika and P. Rangarajan, "On improving the lifespan of wireless sensor networks with fuzzy based clustering and machine learning based data reduction," *Applied Soft Computing*, vol. 83, p. 105610, 2019.

[75] L. Cao, Y. Cai, Y. Yue, S. Cai, and B. Hang, "A novel data fusion strategy based on extreme learning machine optimized by bat algorithm for mobile heterogeneous wireless sensor networks," *IEEE Access*, vol. 8, pp. 16 057–16 072, 2020.

[76] J. Wu and G. Li, "Drift calibration using constrained extreme learning machine and kalman filter in clustered wireless sensor networks," *IEEE Access*, vol. 8, pp. 13 078–13 085, 2019.

[77] I. A. Najm, A. K. Hamoud, J. Lloret, and I. Bosch, "Machine learning prediction approach to enhance congestion control in 5g iot environment," *Electronics*, vol. 8, no. 6, p. 607, 2019.

[78] H. S. Z. Kazmi, N. Javaid, M. Imran, and F. Outay, "Congestion control in wireless sensor networks based on support vector machine, grey wolf optimization and differential evolution," in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–8.

[79] J. C. Kwan, J. M. Chaulk, and A. O. Fapojuwo, "A coordinated ambient/dedicated radio frequency energy harvesting scheme using machine learning," *IEEE Sensors Journal*, 2020.

[80] A. Kumar and T. J. Lim, "Edima: Early detection of iot malware network activity using machine learning techniques," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 289–294.

[81] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7702–7712, 2019.

[82] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[83] J. R. Quinlan, "Simplifying decision trees," *International journal of man-machine studies*, vol. 27, no. 3, pp. 221–234, 1987.

[84] ——, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[85] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[86] S. Otoum, B. Kantarci, and H. Mouftah, "Adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures," in *2018 IEEE international conference on communications (ICC)*. IEEE, 2018, pp. 1–6.

[87] F. V. Jensen *et al.*, *An introduction to Bayesian networks*. UCL press London, 1996, vol. 210.

[88] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[89] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2. IEEE, 2004, pp. 985–990.

[90] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[91] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[92] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[93] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[94] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[95] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[96] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 5049–5059.

[97] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *arXiv preprint arXiv:1904.12848*, 2019.

[98] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[99] H. Xu, "Semi-supervised manifold learning based on polynomial mapping for localization in wireless sensor networks," *Signal Processing*, p. 107570, 2020.

[100] F. Hajjej, M. Hamdi, R. Ejbali, and M. Zaied, "A distributed coverage hole recovery approach based on reinforcement learning for wireless sensor networks," *Ad Hoc Networks*, vol. 101, p. 102082, 2020.

[101] I. Mustafa, S. Aslam, M. B. Qureshi, N. Ashraf, S. Aslam, S. M. Mohsin, and H. Mustafa, "Rl-madp: Reinforcement learning-based misdirection attack prevention technique for wsn," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 721–726.

[102] N. Aslam, K. Xia, and M. U. Hadi, "Optimal wireless charging inclusive of intellectual routing based on sarsa learning in renewable wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8340–8351, 2019.

[103] X. Li, X. Hu, R. Zhang, and L. Yang, "Routing protocol design for underwater optical wireless sensor networks: A multi-agent reinforcement learning approach," *IEEE Internet of Things Journal*, 2020.

[104] Z. Jin, Q. Zhao, and Y. Su, "Rcar: A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10 881–10 891, 2019.

[105] S. Najjar-Ghabel, L. Farzinvash, and S. N. Razavi, "Mobile sink-based data gathering in wireless sensor networks with obstacles using artificial intelligence algorithms," *Ad Hoc Networks*, p. 102243, 2020.

[106] F. Viani, P. Rocca, G. Oliveri, D. Trinchero, and A. Massa, "Localization, tracking, and imaging of targets in wireless sensor networks: An invited review," *Radio Science*, vol. 46, no. 05, pp. 1–12, 2011.

[107] R. Elhabyan, W. Shi, and M. St-Hilaire, "Coverage protocols for wireless sensor networks: Review and future directions," *Journal of Communications and Networks*, vol. 21, no. 1, pp. 45–60, 2019.

[108] L. Wang, J. Yan, T. Han, and D. Deng, "On connectivity and energy efficiency for sleeping-schedule-based wireless sensor networks," *Sensors*, vol. 19, no. 9, p. 2126, 2019.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3035846, IEEE Sensors Journal

[109] Y. Yue, L. Cao, and Z. Luo, "Hybrid artificial bee colony algorithm for improving the coverage and connectivity of wireless sensor networks," *Wireless Personal Communications*, vol. 108, no. 3, pp. 1719–1732, 2019.

[110] R. F. Leppänen and T. Hämäläinen, "Network anomaly detection in wireless sensor networks: A review," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, 2019, pp. 196–207.

[111] R. S. Sachan, M. Wazid, D. Singh, A. Katal, and R. Goudar, "Misdirection attack in wsn: Topological analysis and an algorithm for delay and throughput prediction," in *2013 7th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2013, pp. 427–432.

[112] I. Kaushik, N. Sharma, and N. Singh, "Intrusion detection and security system for blackhole attack," in *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*. IEEE, 2019, pp. 320–324.

[113] W. A. Aliady and S. A. Al-Ahmadi, "Energy preserving secure measure against wormhole attack in wireless sensor networks," *IEEE Access*, vol. 7, pp. 84 132–84 141, 2019.

[114] P. C. Kala, A. P. Agrawal, and R. R. Sharma, "A novel approach for isolation of sinkhole attack in wireless sensor networks," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 163–166.

[115] P. Bhatt and A. Morais, "Hads: hybrid anomaly detection system for iot environments," in *2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*. IEEE, 2018, pp. 191–196.

[116] R. Sathiyavathi and B. Bharathi, "A review on fault detection in wireless sensor networks," in *2017 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2017, pp. 1487–1490.

[117] 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.

[118] D. J. Flora, V. Kavitha, and M. Muthuselvi, "A survey on congestion control techniques in wireless sensor networks," in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*. IEEE, 2011, pp. 1146–1149.

[119] V. Pandey, A. Kaur, and N. Chand, "A review on data aggregation techniques in wireless sensor network," *Journal of Electronic and Electrical Engineering*, vol. 1, no. 2, pp. 01–08, 2010.

[120] Y. Gu, F. Ren, Y. Ji, and J. Li, "The evolution of sink mobility management in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2015.

[121] X. Tang, Z. Mones, X. Wang, F. Gu, and A. D. Ball, "A review on energy harvesting supplying wireless sensor nodes for machine condition monitoring," in *2018 24th International Conference on Automation and Computing (ICAC)*. IEEE, 2018, pp. 1–6.

[122] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *2011 Third International Conference on Computational Intelligence, Modelling & Simulation*. IEEE, 2011, pp. 308–311.

[123] T. Zia and A. Zomaya, "A security framework for wireless sensor networks," in *Proceedings of the 2006 IEEE Sensors Applications Symposium, 2006.* IEEE, 2006, pp. 49–53.

[124] K. Thangaramya, K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, and A. Kannan, "Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in iot," *Computer Networks*, vol. 151, pp. 211–223, 2019.

[125] R. Huang, L. Ma, G. Zhai, J. He, X. Chu, and H. Yan, "Resilient routing mechanism for wireless sensor networks with deep learning link reliability prediction," *IEEE Access*, vol. 8, pp. 64 857–64 872, 2020.

[126] J. Cao, X. Zhang, C. Zhang, and J. Feng, "Improved convolutional neural network combined with rough set theory for data aggregation algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 647–654, 2020.

[127] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, and D. J. Inman, "Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks," *Journal of Sound and Vibration*, vol. 424, pp. 158–172, 2018.

[128] H. Cheng, Z. Xie, Y. Shi, and N. Xiong, "Multi-step data prediction in wireless sensor networks based on one-dimensional cnn and bidirectional lstm," *IEEE Access*, vol. 7, pp. 117 883–117 896, 2019.

[129] H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, "Data prediction model in wireless sensor networks based on bidirectional lstm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 203, 2019.

[130] Y.-F. Zhang, P. J. Thorburn, W. Xiang, and P. Fitch, "Ssim—a deep learning approach for recovering missing time series sensor data," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6618–6628, 2019.

[131] S. N. Mohanty, E. L. Lydia, M. Elhoseny, M. M. G. Al Otaibi, and K. Shankar, "Deep learning with lstm based distributed data mining model for energy efficient wireless sensor networks," *Physical Communication*, p. 101097, 2020.

[132] Z. E. Khatab, A. Hajihoseini, and S. A. Ghorashi, "A fingerprint method for indoor localization using autoencoder based deep extreme learning machine," *IEEE sensors letters*, vol. 2, no. 1, pp. 1–4, 2017.

[133] X. Luo, L. Liu, J. Shu, and M. Al-Kali, "Link quality estimation method for wireless sensor networks based on stacked autoencoder," *IEEE Access*, vol. 7, pp. 21 572–21 583, 2019.

[134] G. Li, S. Peng, C. Wang, J. Niu, and Y. Yuan, "An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 86–96, 2018.

[135] R. A. Alshinina and K. M. Elleithy, "A highly accurate deep learning based approach for developing wireless sensor network middleware," *IEEE Access*, vol. 6, pp. 29 885–29 898, 2018.

[136] Y. Su, X. Lu, Y. Zhao, L. Huang, and X. Du, "Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9561–9569, 2019.

[137] Y. Su, R. Fan, X. Fu, and Z. Jin, "Dqelr: An adaptive deep q-network-based energy-and latency-aware routing protocol design for underwater acoustic sensor networks," *IEEE Access*, vol. 7, pp. 9091–9104, 2019.

[138] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2018.

[139] S. Redhu, P. Garg, and R. Hegde, "Joint mobile sink scheduling and data aggregation in asynchronous wireless sensor networks using q-learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6438–6442.

[140] S. Redhu and R. M. Hegde, "Cooperative network model for joint mobile sink scheduling and dynamic buffer management using q-learning," *IEEE Transactions on Network and Service Management*, 2020.

[141] J. Li, Z. Xing, W. Zhang, Y. Lin, and F. Shu, "Vehicle tracking in wireless sensor networks via deep reinforcement learning," *IEEE Sensors Letters*, vol. 4, no. 3, pp. 1–4, 2020.

[142] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (ddpg)-based energy harvesting wireless communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8577–8588, 2019.

[143] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron," *IEICE Technical Report, A*, vol. 62, no. 10, pp. 658–665, 1979.

[144] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[145] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on wireless communications*, vol. 1, no. 4, pp. 660–670, 2002.

[146] R. Mhemed, N. Aslam, W. Phillips, and F. Comeau, "An energy efficient fuzzy logic cluster formation protocol in wireless sensor networks," *Procedia Computer Science*, vol. 10, pp. 255–262, 2012.

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

[148] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant nocs," in *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2009, pp. 21–26.

[149] K. Fan, J. Lu, D. Sun, Y. Jin, R. Shen, and B. Sheng, "Failure resilient routing via iot networks," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 845–850.

[150] P. M. Mohan, M. Gurusamy, and T. J. Lim, "Dynamic attack-resilient routing in software defined networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1146–1160, 2018.

[151] L.-Y. Sun, W. Cai, and X.-X. Huang, "Data aggregation scheme using neural networks in wireless sensor networks," in *2010 2nd International*

Conference on Future Computer and Communication, vol. 1. IEEE, 2010, pp. V1–725.

[152] Y. YANG and S. LIU, "Data aggregation in wsn based on sofm neural network," *Chinese Journal of Sensors and Actuators*, vol. 26, pp. 1757–1760, 2013.

[153] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[154] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[155] J. W. Graham, "Missing data analysis: Making it work in the real world," *Annual review of psychology*, vol. 60, pp. 549–576, 2009.

[156] N. Jaques, S. Taylor, A. Sano, and R. Picard, "Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2017, pp. 202–208.

[157] M. Das and S. K. Ghosh, "A deep-learning-based forecasting ensemble to predict missing data for remote sensing analysis," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 12, pp. 5228–5236, 2017.

[158] Y. Song, J. Luo, C. Liu, and W. He, "Periodicity-and-linear-based data suppression mechanism for wsn," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, pp. 1267–1271.

[159] A. Sinha and D. Lobiyal, "Prediction models for energy efficient data aggregation in wireless sensor network," *Wireless Personal Communications*, vol. 84, no. 2, pp. 1325–1343, 2015.

[160] Q. Liu, D. Jin, J. Shen, Z. Fu, and N. Linge, "A wsn-based prediction model of microclimate in a greenhouse using extreme learning approaches," in *2016 18th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2016, pp. 730–735.

[161] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[162] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 37–49.

[163] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "Selm: Semi-supervised elm with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566–2572, 2011.

[164] X. Lu, H. Zou, H. Zhou, L. Xie, and G.-B. Huang, "Robust extreme learning machine with its application to indoor positioning," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 194–205, 2015.

[165] Y. Gu, Y. Chen, J. Liu, and X. Jiang, "Semi-supervised deep extreme learning machine for wi-fi based localization," *Neurocomputing*, vol. 166, pp. 282–293, 2015.

[166] T. Jayasri and M. Hemalatha, "Link quality estimation for adaptive data streaming in wsn," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1543–1562, 2017.

[167] M. Deb, S. Roy, B. Saha, P. Das, and M. Das, "Designing a new link quality estimator for sensor nodes by combining available estimators," in *2017 IEEE 7th International Advance Computing Conference (IACC)*. IEEE, 2017, pp. 179–183.

[168] S. Rekik, N. Baccour, M. Jmaiel, and K. Drira, "Holistic link quality estimation-based routing metric for rpl networks in smart grids," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016, pp. 1–6.

[169] W. Sun, X. Yuan, J. Wang, Q. Li, L. Chen, and D. Mu, "End-to-end data delivery reliability model for estimating and optimizing the link quality of industrial wsns," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1127–1137, 2017.

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

[171] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 603–611.

[172] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[173] L. Capra, "Malm: Machine learning middleware to tackle ontology heterogeneity," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*. IEEE, 2007, pp. 449–454.

[174] K. Lingaraj, R. V. Biradar, and V. Patil, "Eagilla: An enhanced mobile agent middleware for wireless sensor networks," *Alexandria engineering journal*, vol. 57, no. 3, pp. 1197–1204, 2018.

[175] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[176] L. F. Vecchietti, T. Kim, K. Choi, J. Hong, and D. Har, "Batch prioritization in multigoal reinforcement learning," *IEEE Access*, vol. 8, pp. 137 449–137 461, 2020.

[177] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, no. 8, p. 716, 1952.

[178] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1223–1229, 2009.

[179] M. A. Jadoon and S. Kim, "Relay selection algorithm for wireless cooperative networks: a learning-based approach," *IET Communications*, vol. 11, no. 7, pp. 1061–1066, 2017.

[180] X. Liang, M. Chen, I. Balasingham, and V. C. Leung, "Cooperative communications with relay selection for wireless networks: design issues and applications," *Wireless Communications and Mobile Computing*, vol. 13, no. 8, pp. 745–759, 2013.

[181] P. Xie, J.-H. Cui, and L. Lao, "Vbf: vector-based forwarding protocol for underwater sensor networks," in *International conference on research in networking*. Springer, 2006, pp. 1216–1221.

[182] M. Al-Bzoor, Y. Zhu, J. Liu, A. Reda, J.-H. Cui, and S. Rajasekaran, "Adaptive power controlled routing for underwater sensor networks," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2012, pp. 549–560.

[183] R. Banerjee and C. K. Bhattacharyya, "Cluster based routing algorithm with evenly load distribution for large scale networks," in *2014 International Conference on Computer Communication and Informatics*. IEEE, 2014, pp. 1–6.

[184] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, 2018.

[185] U. Challita, L. Dong, and W. Saad, "Proactive resource management for lte in unlicensed spectrum: A deep learning perspective," *IEEE transactions on wireless communications*, vol. 17, no. 7, pp. 4674–4689, 2018.

[186] J.-S. Leu, T.-H. Chiang, M.-C. Yu, and K.-W. Su, "Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes," *IEEE communications letters*, vol. 19, no. 2, pp. 259–262, 2014.

[187] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proceedings of the 38th annual Hawaii international conference on system sciences*. IEEE, 2005, pp. 287a–287a.

[188] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "Pw-mac: An energy-efficient predictive-wakeup mac protocol for wireless sensor networks," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011, pp. 1305–1313.

[189] Z. Wang, V. Aggarwal, and X. Wang, "Iterative dynamic water-filling for fading multiple-access channels with energy harvesting," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 382–395, 2015.

[190] M.-L. Ku, W. Li, Y. Chen, and K. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2641–2657, 2015.

[191] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Lyapunov optimization for energy harvesting wireless sensor communications," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1947–1956, 2018.