Fast Convergence and Energy Saving through Machine Learning in Wireless Sensor Networks

Prof. Raxit Jani[a], Dhruv Kansara[b], Priyal Munshi[c], Vaishvi Parikh[d*]

[a]*Assistant Professor(CE), Gandhinagar Institute of Technology, Gandhinagar 382721, Gujarat, India*
[bcd*]*Student(CE), Gandhinagar Institute of Technology, Gandhinagar 382721, Gujarat, India*

Abstract

Environments that are constantly changing are monitored via wireless sensor networks. In connection to the pertinent problem, the benefits and drawbacks of each suggested algorithm are assessed. We also provide a comparison reference to assist WSN designers in developing successful machine learning solutions for their specific application problems. In this study, we provide an energy-saving framework for Wireless Sensor Networks (WSN) that is based on external factors, machine learning methods, and meta-heuristics. As opposed to conventional topology-based energy-saving methods, we focus on the sensor node's energy savings inside the WSN itself. We attempt energy conservation on the sensor nodes twice. The first step in achieving network level energy savings, sometimes referred to as N1-energy savings, is identifying the absolute minimum number of sensor nodes necessary to maintain the WSN's operation. In order to determine the least amount of sensor nodes, we utilise hybrid filter wrapper feature selection, a well-known machine learning approach, to find the optimal feature subsets. Second, by altering the sampling rate and transmission interval of the sensor nodes, we use a method known as N2-energy saving to reduce the energy consumption of the WSNs. In order to do this, we suggest an optimisation technique based on Simulated Annealing (SA), a proven technique that may find the approximative global optimum in datasets where it is challenging to gather precise values due to noise issues, such as sensor data.

Keywords: Event detection, localization, clustering, data aggregation, query processing, wireless sensor networks, machine learning, reinforcement learning, data mining, and compressive sensing.

1. Introduction

Computer architectures are educated in the region of computer science and artificial intelligence known as machine learning so that they automatically improve over time. A sort of artificial intelligence called system studying, or one of its variations, enables computers to get better at a particular task without having been formally instructed on it. In order to forecast or make choices about new data, machine learning algorithms utilise statistical techniques to identify relationships and patterns in data. Systems may learn algorithms via supervised learning, unsupervised learning, and reinforcement learning, among other techniques. A machine learning model is trained on a categorised dataset in supervised learning where each information point is connected to a recognised outcome or label. The model then develops the ability to predict the results for brand-new data points. Unsupervised learning is training a machine learning model on a dataset without labels without knowing the results beforehand, with the goal of having the model detect patterns and correlations in the statistics. In reinforcement learning, a machine learning model is trained to make decisions in a dynamic environment where it receives feedback in the form of rewards or penalties entirely based on its actions. There are several applications for machine learning, including recommendation systems, trickery, computer vision, and natural language processing.

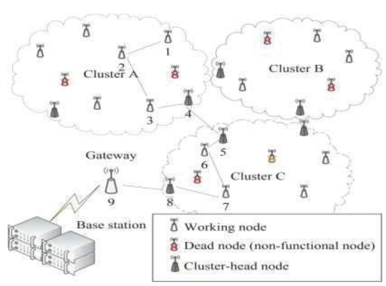

Fig. 1. Data aggregation example with working, dead, and cluster heads nodes in a clustered architecture. [1]

*Vaishvi Parikh
E-mail address: vaishviparikh02@gmail.com

Wireless sensor networks (WSNs) can take advantage of several kinds of machine learning (ML) procedures to increase energy efficiency. Wireless sensor networks (WSNs) can benefit from a diversity of machine learning (ML) technologies to increase energy efficiency. Examples include: Predictive maintenance: ML algorithms can be used to forecast when a node or group of nodes in a WSN will need to be maintained or replaced. By preventing the needless replacement of nodes, this strategy can help lower energy consumption. Anomaly detection: ML algorithms can be secondhand to spot problems with the WSN, including broken nodes, and send out alarms to notify the device administrator. By promptly detecting and resolving issues that could result in increased energy intake, this method can help reduce energy usage. Load balancing: ML methods can be used to evenly distribute the burden among the WSN nodes. The workload might be gently distributed among nodes to improve performance and prevent excessive power usage. Facts compression: By using ML algorithms to compress node statistics within the WSN, less data must be transferred, which saves energy. For instance, using auto encoders can significantly shrink the quantity of the data while keeping the necessary records. Power management in WSNs can be improved with the use of ML algorithms. Reinforcement learning, for instance, can be used to research the best power consumption methods for each node based entirely on network status, node attributes, and ambient conditions. Typically, machine learning can significantly enhance WSN energy performance, resulting in a longer network lifetime, lower energy consumption, and improved network performance.

2. Machine Learning

Small, low-power sensors arranged in networks called wireless sensor networks (WSNs) connect wirelessly with one another to gather data and send it to a gateway or central node. WSNs can utilise machine learning approaches to increase the effectiveness and precision of data collecting and processing. Some uses of machine learning in wireless sensor networks include the following: Analytics of data: Machine learning techniques can be expended to examine the information gathered by the network's sensors. For instance, classification algorithms can be used to find patterns and anomalies in the data, while clustering algorithms can be used to group similar data points together. Energy efficiency: Machine learning can be used to optimise how much energy is utilised by the network's sensors. For instance, decision tree algorithms can be used to forecast which sensors should be turned on or off at any given time, while reinforcement learning can be used to determine the best energy consumption policy for each sensor in the network. Localization: WSNs' sensor localization accuracy can be increased by using machine learning methods. Examples of techniques that can be used to increase the precision of localization based on signal intensity and time-of-flight data include support vector machine (SVM) and neural network algorithms. Security breaches in WSNs can be found and avoided using machine learning techniques.For example, decision tree algorithms can be used to identify potential security concerns and take necessary action, while anomaly detection algorithms can be used to spot odd activity in the network.

3. Supervised Learning

In supervised learning, a labelled training set is used to build the system model (i.e., present inputs and known outputs). This model illustrates the learned connection between the input, output, and system parameters. In this subcategory, the primary supervised learning techniques are discussed with relation to WSNs. In fact, supervised learning algorithms are frequently employed in WSNs to solve a variety of issues, such as localization and object pointing, event discovery and query handling, media access control, security and intrusion discovery, quality of service (QoS), data veracity, and fault discovery.

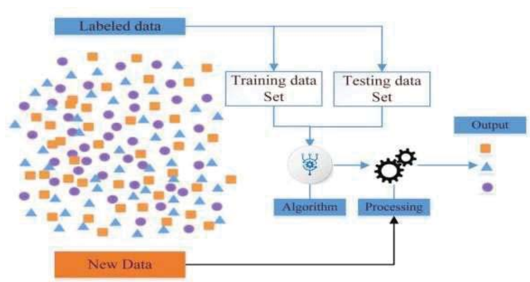

Fig. 2. Supervised Learning. [19]

1) K-nearest neighbor (k-NN): This supervised learning method shapes a data models, known as a query point, using the tags (i.e., the output values) of the neighbouring data models. For instance, lost interpretations of a sensor node may be anticipated using the average readings of surrounding sensors that fall within precise diameter constraints. Many different functions may be used to locate the cluster of nodes that is nearest. One simple procedure is to use the Euclidean distance between several sensors. Knearest Neighbor uses less computer power because the function calculates in respect to neighbouring places (i.e., k-nearest points, where k is a small positive integer). Due to this and the linked readings of neighbouring nodes, the distributed learning method knearest neighbour is a suitable fit for WSNs. The k-NN approach may yield incorrect fallouts when analysing problems in high-dimensional spaces (more than 10-15 dimensions) because the distance between different data samples becomes constant (i.e., the distances to the nearest and farthest neighbours are marginally comparable). The most common use of the k-nearest neighbour approach in WSNs is the query processing subsystem.

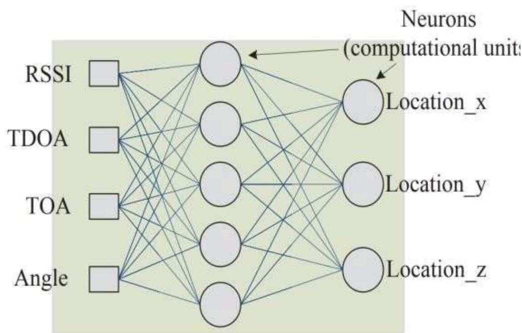

Fig. 3. Using supervised neural networks, node localisation in 3D space in WSNs is demonstrated. [1]

2) Decision tree (DT): It is a classification approach that anticipates input data labels using an iterative learning tree. In order to arrive at a certain category, the feature attributes are compared in relation to the decision conditions. The texts is incredibly rich in results that use the DT algorithm to address various architecture issues with WSNs. For instance, by detecting a few crucial parameters including loss rate, average time to failure (MTTF), and average time to restoration, DT offers a straightforward but effective technique to determine link dependability in WSNs . Nevertheless, DT is limited to data that can be separated linearly, and creating the best discovering trees is NP-complete.

3) Neural networks (NNs): A cascade chain of decision units, such as radial basis functions, which are employed to recognise complex and nonlinear functions, could be employed to build this learning process. Due to the significant computational costs associated with learning the network weights and the considerable administrative overhead, the use of distributed neural networks in WSNs is still relatively uncommon. The ability of neural networks to simultaneously learn many outputs and decision boundaries in centralised solutions, however, makes them suited for employing the same model to solve a variety of network difficulties. We use the issue of localising sensor nodes—that is, figuring out the location of each node—as an instance for neural networks in WSNs. The propagation distance and angle dimensions of the signals received from anchor nodes can be used for node localisation. Such measurements can comprise the received signal strength indicator (RSSI), time of arrival (TOA), and time difference of arrival (TDOA). After supervised training, a neural network produces vector-valued coordinates in three-dimensional space representing an estimated node location. Algorithms related to neural networks include learning vector quantization (LVQ). One of the crucial uses of neural networks, in addition to perform estimation, is the tuning and dimensionality reduction of big data (high-dimensional and complicated data sets).

4) Support vector machines (SVMs): This technique classifies data points as it learns using labelled training examples. One technique for determining whether a node is engaging in malicious activities is to use SVM to examine the temporal and geographical correlations of data. Take SVM, for instance, which, given WSN data as instants in the feature space, splits the space into sections. Each reading will be categorised according to which side of the separation gaps it lands on, as shown in Fig., where these components are separated by as wide of margins as is practical. The SVM approach includes optimising a quadratic function with linear constraints as an substitute to the multi-layer neural network with non-convex and unhindered optimisation problem (that is, the challenge of creating a set of hyperplanes). Two potential applications for SVM in WSNs are security and localization.

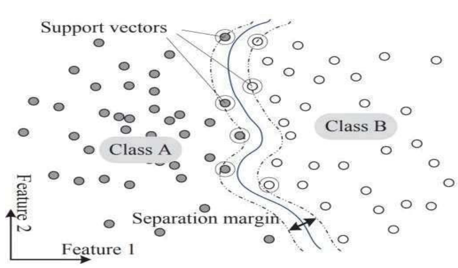

Fig. 4. A nonlinear support vector machine illustration. [1]

5) Bayesian statistics: Contrary to the majority of machine learning techniques, Bayesian inference only needs a modest number of training data. Without over-fitting, Bayesian algorithms efficiently adapt probability distribution to learn ambiguous ideas (like). The key is to transform previous ideas into new beliefs using current information (for instance, acquired data denoted by the letter D). $p(\theta|D) \square\, p(\theta)p(D|\theta)$, where $p(\theta|D)$ is the posterior probability of the parameter θ given the observation D, and $p(D|\theta)$ is the probability of the observation D given the parameter θ. One usage of Bayesian inference in WSNs is the assessment of event consistency with partial data sets (θ) by using previous environmental information. Nevertheless, the extensive application of Bayesian algorithms in WSNs is constrained by the requirement for statistical expertise. An associated statistical learning approach is Gaussian Process Regression (GPR) mode.

4. Unsupervised Learning

Unsupervised students are not given labels (i.e., there is no output vector). An unsupervised learning algorithm's main objective is to divide the sample set into various groups by examining their similarities. As might be expected, node clustering and data aggregation problems frequently exploit this topic of learning techniques. This widespread adoption is, in fact, a result of data architectures (i.e., the lack of labelled data) and the intended outcome in such cases.
1) K-means clustering: To classify data into different groups, the k-means algorithm is employed (known as clusters). The linear complexity and straightforward implementation of this unsupervised learning method make it a popular choice for solving the sensor node clustering problem.
2) Principal component analysis (PCA It is a multivariate technique for data density and dimensionality decreases that seeks to identify key information in data and display it as a collection of new principal components, which are orthogonal variables. The principle components are arranged so that the first component, and subsequent components, correlate to the direction of the data with the highest variance. So, as they have the least informational content, the least-variance components can be ignored. For instance, To decrease the amount of data sent between sensor nodes, PCA identifies a limited group of uncorrelated linear groupings of the initial readings. Moreover, the PCA approach simplifies the resolution of problems with a high number of variables by considering just a small number of conditions.

There are several machine learning techniques that can be used in wireless sensor networks, including:

5. Clustering: In the data from the sensor network, clustering algorithms can be used to group together comparable data points. By doing so, it may be possible to find patterns and trends in the data that can be used to make predictions about the future or spot abnormalities.

5.1 Introduction to Clustering
In essence, it is a kind of unsupervised learning technique. The process of drawing references from datasets of input data without labelled replies is known as unsupervised learning. It is typically used as a method to identify the groups, generative qualities, and significant structures that are inherent in a set of instances. In order to make the data points within each group more similar to one another and distinct from the data points within the other groups, clustering divides the populace or collection of data points into a number of collections. In essence, it is a classification of items according on how similar and unlike they are to one another.

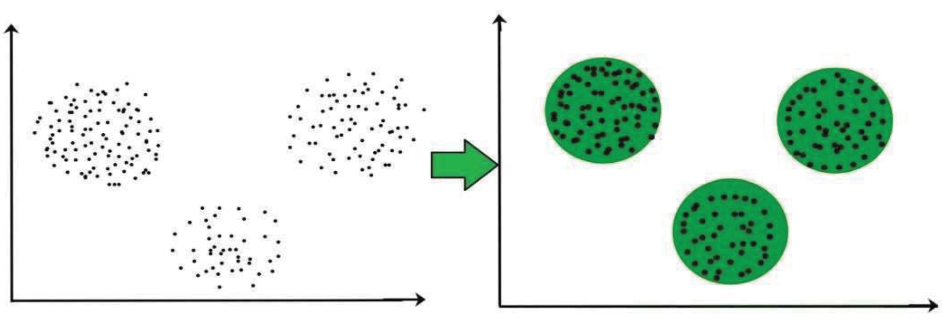

Fig. 5. Clustering.

5.2 Clustering Methods
Clusters, according to density-based methods, are a dense portion of the space that varies from and resembles the lower dense region. These techniques can merge two clusters and are accurate. Examples include OPTICS (Ordering Points to Detect Clustering Structure), DBSCAN (Density-Based Spatial Clustering of Applications with Noise), etc. Techniques that are based on hierarchy: Based on the hierarchy, these clusters build a structure that resembles a tree. The already- formed cluster is utilised to create new ones. There are two classifications. Agglomerative (bottom-up method) and Dividing (top- down strategy) Examples include stable iterative reducing clustering and utilising hierarchies (BIRCH), clustering using

representatives (CURE), etc. The objects are divided into k clusters using these partitioning methods; each split creates a new group. This technique is used to improve an objective criteria similarity function, such K-means or CLARANS, where distance is a key parameter (Clustering Large Applications based upon Randomized Search).
Grid-based Methods: In this approach, the data space is divided into a limited number of grid-like cells. These grids, including STING (Statistical Information Grid), wave cluster, CLIQUE (Clustering In Query), etc., can quickly and regardless of the quantity of data items do all clustering operations.

5.3 Clustering Algorithm

The models of the clustering algorithms, which were previously described, can be used to categorise them. Many clustering techniques have been described, however only a few of them are frequently utilised. The type of data we use determines the clustering algorithm. For instance, some algorithms must estimate the number of clusters in the supplied dataset, while others must determine the shortest distance between the dataset's observations. Here we are examining mainly popular Clustering algorithms that are broadly used in machine learning:

1. **K-Means algorithm:** One of the most broadly used clustering techniques is k-means. By grouping the samples into various clusters with similar variances, it classifies the dataset. With this approach, the amount of clusters must be given. It is rapid, requires less computations, and has linear complexity of **O(n)**.
2. **Mean-shift algorithm:** The mean-shift technique seeks out solid regions within a even distribution of data points. It serves as an illustration of a centroid-based model that updates the potential centroid candidates to serve as the geographic centre of the points inside a particular region.
3. **DBSCAN Algorithm:** Density-Based Spatial Clustering of Applications with Noise is the abbreviation for the DBSCAN algorithm. It serves as an illustration of a density-based model that is comparable to the mean-shift but has several notable rewards. The algorithm divides the low density zones into the high density zones. The clusters can therefore be found in any random shape.
4. **Expectation-Maximization Clustering using GMM:** The k-means algorithm can be replaced with this approach, or it can be employed in situations where K-means can fail. The data points in GMM are thought to have a Gaussian distribution.
5. **Agglomerative Hierarchical algorithm:** The bottom-up hierarchical clustering is carried out by the Agglomerative hierarchical method. In this, each data point is initially handled as a single cluster and then subsequently merged. A tree-structure can be used to illustrate the cluster hierarchy.
6. **Affinity Propagation:** It differs from earlier clustering methods in that it does not call for a certain number of clusters to be specified. Each data point communicates with the other until the pair of data points converge. The fundamental flaw with this algorithm is that it takes a lot of time O(N2T).

Table. 1. Compassion of different machine learning-based data aggregation and node clustering mechanisms.

Mechanisms	Machine learning algorithm(s)	Complexity	Balancing energy con - sumption	Delay	Overhead	Topolog y aware
Large scale network clustering	NNs	Moderate	Yes	High	Low	Yes
Cluster head election	DT	Low	Yes	Low	Low	Yes
Gaussian process models for censored sensor readings	GPR	Moderate	No	Moderate	Moderate	No
Adaptive sampling		High	Yes	High	High	No
Clustering using SOM and sink distance	SOM	Moderate	No	High	Moderate	Yes
Online data compression	LVQ	High	No	High	High	Yes
Data acquisition using compressive sensing	PCA	High	Yes	High	High	Yes
Transmission reduction		Moderate	No	High	High	Yes
Consensus-based distributed PCA		Moderate	Yes	High	High	No
Lossy data compression		Moderate	No	Moderate	High	Yes
Collaborative signal processing	k-means	Low	Yes	Moderate	Moderate	No
Advanced surveillance systems		Moderate	Yes	Low	Low	Yes
Role-free clustering	Q-learning	Low	No	Low	Low	No
Decentralized learning for data latency	RL	Moderate	Yes	Low	Low	No

6. Reinforcement learning

Algorithms for reinforcement learning can be used to enhance how the network's sensors behave. For each sensor in the network, the ideal power consumption policy might be learned using, for instance, a reinforcement learning algorithm. With the use of reinforcement learning, an mediator (such a sensor node) can learn by interacting with its surroundings. Using its own expertise, the agent will discover the optimum course of action to maximise its long-term benefits. Q-learning is the most popular reinforcement learning method.

An mediator periodically updates its attained prizes depending on the actions conducted at a certain stage, as depicted in Fig. Eq is used to calculate the future total reward (also known as the Q-value) for carrying out an action in a certain state st.

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

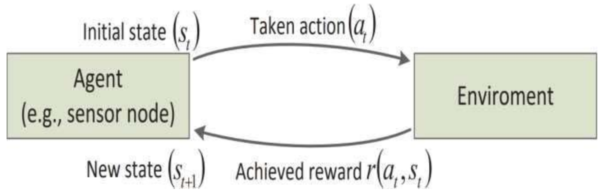

Fig. 6. A Q-learning approach visualisation. [1]

Where γ is the studying rate that defines how quickly learning takes place and r(st, at) is the instant prize of carrying out an action at a certain stage st (typically set to value between 0 and 1). With a distributed design like a WSN, where every node aims to select actions that are anticipated to maximise its longstanding benefits, this method may be simply implemented. It is important to highlight that the WSN routing issue has successfully and widely exploited Q-learning.

Aggregating data and clustering Transmission of all data openly to the sink is wasteful in large-scale sensor networks with limited energy. Sending the data to a local collector, often referred to as a cluster head, which aggregates data from all the sensors in its cluster and delivers it to the sink, is one effective option. Energy will often be saved as a result of this. The best choice for the cluster head has been covered in a number of works. It compares conventional clustering methods and provides a classification for them. Data collection in WSNs on a cluster-based basis from suppliers to an access point. It could be required to take any troublesome nodes out of the network in this situation. Such flawed nodes could provide false readings, which might reduce the accuracy of the network as a whole. The following are the major ways that node clustering and data aggregation are improved by ML techniques: Using machine learning to effectively extract similarity and dissimilarity (for example, from malfunctioning nodes) in readings from various sensors, cluster heads can compress data locally. In order to efficiently choose the cluster head, machine learning methods are used. The network's lifespan will be increased and energy consumption will be considerably reduced with wise cluster head selection. In the event where the method allocates computation, demanding jobs among all nodes while taking the remaining energy information into account, it is indicated in the column "Balancing energy usage". The column "Topology aware" denotes the need for in-depth network topology knowledge.

The winning neuron j* is identified as the one whose weight vector w(t) is closest to the input vector x(t):

$$j^* = \arg\min_j \left\| x_j(t) - w_j(t) \right\|, j = 1, ..., N$$

where N stands for the second layer's total number of neurons. Moreover, the neighbouring nodes of the winner node are changed as follows:

$$w_j\left(t+1\right) = w_j\left(t\right) + h\left(t\right)\left(x_j\left(t\right) - w_j\left(t\right)\right)$$

where w (t) and w (t + 1) are, respectively, a neuron's values at time t and t+1. The Gaussian neighbourhood function, denoted by the notation:

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

By using CODA for data aggregation, data quality will be improved, network energy will be conserved, and network traffic will be decreased. 5) Using vector quantization to compress data online: While the methods mentioned above necessitate complete network topology knowledge, some algorithms might not. A method called "Adaptive Learning Vector Quantization" (ALVQ), for instance, to properly recover condensed variants of data from the sensor nodes, was created by Lin et al. The LVQ learning approach is used by ALVQ to predict the code-book by using training samples from the past and utilising data correlation and historical patterns. The ALVQ method uses less bandwidth during transmission and increases the accuracy of original reading recovery from the lossless compression. The major drawback of utilising LVQ for online data aggregation is that dead neurons located far from the training samples would never compete. So, it is crucial to create algorithms that are resistant to outliers. In the same vein, LVQ is useful for encoding large data sets with a small number of vectors. Principal component analysis (PCA) is a powerful tool that may be effectively combined with two other key methods to improve data aggregation in wireless sensor networks (WSNs). Compressive sensing (CS) has lately been investigated as a potential alternative to the conventional "sample then compress" method. In order to reconstruct the original signal from a few random measurements, CS investigates the sparsity property of signals. An iterative technique called Expectation-Maximization (EM) consists of two steps: an expectation (E) step and a maximisation (M) phase. In its E-step, EM fixes the present expectation of the system parameters while formulating the cost function. The M step then recalculates the parameters to minimise the cost function estimation error. A technique for estimating dispersed observations using sparsely collected samples from a WSN was developed by Masiero et al. The PCA method, which is the basis for this solution, creates orthogonal components that compressive sensing uses to rebuild the original signals. Furthermore, because this method can estimate data spatial and temporal correlations, it is not dependent on the routing protocol. Similar to this, Rooshenas et alPCA .'s method was used to enhance readings' direct transmission to a base station. PCA significantly reduces traffic by merging the data acquired by nodes into fewer packets. This distributed method combines all incoming packets rather than sending them on to their destinations and is carried out in intermediate nodes. Another significant development was the introduction of distributed consensus-based data compression techniques by Macua et al. employing PCA and maximum likelihood of the observed data. Some techniques include "Consensus-based EM Distributed PCA" and "Consensus-based Distributed PCA," which both rely on probing the eigenvectors of local covariance matrices (CB-EM DPCA). A distributed EM algorithm is used in the latter.

7. Neural networks

In wireless sensor networks, neural networks, a form of machine learning technique, can be utilised for a range of tasks including classification, regression, and prediction. For tasks involving intricate, non-linear interactions between variables, neural networks can be especially helpful.Wireless sensor networks (WSN) can employ neural networks, a form of machine learning technique, for a number of tasks including classification, regression, and prediction. For tasks involving intricate, non-linear interactions between variables, neural networks are very helpful. A neural network is made up of several interconnected layers of nodes, often known as neurons, where each node takes input from the layer above and outputs to the layer below. Up until the final output is formed, the input to the network is fed forward through the layers, each layer changing the input into a more abstract representation.

Neural network architectures with applications in WSN include, for instance: Multi-Layer Perceptron (MLP): An example of a neural network with many layers of completely linked nodes is the MLP. MLP is applicable to projects like classification and regression. CNNs (Convolutional Neural Networks): CNNs are a particular kind of neural network that are frequently employed for image classification jobs. For tasks like image identification and sensor data processing, CNN can also be used in WSN. Recurrent Neural Networks (RNN) are a particular class of neural network that are made to handle sequential input, such time series data. With sensor information, RNN can be utilised for tasks like forecasting environmental factors. Long Short-Term Memory (LSTM): This RNN type was created specifically to deal with long-term dependencies in sequential data. LSTM can be used to do tasks like forecasting a WSN's energy usage. Neural networks can be computationally costly and need a lot of training data to learn complicated correlations between variables. They can, however, be extremely accurate and adaptable for a variety of WSN applications.

8. Anomaly detection

Algorithms for anomaly detection can be used to spot data points or occurrences that deviate from typical patterns of behaviour. This might be helpful for spotting equipment problems or security risks in the sensor network. Several machine learning approaches can be used to detect anomalies in wireless sensor networks (WSNs). In a WSN, anomaly detection is normally carried out by examining the data gathered from various sensors in the network to spot outliers or odd behaviour that significantly deviates from expected trends.

Typical machine learning methods for WSN anomaly detection include:

Unsupervised Learning: Unsupervised learning techniques are frequently employed for anomaly detection in WSNs because they don't require labelled data. Data points can be grouped together depending on how similar they are using clustering techniques like k-means and hierarchical clustering, and outliers that do not fit into any cluster can be found.

Supervised Learning: To train the model in supervised learning methods, labelled data is necessary. To categorise new data points as normal or anomalous in WSN, a supervised learning system can be trained on data with previously observed normal and abnormal behaviour. For the purpose of detecting anomalies, decision trees, support vector machines (SVM), and neural networks are often used supervised learning methods. Deep learning is a kind of machine learning that employs neural networks to discover intricate data patterns. By first training a deep neural network on typical sensor data and then detecting deviations from the typical pattern, deep learning can be used for anomaly detection in WSN.

Ensemble Learning: To enhance the overall performance of the anomaly detection system, ensemble learning combines many machine learning models. By mixing various anomaly detection models, ensemble learning can be employed in WSN to improve the system's accuracy and robustness. Anomaly detection is a crucial WSN duty since it can be used to spot problems like equipment malfunctions or environmental events that may need more research. Anomaly detection in WSN can be greatly aided by machine learning techniques, however the technique to use relies on the system at hand as well as the features of the data.

9. Uses for ML in Securing WSN Networks

In this part, we go through how ML algorithms may be used to improve WSN security. We examine the role of ML algorithms in meeting these criteria based on what was formerly said about the protection needs in WSNs. The majority of computer vision (ML) security applications have been employed in intrusion detection technologies to comprehend packet flow in the network. Reducing DDoS and DoS assaults is one way that these ML algorithms contribute to network availability. Others, like ransomware assaults, assist in analysing the behaviour of infections and lowering the dangers to data integrity. Moreover, several ML techniques aid in reducing the risk of authentication attacks between WSN nodes. The next sections will provide a detailed presentation of each of these subsections.

Availability One of the key prerequisites for network security is availability. Consequently, numerous deliberate or accidental assaults such as DoS, equipment damage, or power reach the core of WSN devices under the guise of availability. Network availability may be improved using techniques like congestion management, intrusion detection, and error detection, for instance.

Intrusion Detection The main duties of an intrusion detection system are, in general, to monitor hosts and networks, assess network activity, generate alarms, and respond to suspicious activity. Since they keep an eye on connected hosts and connections, intrusion detection systems are frequently installed next to secured network devices (e.g., the switches). Since each WSN node functions independently as a host and a network device (a router and switch) in WSNs, each node must carry out the identical intrusion detection procedure. The two types of detection are signature-based and anomaly-based, with anomalies being preferred in terms of teaching skills to WSN nodes. The ML training process, which we discuss in the subsection on WSN challenges, is still a concern. As a result, numerous studies in this section have tried to enhance the wireless sensor network's machine learning training process by speeding up training, using a smaller data set, and increasing accuracy. A unique model was developed by authors in order to improve DoS identification and decrease power consumption in WSNs. The authors also proposed a novel cluster architecture for the LEACH protocol to distribute forwarding messages among WSN nodes. After that, they used feature selection and a classifier approach to improve DDoS Attack identification. Feature selection is a different technique for minimising the amount of features in a dataset; it entails selecting the traits that will be used for training and excluding the rest. The authors' calculations of the recommended method's power use on WSN also revealed a 5% increase in power usage. The authors claim that one of the best machine learning techniques for protecting wireless sensor networks against DoS is the decision tree, which provides a 100% correct response. The authors of this study also looked at how different ML algorithms affected the ability of WSNs to identify DoS. They chose ML methods of various types (statistical, logical, instance, and deep learning) and applied them to various dataset sizes in order to evaluate the effect of storage capacity on the training process in ML algorithms. They also researched the WSN nodes' lightweight ML algorithms. According to the findings, datasets of between 3,000 and 6,000 records work best—as long as the proportion of attacked to un-attacked records is 1:1. The outcomes also demonstrated that the G-boost classifiers, which are part of the logic-based (decision tree) category, are the best classifiers. The top DoS detection system also resulted in a 32% increase in network power consumption. Also, in the same context as assessing deep learning and conventional machine learning algorithms on wireless sensor network traffic packets. Simple models like (LR, DT, and SVM) have been demonstrated by the authors in to be excellent for the practical application of intrusion detection from deep learning approaches. The use of statistical analysis has been recommended as a different method for online DoS detection. The authors applied binary logistic regression to the forward-selective and black-hole attacks. Once a run-time monitor tool was used to aggregate the local WSN node activity, evaluating whether the packets were malicious or benign, binary logistic regression was utilised to estimate the detection accuracy. The method's output (logistic regression) was then applied to the WSN network to monitor node behaviour in terms of threat detection. The accuracy of their recommendations varied from 96 to 100%. A alternative rule-based ML approach was recommended in. The authors created hybrid approaches that combine fuzzy logic and other methodologies with a rule-based approach to deal with ambiguities, errors, and vagueness. Next, the reliability of these aspects was evaluated. In, the authors suggested a brand-new model to boost intrusion detection effectiveness while also extending network lifetime.

The authors suggested an adaptive chicken swarm optimization methodology to reduce a WSN node's power consumption, and they employed two tiers of the SVM method for intrusion detection. The SVM will be used to inspect packets at the second level after being used to identify the spiteful node at the first level. Although the paper addressed the issue of extending WSN lifetime, the findings do not provide any justifications for the amount of energy that the suggested solution has saved. The authors of also created a versatile intrusion detection technique using a deep neural network (DNN). The outcomes also shown an improvement in the results' accuracy for various kinds of network traffic. The proposed method's performance accuracy was also described in the study, however it omitted to indicate how much electricity and CPU it would need. The authors developed a lightweight intrusion detection approach for WSN networks by combining particle swarm optimisation (PSO) and a backpropagation neural network (BNN). The authors provided a hybrid feature selection technique together with a two-level classifier in order to improve the performance of intrusion prevention accuracy (rotation forest and bagging).

SVM and MLP were additionally utilised to categorise traffic data and spot rogue nodes in the WSN network. Other writers have created a hybrid classifier between synthetic groups of machine learning algorithms, going in a different path. The authors of the study recommended a hybrid classifier that blends deep learning and traditional machine learning techniques. The idea merged the

LTSM model and the Gaussian Bayes model to improve intrusion detection in WSNs. The suggestion in contrast combined the MLP model with a genetic algorithm (GA). However all of the proposed intrusion detection algorithms that were previously described use a lot of electricity. As a result, in some other trials, the training process was moved from the WSN node to the console using Software Defined Network (SDN) technology. As a result, these suggestions are quite effective at lowering the effort on WSN nodes. To transfer the training results to the sensors promptly, these methods must change a number of protocols that take place between switches, controllers, and WSN nodes. Authors of distributed machine learning approaches for intrusion detection train terminal nodes to avoid any effects of detection processes using a layered approach between controller and switch. The first stage of the controller's training involved the usage of a decision tree, KNN, NB, and LR, while the second stage involved the implementation of the switches. Nevertheless, the research avoided going into great detail about the improvements and modifications it made to the SDN protocols to back up this claim. Also, in order to create an advanced intelligence framework, the authors combined KNN with the arithmetic optimization method (AOA) in evolutionary computation. Also, the authors improved the detection of phishing assaults utilising SDN in the same scenario.

The features extraction procedure, which was based on the URL and subject of websites, was combined with traditional approaches (blacklist and whitelist) to create the optimization. Based on the results of the features extraction of packets that originate from users, the blacklist and whitelist are updated. The feature extraction procedure made use of the naive Bayes classifier. The controller then makes adjustments to the flow rule table before sending it to switches so they can take appropriate action for each packet that complies with the rules. The previous process will be repeated if the packet does not match any value in the rule action table. The suggested solution is substantial and difficult, notwithstanding the advancements shown by their results. To recognise URL packets that are neither blacklisted or whitelisted, the authors in used a method of machine learning based on stacking. Also, the authors in employed CNN and SDN together to increase the accuracy of URL detection. To categorise a URL in a signature-based database to various phishing assaults, the CNN is employed in the controller. This classification determines whether the incoming packet inspection is delivered directly to the destination or enters slow mode. It will conduct additional inspections in the slow mode in order to update the signature-based database. Despite the economic viability of feature selection in reducing training costs and enhancing performance, none of the three took it into account in their plans.

Error Detection Error-detecting algorithms used in machine learning are excellent examples. WSNs are also prone to mistakes and malfunctions because of their various software and hardware concerns, as well as the fact that they are used in many different domains. To quickly find faults in a WSN, extensive application detection methods are needed due to all of these challenges. The authors employed a decision fusion technique with a trust mechanism. The KNN, extreme learning machine, SVM, and recurrent learning machine classification algorithms are four that are offered in order to increase the efficacy of the belief function fusion approach. This approach, however, does not take the dynamics of various WSN node failures into consideration. A hidden Markov model was utilised by the authors to identify the dynamics of transitions brought on by errors, and neural networks were used to categorise faults based on the state transition probability produced by the Markov model. This made it possible to capture the WSN nodes in real time when a malfunction occurred. By fusing a hidden Markov model with several neural networks, such as learning vector quantization, probabilistic neural network, probabilistic adaptive neural network, and radial basis function, the authors emphasised on error detection and classification. Although the authors in employed SVM classification, the authors in used the SVM regression model for error detection in WSNs using conventional ML techniques. The authors of further suggested a multi-class SVDD classifier with recursive PCA as a real-time live error detection approach. Using the rapid recursive principal component analysis method, the error in WSNs was discovered. The WSN node error rate is used by this module to identify the problematic nodes.

Congestion Control Congestion management may be seen as one of the duties that helps to assure network availability, even if some people consider it to be a part of the quality of service. Additionally, machine learning techniques are quite helpful in this subject. When a node or communication channel receives more data than it can handle, congestion develops in WSNs. Congestion is brought on by a number of factors, including buffer node bypass, transmission channel contention, packet collision, dynamic time shift, and transmission rate. End-to-end latency, energy utilisation, and packet loss are all impacted by congestion. ML algorithms may help with congestion management issues by figuring out the optimum path and predicting network traffic. The authors estimated congestion and estimated the probability of packet loss using the Random Early Detection (RED) active queue management approach. This protocol modifies each WSN node's data transmission and lowers the buffering queue using percentage integration differentiation theory and fuzzy logic. Transmission rate change, congestion reporting, and congestion detection are the three steps of this system. Congestion is originally identified using RED and fuzzy proportional integral derivative (FuzzyPID) controller techniques. Implicit congestion reporting is established when congestion is found. Finally, a fuzzy controller is used to change the transmission rate in order to reduce congestion. The authors also used a buffer occupancy-based active queue management approach to identify congestion. It determines the amount of packet loss based on the size of the current queue and modifies the queue length appropriately. They offer the first implementation of WSN node queue management using the relative integration differentiation control theory. The relative integration differentiation controller's percentage, integral, and differential parameters are then changed using an online weighting scheme produced by the neurons' capacity for self-learning and self-regulation.

Authentication A group of security measures known as authentication make assurance that data has originated from the source and has not been tampered with along the route. Because of its strategy, active attacks like DoS and spoofing are mitigated. Both the network component and message features are included in authentication. Entity authentication is accomplished because both the claimant and the verifier engage and communicate without providing any crucial information besides the claim to be a certain entity. Message authentication would guarantee appropriateness even though it cannot confirm when a message was generated. In conventional networks, authentication is carried out using typical public-key cryptography algorithms and techniques like RSA, ECC, Defihelman, and others. The implementation of such processes, however, results in power exhaustion because of the features of wireless sensor networks that were previously addressed. It can be employed in various works in addition to contemporary techniques of identification based on motion sensors for users (devices or humans), but it also necessitates a powerful CPU and battery capacity. As a result, the physical layer authentication method is an excellent choice for wireless sensor network setups.

Figure illustrates how 41% of the studies that were analysed used ML algorithms for intrusion detection. In terms of error detection, 18% comes in second. The remaining 14% congregate for further research. This is because ML algorithms are expensive to implement on hardware (devices) and require training procedures. Additionally, the majority of it was used to preserve network availability, and its application to both confidentiality and integrity was challenging. The application of ML algorithms in the security of WSNs of three categories (confidentiality, integrity, and availability) is covered in the following sections, along with certain unresolved concerns that still require in-depth investigation. We also offer potential solutions to these problems.

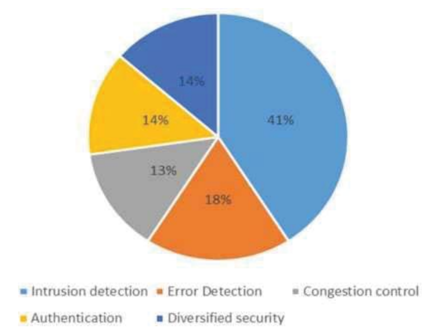

Fig. 7. Statastical analysis for the reviewed ML algoriyhms in WSN [19]

WSN-IoT Research Problems That Can Be Solved Using Machine Learning Techniques The WSN-IoT research issues that ML techniques can answer are given below. IoT node localization In a WSN scenario, node localization refers to determining a sensor node's precise position at any given time. For mobile WSN nodes, the route planning procedure is essential. Node localization is seen as a classification issue since all nodes are divided (classified) into range-based and range-free nodes. For the classification issue of node localization, WSN-IoT employs a variety of ML approaches, including SVM, K-NN, and RL-based techniques (Q-learning, SARSA). IoT node coverage and connection The sensing coverage (FOI), where at least one sensor node includes each location, is the area of interest in a WSN scenario. Hence, deciding where to put sensor nodes is a design problem. To increase the WSN lifespan, the connection between the neighbour nodes must be at its best. Routing Layer Issues Data packets are sent from one node to another via intermediate nodes using a process called routing. Throughout the routing process, the gateway nodes maintain comprehensive routing tables that include the source and destination addresses of each packet in the network. The main gateway node receives the sensed data from the end nodes of a WSN. If the routing route is excessively lengthy in a WSN network, unnecessary energy is wasted. Intelligent routing algorithms must be appropriately designed in order to identify the optimum routes between end nodes and gateway nodes. Several machine learning techniques, such as decision trees, random forests, ANNs, SVMs, and Bayesian learning, are used to find the optimum route in WSNs. A MAC Layer The MAC layer controls how the media is accessed in a WSN. The sensor MAC (SMAC) protocol is often used in WSN. WSN employs approaches based on reinforcement learning for MAC protocol design (RL). RL-MAC algorithms regulate the sleep, wake, transmission, and reception in sensor networks.

10. Experiments and Performance Evaluation

We design a number of tests using the two CASAS smart hone sensor datasets gathered from the WSN single resident apartments to demonstrate the advantages of the proposed framework. These sensor datasets include sensor data from a variety of in-home devices as well as activity labels for things like sleeping, dressing, using a phone, grooming, and other things. All datasets have the same types of sensor nodes, and Table 2 gives a description of the sensors and sensor data types. At a minimum, the smart homes have spaces for sleeping, dining, cooking, and living. The WSN of the datasets is varied in terms of the number of sensor nodes, deployment, internal structure of the home, and other factors even when the types of sensor nodes are the same. Table 3 explains the WSN's statistical data. We conduct the experiments to demonstrate the efficacy of the suggested filter-wrapper method and the optimal sample rate and transmission interval method using the CASAS datasets.

THE FILTER-WRAPPER SENSOR SELECTION'S PERFORMANCE IN A SMART HOME

For the purpose of detecting and sensing activities at home, CASAS sensor nodes are deployed. Even in a room, there are numerous sensor nodes, which results in extra or redundant sensors being used to detect activity. In order to compare the predictive performance of the chosen sensor subset and all sensor sets, we carry out the suggested filter-wrapper sensor selection. Also, we compare the quantity of evaluations of wrappers performed using a batching mechanism and all conceivable subsets as the evaluation target set. It will demonstrate the batch method's quick performance and simplicity of computation. Figure displays the prediction performance for all sizes of the chosen sensor subsets.

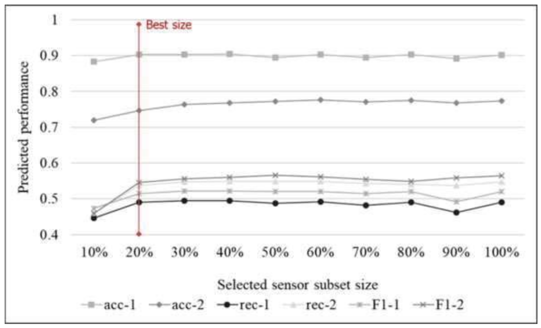

Fig. 8. Predictive performance of the size of selected sensor subset. [25]

From 20% on, the majority of the measures do not cause substantial performance disparities, according to the experiment's findings. In other words, even when just utilising 20% of the original WSN's sensor subset, the predictive performance is comparable. The majority of measurements are poor when the sensor subset is 90% in size. The cause is that unsuitable sensor nodes may produce noise that interferes with the classification of patterns. As a result, improving the predictive performance by adding more sensors is not always a good idea. Based on these findings, our sensor selection approach can reduce N2-energy consumption while keeping the original WSN's adequate predictive performance with a much smaller number of sensors. In addition, the ideal sensor setup and sensor count to maximise Also, it is possible to determine the ideal sensor configuration and quantity for maximising the WSN's predictive performance.

THE RESULTS OF OPTIMAL SAMPLING RATE AND TRANSMISSION INTERVAL BY ENERGY-AWARE AND MACHINE LEARNING-BASED SA

Using modified SA, the suggested framework detects and establishes the ideal sampling rate and transmission interval. The area to be searched for by the SA is described using potential sample rates and transmission intervals with the prior best 20% of the sensor subsets, and the location of the optimum we identified is analysed to demonstrate the N1-energy savings performance of the suggested method. Figure displays the results, with each area representing the datasets 1, 2, and 3. Instead of using the sample rate, we used 1/sampling rate to equilibrate the x- and y-axis measures. As seen in Figure, the loss rises with greater sampling rates and falls with longer transmission intervals. This is an inevitable outcome. With the tiny sensor data set, a faster sampling rate results in a bigger loss. Similar to this, large transmission intervals lengthen the time it takes for sensor data to be delivered, which causes loss because it makes it harder to identify trends. The accuracy cannot be improved by a sensor node with a too-low sample rate and short transmission interval since this loss quickly stabilises. Instead, it leads to needless energy usage. Finding the best sample rate and transmission intervals in this trade-off relationship is crucial since it can significantly minimise the energy used by the sensor node. Looking at the outcome of our method, we can identify a sample rate and transmission that come very near to being ideal. Although the transmission rate and sampling rate's default parameters were 0.05 seconds, our technique proposed values that were more than 10 times larger. Moreover, the WSNs perform similarly and with the highest degree of accuracy. This indicates that our approach can cut the energy use by more than 90%. The sample rate and transmission could not, however, be clearly balanced. For the sample rate, the solution from dataset 1 produced superior results, while the solution from dataset 2 produced better results for the transmission intervals. Also, because we chose recall rather than acc to discover a solution based on the stability of the sensor data, we did not set the sampling rate and transmission interval with as much precision. Because to this, our technique successfully solved both datasets, saving the sensor nodes from wasting energy.

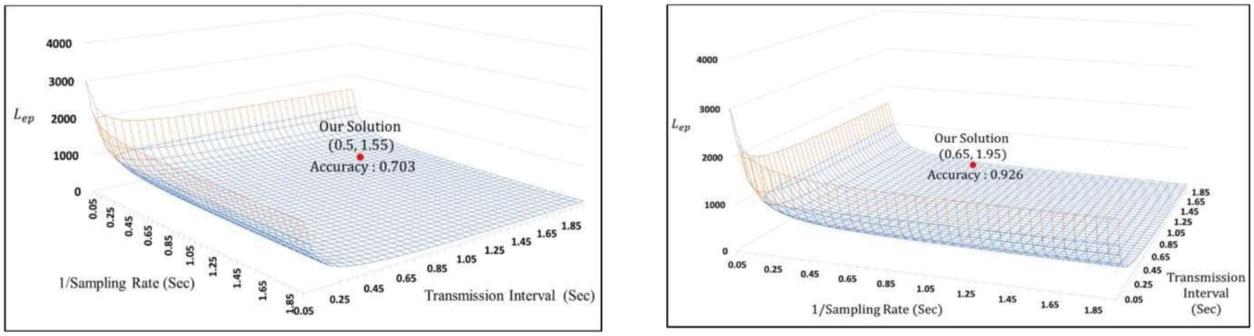

Fig. 9. The search space and our solution use SA in accordance with sample rates and transmission intervals. [25]

11. Conclusion

This study offered a thorough analysis of the 2002–2013 research literature on these subjects. In conclusion, it is critical to consider the network's limited resources as well as the range of learning themes and patterns that would be effective for the specific circumstance when applying machine learning algorithms in wireless sensor networks. The development of distributed and lightweight message-passing techniques, online learning algorithms, hierarchical clustering patterns, and the application of machine learning to the problem of resource management in wireless sensor networks are just a few of the many questions that still need to be investigated further. We proposed energy reductions for the WSN framework using machine learning techniques and meta heuristics while taking environmental variables into account. We have conducted a number of tests to show the superiority of the proposed framework. Instead of using typical energy-saving techniques that alter the design of the WSN, we have achieved energy savings by directly reducing the sensor nodes or by altering their sample rate and transmission interval in the WSN. In order to offer a novel method to energy saving, we also applied machine learning techniques to the meta-heuristics utilising the information from the sensor data. Using studies using CASAS datasets in real-world sensor data, we validated the potential that the modified WSN generated with our technology may have good QoS on sensor data with low energy usage. The suggested method has the following limitations despite offering greater performance. Because we omitted to consider the topology of the WSN, our method might not work in practice owing to subpar routing protocols and complicated topology.

References

1. Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato and Hwee-Pink Tan, Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications.
2. T. O. Ayodele, Introduction to machine learning, in New Advances in Machine Learning. InTech, 2010.
3. A. H. Duffy, 1997. The “what” and “how” of learning in design,” IEEE Expert, vol. 12, no. 3, p. 71–76.
4. P. Langley and H. A. Simon, 1995. Applications of machine learning and rule induction, Communications of the ACM, vol. 38, no. 11, p. 54–64.
5. L. Paradis and Q. Han, 2007. A survey of fault management in wireless sensor networks,” Journal of Network and Systems Management, vol. 15, no. 2, p. 171–190.
6. B. Krishnamachari, D. Estrin, and S. Wicker, 2002. The impact of data aggregation in wireless sensor networks,” in 22nd International Conference on Distributed Computing Systems Workshops, p. 575-578.
7. Zaki Ahmad Khan and Abdus Samad, A Study of Machine Learning in Wireless Sensor Network.
8. P. Langley and H. A. Simon, 1995, Applications of machine learning and rule induction,” Communications of the ACM, vol. 38, no. 11, p. 54–64.
9. L. Paradis and Q. Han, 2007. A survey of fault management in wireless sensor networks,” Journal of Network and Systems Management, vol. 15, no. 2, p.171–190.
10. B. Krishnamachari, D. Estrin, and S. Wicker, 2002. The impact of data aggregation in wireless sensor networks,” in 22nd International Conference on Distributed Computing Systems Workshops, p. 575–578.
11. J. Al-Karaki and A. Kamal, 2004. Routing techniques in wireless sensor networks: A survey, IEEE Wireless Communications, vol. 11, no. 6, p. 6–28.
12. K. Romer and F. Mattern, 2004. The design space of wireless sensor networks,” IEEE Wireless Communications, vol. 11, no. 6, p. 54–61.
13. Himanshu Sharma, Ahteshamul Haque, Frede Blaabjerg, Machine Learning in Wireless Sensor Networks for Smart Cities: A Survey.
14. Alsheikh, M.A.; Lin, S.; Niyato, D.; Tan, H.-P., 2014. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. IEEE Commun. Surv. Tutor., 16, p. 1996–2018. [CrossRef]

15. Du, R.; Santi, P.; Xiao, M.; Vasilakos, A.V.; Fischione, C. 2019. The sensable city: A survey on the deployment and management for smart city monitoring. IEEE Commun. Surv. Tutor., 21, p. 1533–1560. [CrossRef]
16. Morello, R.; Mukhopadhyay, S.C.; Liu, Z.; Slomovitz, D.; Samantaray, 2017, S.R. Advances on sensing technologies for smart cities and power grids: A review. IEEE Sens. J., 17, p. 7596–7610. [CrossRef]
17. Anagnostopoulos, T.; Zaslavsky, A.; Kolomvatsos, K., 2017. Challenges and opportunities of waste management in IoT-enabled smart cities: A survey. IEEE Trans. Sustain. Comput., 2, p. 275–289. [CrossRef]
18. Horng, G.-J.; Liu, M.-X.; Chen, C.-C. 2020. The smart image recognition mechanism for crop harvesting system in intelligent agriculture. IEEE Sens. J., 20, p. 2766–2781. [CrossRef]
19. Rami Ahmad, Raniyah Wazirali, Tarik Abu-Ain, Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues.
20. Wazirali, R.; Ahmad, R.; Al-Amayreh, A.; Al-Madi, M.; Khalifeh, A., 2021, Secure Watermarking Schemes and Their Approachesin the IoT Technology: An Overview. Electronics, 10, 1744. [CrossRef]
21. Bouaziz, M.; Rachedi, A., 2016. A survey on mobility management protocols in Wireless Sensor Networks based on 6LoWPAN technology. Comput. Commun., 74, p. 3–15. [CrossRef]
22. Al-Kashoash, H.A.A.; Kharrufa, H.; Al-Nidawi, Y.; Kemp, A.H., 2019. Congestion control in wireless sensor and 6LoWPAN networks: Toward the Internet of Things. Wirel. Networks, 25, p. 4493–4522. [CrossRef]
23. Moridi, M.A.; Kawamura, Y.; Sharifzadeh, M.; Chanda, E.K.; Wagner, M.; Okawa, H. 2018. Performance analysis of ZigBee network topologies for underground space monitoring and communication systems. Tunn. Undergr. Sp. Technol., 71, p. 201–209. [CrossRef]
24. Ertürk, M.A.; Aydın, M.A.; Büyükakka şlar, M.T.; Evirgen, H., 2019. A Survey on LoRaWAN Architecture, Protocol and Technologies. Future. Internet, 11, 216. [CrossRef]
25. Jaewoong Kang , Jongmo Kim, Minhwan Kim, And Mye Sohn , (Member, IEEE), Machine Learning-Based Energy-Saving Framework for Environmental States-Adaptive Wireless Sensor Network
26. C. Zhu, V. C. M. Leung, L. Shu, and E. C.-H. Ngai, 2015. Green Internet of Things for smart world, IEEE Access, vol. 3, p. 2151–2162, doi: 10.1109/ACCESS.2015.2497312.
27. Y.-H. Lin, Z.-T. Chou, C.-W. Yu, and R.-H. Jan, 2015. Optimal and maximized configurable power saving protocols for corona based wireless sensor networks, IEEE Trans. Mobile Comput., vol. 14, no. 12, p. 2544–2559.
28. T. Amgoth and P. K. Jana, 2015. Energy-aware routing algorithm for wireless sensor networks,'' Comput. Electr. Eng., vol. 41, p. 357–367.
29. S. Shamshirband, J. H. Joloudari, M. Ghasemigol, H. Saadatfar, A. Mosavi, and N. Nabipour, 2019. FCS-MBFLEACH: Designing an energy-aware fault detection system for mobile wireless sensor networks, Mathematics, vol. 8, no. 1, p. 28.
30. F. Deniz, H. Bagci, I. Korpeoglu, and A. Yazıcı, 2016. An adaptive, energy- aware and distributed fault-tolerant topology-control algorithm for heterogeneous wireless sensor networks,'' Ad Hoc Netw., vol. 44, p. 104–117.