

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,500

Open access books available

176,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Machine Learning Algorithms from Wireless Sensor Network's Perspective

Rakesh Chandra Gangwar and Roohi Singh

Abstract

In the last few decades, wireless sensor network (WSN) emerged as an important network technology for real-time applications considering its size, cost-effectiveness and easily deployable ability. Under numerous situations, WSN may change dynamically, and therefore, it requires a depreciating dispensable redesign of the network. Machine learning (ML) algorithms can manage the dynamic nature of WSNs better than traditionally programmed WSNs. ML is the process of self-learning from the experiences and acts without human intervention or re-program. The current Chapter will cover various ML Algorithms for WSN and their pros and cons. The reasons for the selection of particular ML techniques to address an issue in WSNs, and also discuss several open issues related to 'ML for WSN'.

Keywords: wireless sensor network, machine learning, supervised learning, unsupervised learning, reinforcement learning

1. Introduction

1.1 Introduction

A wireless sensor network (WSN) is a network that entails distributed tiny sensor nodes which might be supposed to screen bodily or environmental conditions and communicate with every different and alternate facts and information. Due to their small size, sensor nodes have limited computational power and energy resources. Additionally, the environment wherein they are placed varies dramatically over time. It is essentially critical to analyse sensor data as soon as it is collected. Sensor node data that has not been processed for a long duration is assumed as incomplete and inaccurate. Since WSNs are usually dynamic in nature, their topologies will frequently change. As a result of the connection loss, the network needs to add a new node. The future scope of WSN technology is bright across a wide range of application areas. In this Chapter; we list a few of the most useful ones and also how the different machine learning (ML) techniques are used in deploying the various sensor networks. However there are various other issues when it comes to these networks. The ML algorithms have been proven excellent in resolving issues particularly functional or operational

issues, for example: clustering, processing of query, aggregation of data, localization, etc. While some algorithms focus on non-functional and non-operational issues like quality and efficiency of sensors, quality-of-service (QoS), security and integrity of data, etc. There are also several practical explanations that maximise resource utilisation and extend the life of the network. This chapter is prearranged as follows: Section 1 covers all the basics of WSNs along with applications issues and need of ML techniques. Section 2 presents taxonomy of machine learning algorithms and their details. Finally Section 3 concludes the chapter.

1.2 Overview

A sensor is a very small gadget that is used to capture data about a physical process or phenomenon and convert it into electrical signals that can be processed, monitored, and analysed further to get the purposeful information. Any type of information from the real environment, including temperature, pressure, light, sound, motion, position, flow, humidity, and radiation, could be referred to as a physical process.

In order to record, observe, and respond to an event or a phenomenon, a structure made up of sensors, processing units, and communication components are known as a sensor network. The controlling or observing body may be a consumer application, a government agency, a civil organisation, a military force, or an industrial entity, and the event may be connected to anything, including the physical world, an industrial environment, a biological system, or an IT (information technology) framework. Such sensor networks can be used for data collecting, surveillance, monitoring, medical telemetry, and remote sensing, etc. Sensors with a controller (base station), and a communication system make up a typical sensor network [1]. **Figure 1** illustrates what is known as a Wireless Sensor Network (WSN) when the connectivity in a sensor network is established utilising a wireless protocol.

1.3 Elements of WSN

There are two basic elements of WSN:

- a. **Sensor node:** A Wireless Sensor Network (WSN) [2] consists of sensor nodes that are deployed in close proximity and frequently on massive scales, and it

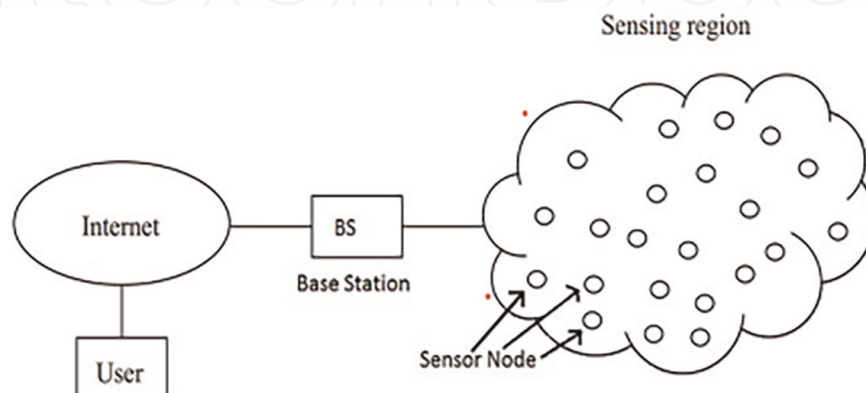


Figure 1.
A basic working model of wireless sensor network.

supports sensing, statistical processing, embedded computing and connectivity. WSNs are innately aid constraints and are chargeable for self-organising the suitable community infrastructure frequently with multi-hop communication with them. It consists of four basic components:

- Power supply;
- Sensor;
- Processing unit; and
- Communication system.

The sensor node collects the analogue statistics from the consumer, and analogue-to-digital convertor (ADC) converts the records into the digital shape. The processing unit is the primary unit, it consists of a storage unit and a microcontroller/microprocessor, which primarily do the information processing and manipulation. It also consists of various other features like network analysis, data correlation and fusion of data from another sensor with its own. The communication system consists of any kind of system that is typically a short-range radio for data transmission and reception (Figure 2) [1].

- Network architecture:** When a massive number of sensor nodes are deployed in a large region to cooperatively screen the physical surroundings, the network of these sensor nodes is equally important. A sensor node in a WSN no longer only communicates with other sensor nodes, however, also with a Base Station

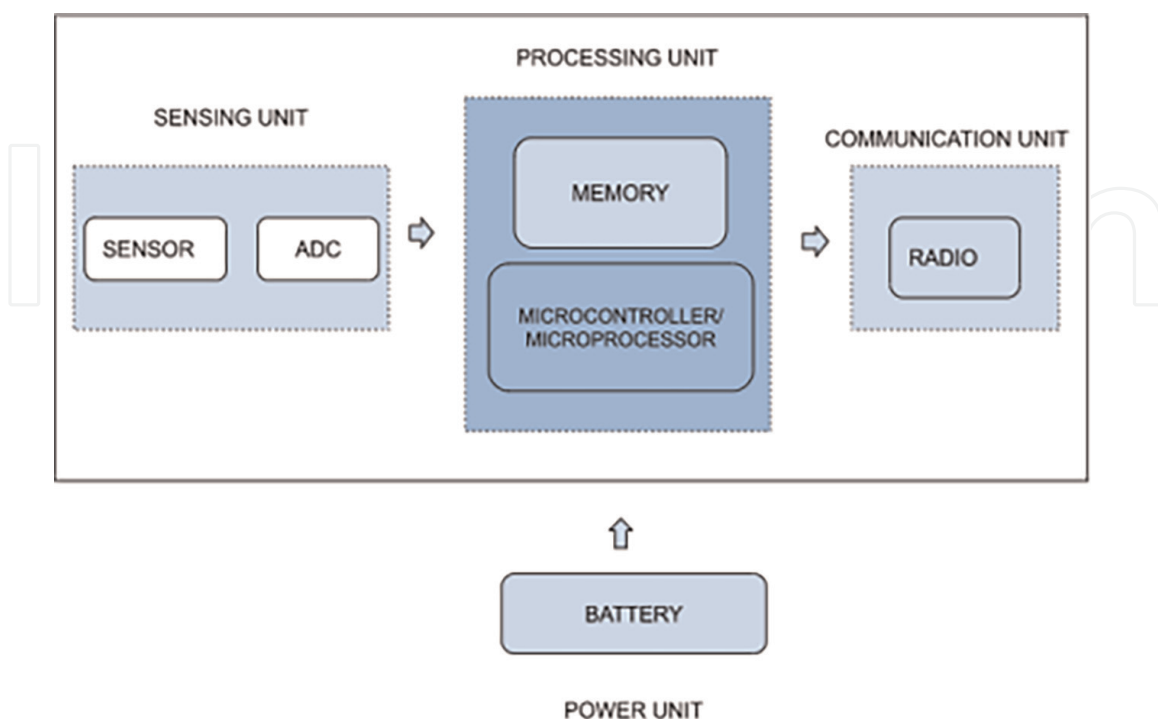


Figure 2.
Structure of a sensor node.

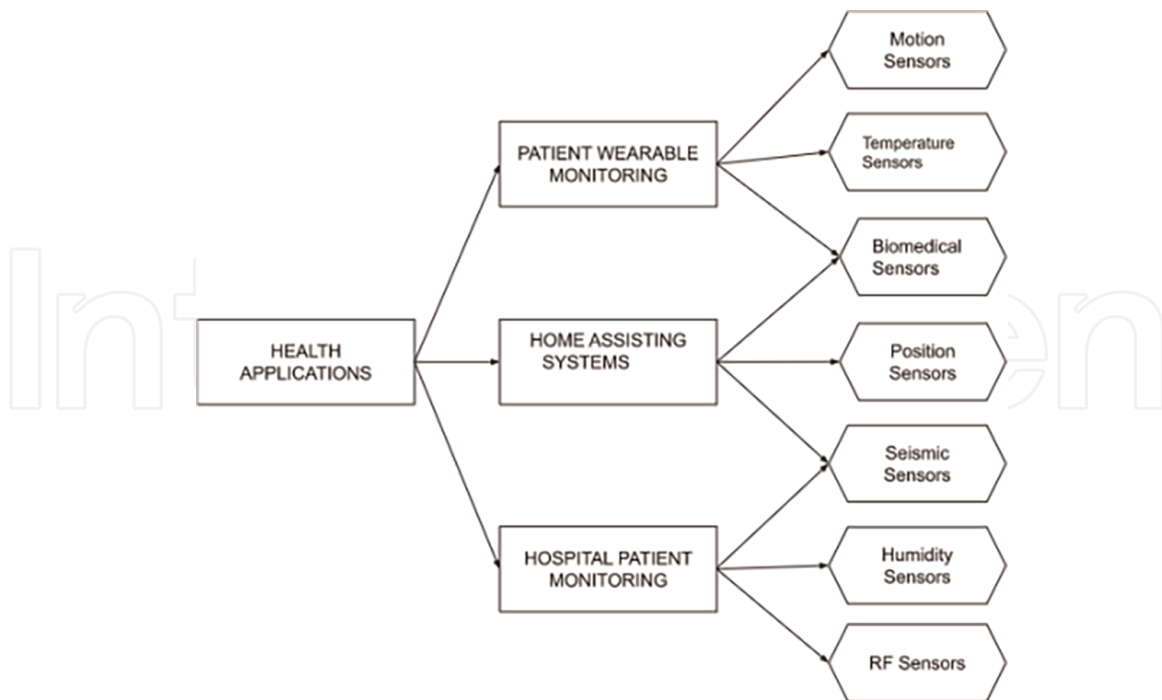


Figure 3.
Subcategories of Military applications of WSN.

(BS) using wireless communication. The base station sends instructions to the sensor nodes, and the sensor nodes perform the task by collaborating with each other. After gathering all the necessary information, the sensor nodes send the data back to the base station. After receiving the records from the sensor nodes, a base station performs processing of records and sends the updated data to the person using internet.

1.4 Applications of WSN

Numerous applications [3] of WSNs are currently either in use or in the process of their development. A few applications of WSNs are listed below:

- 1. Military applications:** The military area is not always the simplest the first field of human pastime that used WSNs, however, it is also considered to have the initiation of sensor network research. Smart dust is an example of those initial studies, efforts which were carried out within the late 19 if you want to broaden sensor nodes which notwithstanding their very small size might be capable of engaging in spying sports. The main subcategories of the military applications [4] of WSNs are battlefield surveillance [5], combat monitoring and intruder detection as shown in **Figure 3**.
- 2. Health applications:** In the health realm, WSNs make use of superior medical sensors to display the real-time tracking of patient's vitals [6]. **Figure 4** shows the primary sub-categories of the health package of WSN.
- 3. Environmental applications:** Environmental programs that seek for the monitoring of the ambient situations at adverse and far flung areas may advance

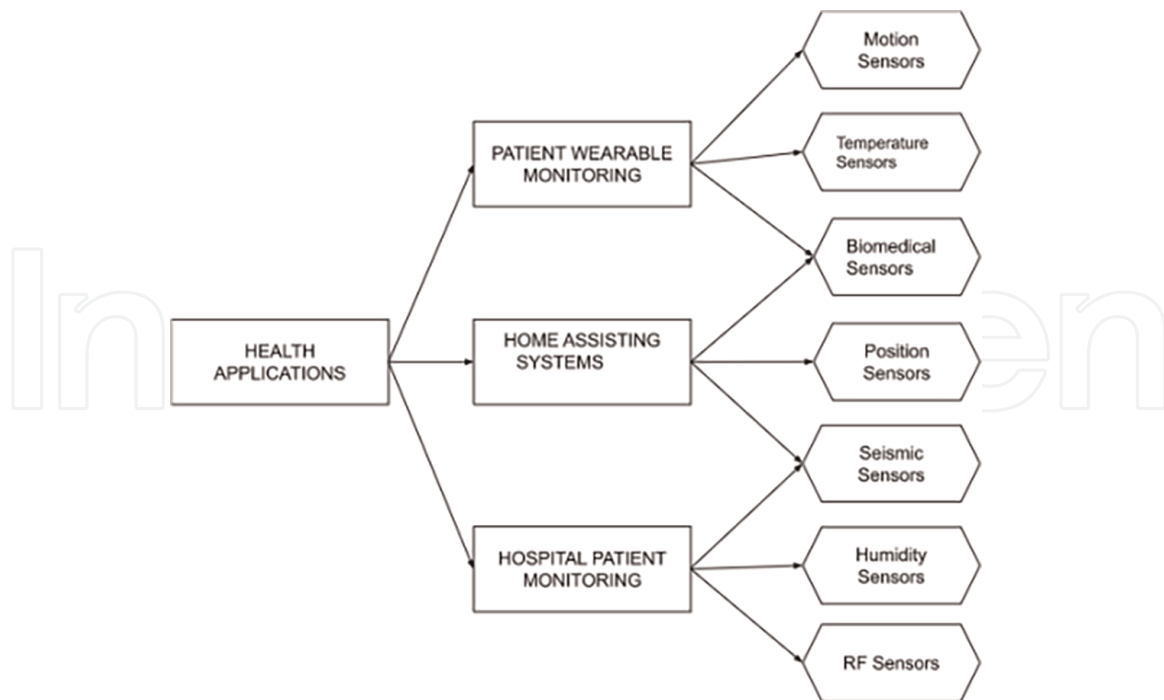


Figure 4.
Subcategories of Health Applications of WSN and the types of sensors used.

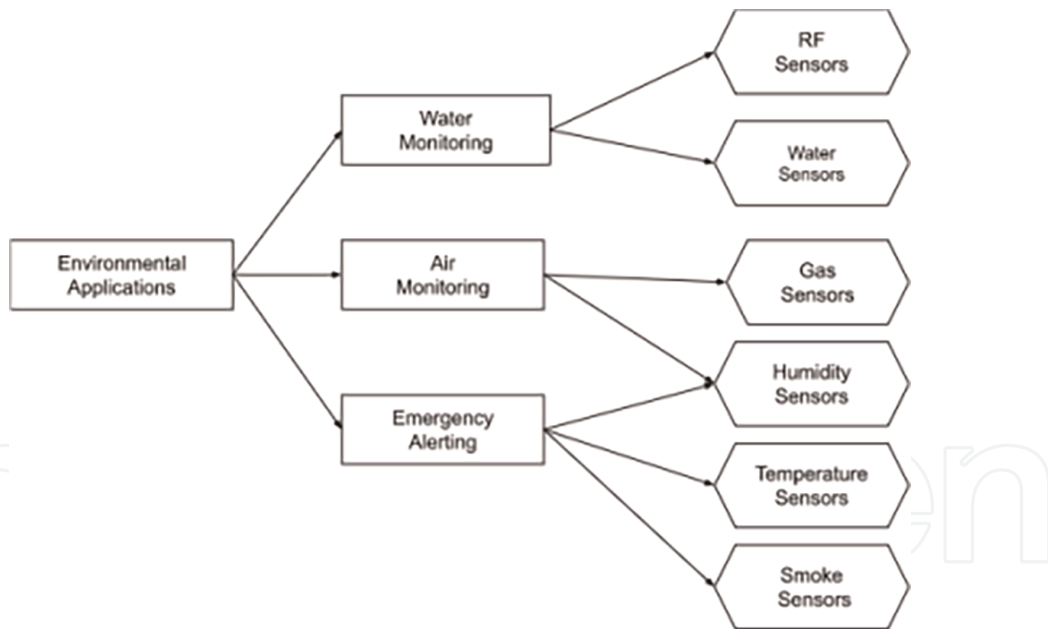


Figure 5.
Environmental Applications of WSNs and the types of sensors used by them.

with the usage of WSNs. The main sub-categories of environmental applications of WSNs such as water tracking [7], air tracking [8], and emergency alerting [9], are depicted alongside the types of sensors in **Figure 5**.

4. Flora and fauna applications: Both flora and fauna domains are vital for every country. The animal behaviour can be studied in some crucial areas, their tracking, and to control the use of wildlife passage by the local fauna [10]. The subcategories are mentioned in **Figure 6**.

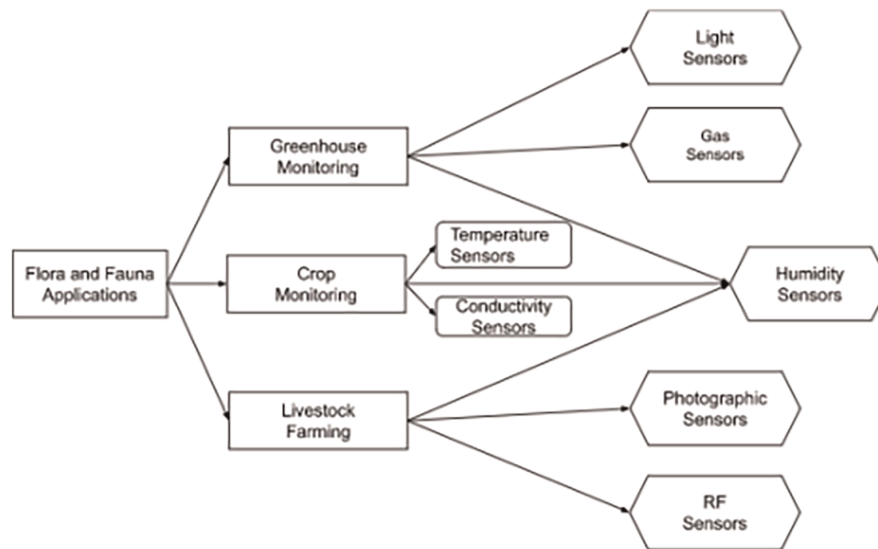


Figure 6.
Flora and Fauna Applications of WSNs and the types of sensors used by them.

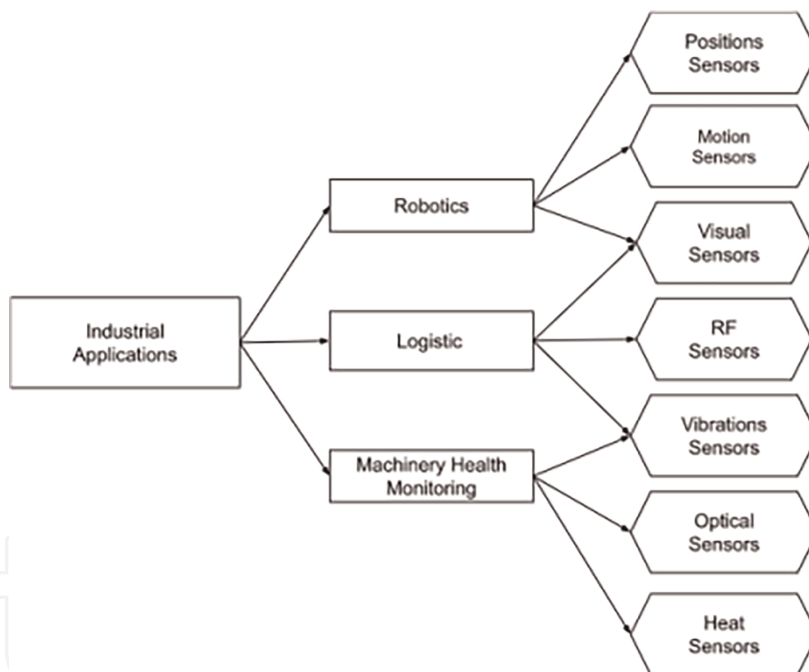


Figure 7.
Industrial Applications of WSNs and the types of sensors used by them.

5. Industrial applications: The main advantage of WSNs is the absence of any wiring due to which this can be integrated up to a larger scale as well. The main sub categories of industrial applications along with their sensors are given in **Figure 7**.

6. Urban applications: WSNs can be used to solve the various urban problems, for example, coordination of the specialised vehicles like ambulance, fire tenders, rescue vehicles, police automobiles, logistics of public transportation, traffic management, monitoring chemical/physical environmental parameters, building security and many others (**Figure 8**) [7].

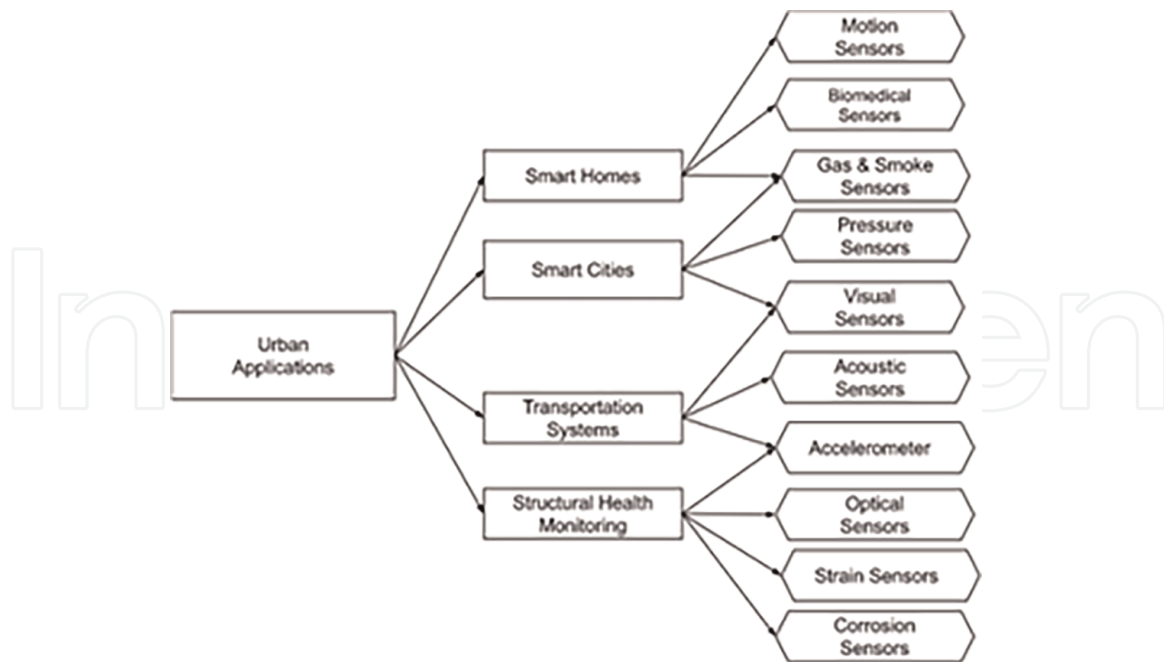


Figure 8.
Urban Applications of WSNs and the types of sensors used by them.

1.5 Issues in WSN

There are benefits and drawbacks to every technology. Similarly wireless sensor networks [11] also have some issues although being an excellent tool for application in many areas. The main problematic areas are: Design and the Topology.

A. Design issues

1. **Fault tolerance communication:** Due to the deployment of the sensor nodes in an out of control or harsh environment, it is not always uncommon for the sensor nodes to grow to be faulty and unreliable;
2. **Scalability:** A system whose overall performance improves after adding hardware proportional to the capability added is stated to be a scalable device. The variety of sensor nodes deployed within the sensing area can be in order of hundreds or thousands or even in millions;
3. **Low latency:** Low latency is imperative as the customers expect to interact with technology in real-time with no delays. Issues with high latency and time delays can cause users to quit and go for alternatives;
4. **Transmission media:** Long range transmission is generally point-to-point and calls for high transmission power with the chance of being eavesdropped, hence for better performance and security short range transmission can be opted;
5. **Coverage problems:** Coverage problems mean how to monitor a network problem effectively. It also reflects the quality-of-service (QoS) provided by the network.

B. Topology issues

1. **Geographic routing:** It is one of the most extensively used technique, however, the recent studies has shown that geographic routing can sometimes be useless in actual time deployments where location estimation system introduce regional errors;
2. **Sensor holes:** A routing hole is a region in the sensor network wherein nodes are not either available or if available these cannot participate in the actual routing of the data due to various possible reasons. The mission of identifying holes is specifically difficult considering the fact that ordinary wireless sensor networks encompass lightweight, low capability nodes, which can be ignorant of their geographic place;
3. **Coverage topology:** Coverage of the sensor network represents how well the sensors monitor a field of interest where they are deployed. It is the performance measure of the network sensing capability. Connectivity represents how well the nodes communicate.

C. Other issues are:

1. Synchronisation;
2. Computation and energy constraints;
3. Security;
4. Cost effective;
5. Limited bandwidth/framework;
6. Node costs;
7. Power management;

1.6 Need of machine learning in WSN

Machine learning [12] is a branch of Artificial Intelligence (AI). It is basically defined as the capability of a machine to mimic the behaviour of the human being that focuses on interpreting and analysing the patterns.

Meanwhile, its focus evolved and shifted more to the algorithms, which are more achievable and reliable. The machine learning techniques have been used extensively for variety of responsibilities which includes classification, regression, biometrics (such as speech recognition, eye detection, and fingerprint detection), fraud detection, etc. The sensor nodes in the WSNs might be heterogeneous, which are designed using numerous types of sensors according to the requirements of the network. The creators of network are more vulnerable to the issues regarding aggression or collection of statistics, reliability, clustering of nodes, safety and fraud detection [13]. Wireless sensor networks keep an eye on situations that are always changing. Either external factors or the system designers themselves started this dynamic behaviour.

Sensor networks frequently use machine learning techniques to adapt to such circumstances in order to avoid needless redesign. A lot of doable solutions that maximise resource usage and increase network longevity are also inspired by machine learning. Lately, the use of machine learning algorithms has been experienced in WSNs. It enhances the performance of the network without the need for reprogramming. The algorithms also extract different levels of abstractions needed to perform variety of tasks with limited or no intervention. Some of the algorithms of ML deal with the design and functional issues of the network which were stated above.

The machine learning techniques [14] are used for the following reasons:

- WSN reveal dynamic environments that change unexpectedly with time which are due to outside factors or by the designers themselves. The networks in such situations adopt the ML techniques to take away the need of unnecessary redesign;
- The machine learning additionally encourages many realistic solutions that maximise aid usage and prolong the lifespan of the network;
- The ML algorithms used to discover important correlations in the sensor data;
- WSN encouraged by means of machine studying strategies can offer low intricacy approximations for the gadget fashions, enabling its implementation within sensor nodes;
- It also offers low complexity mathematical models for complex environment;
- These are also efficiently followed for predicting future events based on the previous sensor network records.

2. Machine learning techniques for wireless sensor networks

Most of the machine learning algorithms falls into three broad categories:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

2.1 Supervised learning

Supervised Learning as the name indicates it means “having a mentor” to supervise, which means that the output is already present and the input gives the output accordingly. It contains a model that is used to predict the outcomes with the help of labelled dataset. The learned relationship between the input, output and the parameters of the system is learned by system model. The training is given to this model and once it is complete the model is tested on the basis of test data and then it predicts the output with the help of labelled dataset (it is a dataset wherein the target answer is already known) (**Figure 9**) [12].

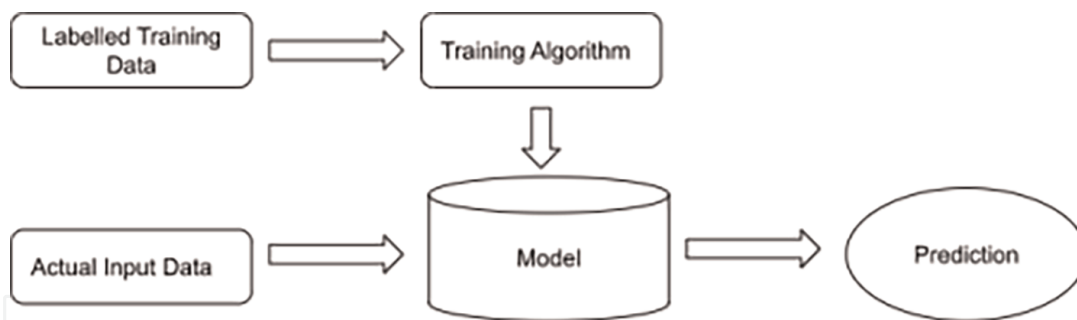


Figure 9.
Supervised learning model.

It also finds the mapping function to map the input variable and the output variable. This type of approach is used to solve diverse issues for WSN such as object targeting and localization, processing of query and event detection, medium access control, intrusion detection and security, image classification, spam filtering, data integration and security.

Guided learning involves the following steps:

- Initially, choose the training dataset type;
- Obtain the labelled training data;
- The data should be divided into a training dataset, a test dataset, and a validation dataset. The training dataset's input features should be identified, and they should contain sufficient information to allow the model to correctly predict the output;
- Choose an appropriate method for the model, such as a decision tree, support vector machine, etc.;
- Run the algorithm on the training dataset. Validation sets, a subset of training datasets, are occasionally needed as control parameters;
- By supplying the test set, you may determine whether the model is accurate. If the model accurately predicts the desired outcome then it is accurate.

The supervised learning can be further divided into two categories: *Regression* and *Classification*.

2.1.1 Regression algorithms

If there is a link between the input and output variables and the output has a real or continuous value, regression methods are used. This kind of learning strategy is employed in situations where a relationship between two variables and the changes in one affects the others. In simple words, "Regression exhibits a line or curve that traverses through all the data points on target-predictor graph in such a way that the distance between the data points and the regression line is small". Prediction, forecasting, time series modelling, and establishing the causal connection between variables are its key applications.

Some of the few examples of regression are as follows:

- Prediction of rain using temperature and other factors;
- Determining Market trends;
- Prediction of road accidents due to rash driving.

2.1.2 Classification algorithms

These algorithms are used when we have to classify something into group after getting output into a category. It is a method for classifying observations into several groups according to a condition. Binary classification, such as Yes-No, Male-Female, True-false, etc., refers to an algorithm's attempt to categorise data into two separate groups. Multiclass classification is the practise of choosing from more than two categories. In the classification algorithm, an input variable is transferred to a discrete output function (y) (x).

$$y = \text{categorical output}, f(x) = y$$

The primary goal of classification algorithms is to determine the category of a given dataset, and these algorithms are primarily employed to forecast the results for categorical data. Email Spam Detector is the best illustration of an ML classification method. It can also be used to categorise various objects, such as fruits, according to their taste, colour, size, etc. When given new data, a machine that has been properly trained using a classification method may predict the class with ease. It can classify everything, including fruit, automobiles, houses, signs, and more. However, there are numerous ways to perform a same task in order to predict whether a given person is male or a female, a machine has to be trained first and further there are numerous ways in doing so. For predictive analytics some of the commonly used algorithms are:

1. Decision tree;
2. Random forest tree;
3. Bayesian statistics;
4. Support vector machine (SVMs);
5. KNN.

2.1.2.1 Decision tress

Decision Tree is a supervised learning technique that is used to effectively handle non-linear data sets [15]. It is a classification learning algorithm. It is a visual representation of every scenario that could lead to a choice. It solves a problem by using a tree representation. The decision tree, as shown in **Figure 10**, resembles a tree that has two different types of nodes: the Choice (Decision) node and the Leaf node. Decisions are made using choice nodes, which have numerous branches, whereas leaf nodes are

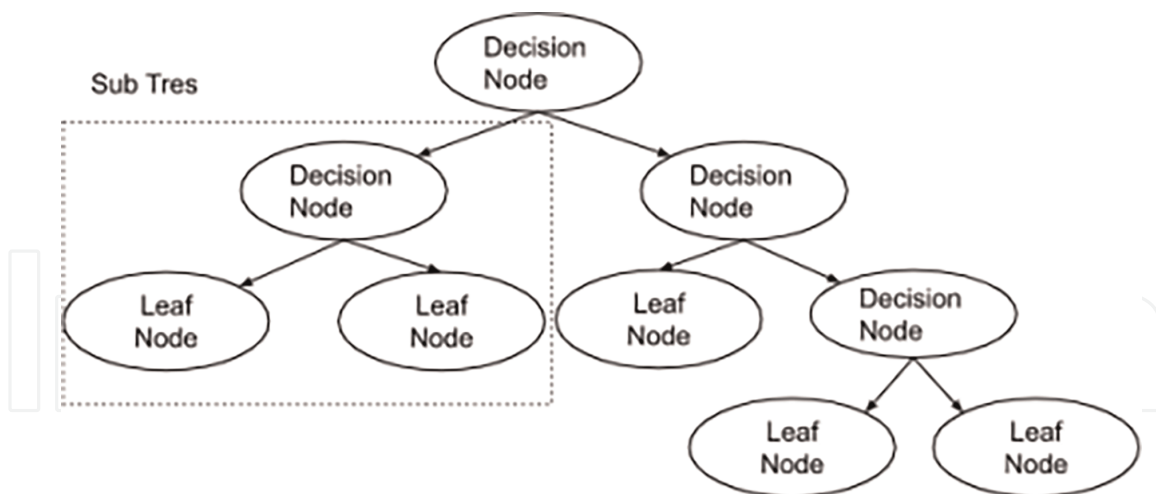


Figure 10.
Decision tree.

the results of those decisions and do not have any additional branches. CART, or Classification and Regression Tree Algorithm, is employed in this case to construct a decision tree. It simply asks a question and on the basis of that (either YES/NO) the tree is split further.

The fundamental idea of the decision tree is to test the most significant attribute first. The most important attribute is the one that impacts the classification of an example the most. This will ensure that we obtain the accurate classification with smaller number of tests wherein all the paths in the tree are short and the tree in general is a small one. Decision tree algorithms are used to solve many unresolved issues in WSNs layouts like link loss, reliability, restore, corruption rate, and mean-time-to-failure (MTTF).

2.1.2.2 How decision tree algorithm works?

In the decision tree (DT), for predicting the class of the given dataset, the algorithm begins from the foundation node i.e., the decision node or root node of the tree. The algorithm compares the value of root characteristics with the record (real dataset) characteristics and based on the comparison the node is split on. For the subsequent node, the algorithm once more compares the attribute value with the alternative sub-node and circulates similarly. It keeps the manner until it reaches the leaf node of the tree. The whole technique can be better understood by the use of below algorithm:

- Step 1:** Start the tree with the foundation node (let suppose S), which contains the complete dataset;
- Step 2:** Find the high quality attribute within the dataset using the characteristics attribute selection measure (ASM);
- Step 3:** Divide the foundation node(S) into the sub-nodes that contains the feasible values for the high quality attributes;
- Step 4:** Generate the decision tree node, which incorporates the most appropriate value or attribute;
- Step 5:** Recursively make new choices by using the subsets created in the step 3;

Maintain this process until a degree is reached wherein you cannot further classify the nodes known as the final node known as the leaf node.

2.1.2.3 Attribute selection measures (ASM)

Whilst enforcing a decision tree, the principle issue arises that how to select the first rate attribute for the foundation node and for sub nodes. So, to resolve such issues there may be a technique, which is known as characteristic choice measure or ASM. With this one can effortlessly choose the first-rate attribute for the nodes of the tree.

The two basic techniques for ASM are:

1. Information gain;
2. Gini Index.

Information advantage: Information gain is the clever idea for defining impurity. Impurity measure is a heuristic for selection of the splitting criterion that separates a given dataset of class labelled training tuples into individual classes. If we divide D into smaller partitions as per the outcomes of the splitting criterion, each partition should ideally be pure with all the tuples falling into each partition belonging to the same class. It is the change in entropy after the segmentation of a dataset primarily based on characteristics. It calculates how tons of information is provided about a category. Consistent with the cost of records gain, the node can be split up and build the selection tree.

The information gain can be calculated by using the following notations:

$$IG = \text{Entropy (S)} - [\text{weighted average}^* \text{ Entropy (each feature)}]$$

Entropy is the measure of randomness of the data. It can be calculated as:

$$E = -(\text{Probability of yes}) - (\text{Probability of no})$$

$$E = -P(\text{yes}) \log_2 P(\text{yes}) - P(\text{no}) \log_2 P(\text{no})$$

Gini Index: It is the degree of impurity or purity used at the same time as developing a decision tree for CART algorithm. An attribute with a low Gini index must be desired in comparison to the high Gini index. It only creates the binary splits, which are done by the algorithm by using the Gini index. It can be calculated by the usage of following method:

$$GI = 1 - \sum_j P_j^2$$

Some pros of using a decision tree:

1. It is straightforward to apprehend because it follows the same system, which humans observe at the same time as making any choice-related issues;
2. In comparison to different algorithms selection criteria requires much less effort for the information training throughout pre-processing;

3. A decision tree does not require any kind of normalisation of the records;
4. A decision tree does not require scaling of the statistics;
5. The missing values inside the information (datasets) also does not affect the process of building a decision tree to any extensive extent;
6. This type of tree version is very intuitive and easy to explain to the technical groups also.

Cons of using a decision tree:

1. A small change inside the records can reason a large change within the shape of the choice tree inflicting instability;
2. They are vulnerable to overfitting;
3. For a decision tree some random calculation can go far more complicated compared to other algorithms;
4. This often entails higher time to educate the model;
5. It is cost effective sometimes because of its complexity as it may contain lots of layers;
6. It is insufficient for applying regression and predicting continuous values.

Decision trees are frequently used in WSNs because they describe relationships between attributes and classes in a clear and understandable way. A decision tree's paths are made up of a series of conditions that each describes a class. Such decision tree paths can be used to develop rules that can be employed in a WSN to distinguish between different outcomes or phenomena based on measurements taken from sensed data. A decision tree's clarity provides important insights because of the open model learning approach it employs. They are also used because of their simplicity and interpretability in their regulations that can be easily derived from the shape of the tree. It also identifies hyperlink reliability, which is very successful but this set of rules works best with linearly separable records only.

2.1.2.4 Random forest tree

Random Forest tree, as shown in **Figure 11**, is a famous gadget mastering algorithm that belongs to the supervised learning approach. It is able to be used for both type and Regression issues in ML. It is primarily based on the concept of ensemble gaining knowledge, which is a system of mixing a couple of classifiers to solve a complicated trouble and to enhance the performance of the model.

Random forest is a classifier that incorporates a number of decision bushes on numerous subsets of the given dataset and takes the common to enhance the predictive accuracy of that dataset. In place of relying on one selection tree, this algorithm takes the prediction from every tree and based on most people votes of predictions, and it predicts the final output.

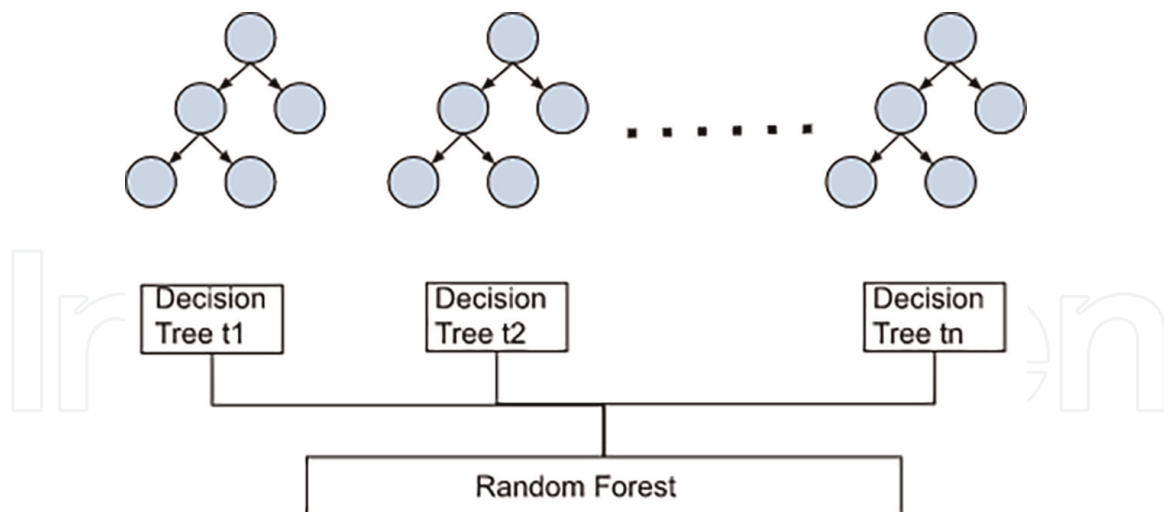


Figure 11.
Random forest tree.

2.1.2.5 How does random forest algorithm work?

Random forest algorithm works in two-section first is to create the random forest by using combining N choice tree, and 2-D is to make predictions for every tree created in the first section.

The running method may be defined as follows:

- Step-1:** Pick out random k records points from the education set;
- Step-2:** Construct the decision timber associated with the selected information factors (Subsets);
- Step-3:** Choose the variety N for choice bushes that you want to construct;
- Step-4:** Repeat Step 1 and 2;
- Step-5:** For brand new statistics points, discover the predictions of each selection tree, and assign the brand new data factors to the class that wins the majority votes.

Some pros of using random forest algorithm:

1. Sturdy to outliers;
2. Works well with non-linear records;
3. Decrease threat of overfitting;
4. Runs effectively on a huge dataset;
5. Higher accuracy than other type algorithms;
6. No function scaling required;
7. This algorithm can robotically take care of lacking values;

8. This algorithm may be very stable. Despite the fact that a brand new data factor is delivered in the dataset, the general set of rules is not always affected a great deal since the new facts might also impact one tree, but it is very difficult for it to affect all the timber.

9. It is much less impacted by way of noise.

Some cons of using random forest algorithm:

1. Random forests are found to be biased while handling express variables;
2. Sluggish training;
3. Now not suitable for linear techniques with a whole lot of sparse functions.

WSN is a difficult problem due to the diversity of deployment and the restrictions within the sensors' sources. This supervised device mastering-based total approach is considered to scrutinise the behaviour of sensors through their statistics for the detection and prognosis of faults. Maximum of the faults that generally arise in WSN are considered: handover, glide, spike, erratic, information-loss, stuck, and random fault. A hybrid strategy was put forth [16] for real-time network intrusion detection systems (NIDS). For feature selection, they use the random forest (RF) algorithm. In order to remove the unnecessary features, RF presents the variable importance as numeric values. The experimental findings demonstrate that the new strategy is quicker and lighter than the prior methods while still ensuring high detection rates, making it appropriate for real-time NIDS.

2.1.2.6 Bayesian statistics

Bayesian records are the mathematical method for calculating possibilities wherein inferences are subjective and get updated while extra facts are delivered. This record is in comparison with classical or frequentist information where probability is computed through comparing the frequency of a specific random occasion for an extended length of repeated trials where inferences are intended to be the goal. Those statistical inferences are the manner of extracting conclusions out of massive datasets via studying a small portion of sample statistics. For this, the data professionals:

- First examine the pattern information and extract the belief, this is called as prior inference;
- After this, they check another sample of records and revise their end, this revised data is known as posterior inferences.

As Bayesians, a concept of a notion, called a previous, gain some information and use it to update the notion. The final results are called a posterior. As attain even greater facts, the antique posterior becomes a brand new prior and the cycle repeats.

This system employs the Bayes rule:

$$P(A|B) = P(B|A) * P(A)/P(B)$$

$P(A|B)$, examine as “possibility of A given B”, shows a conditional chance: how possibly is A if B happens.

In WSNs, these styles of Bayesian learners are useful for assessing event consistency. Numerous variations of Bayesian newcomers allow better getting to know of relationships, consisting of Gaussian combination fashions, Dynamic Bayesian Networks, Conditional Random Fields as well as Hidden Markov fashions.

2.1.2.7 Support vector machines (SVMs)

One of the most well-liked supervised learning algorithms, Support Vector Machine (SVM) [17] is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification issues. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allow to quickly classifying fresh data points in the future. A hyper plane is the name given to this optimal decision boundary.

SVM selects the extreme vectors and points that aid in the creation of the hyper plane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method. Take a look at the **Figures 12** and **13**, where two distinct categories are identified using decision boundary:

In n-dimensional space, there may be several lines or decision boundaries used to divide classes; however, the optimal decision boundary for classifying the data points must be identified. The hyperplane of SVM is a name for this optimal boundary. The features of dataset determine the dimensions of hyper plane, therefore, if there are just two features, as shown in **Figure 12** [17], the hyperplane will be a straight line. Additionally, if there are three features, the hyperplane will only have two dimensions.

Support Vectors: The data points or vectors that are closer to the hyperplane and influence the position and orientation of the hyperplane. The SVM method aids in identifying the ideal decision boundary or region, often known as a hyperplane. The SVM algorithm determines which line from each class is closest to the other. Support

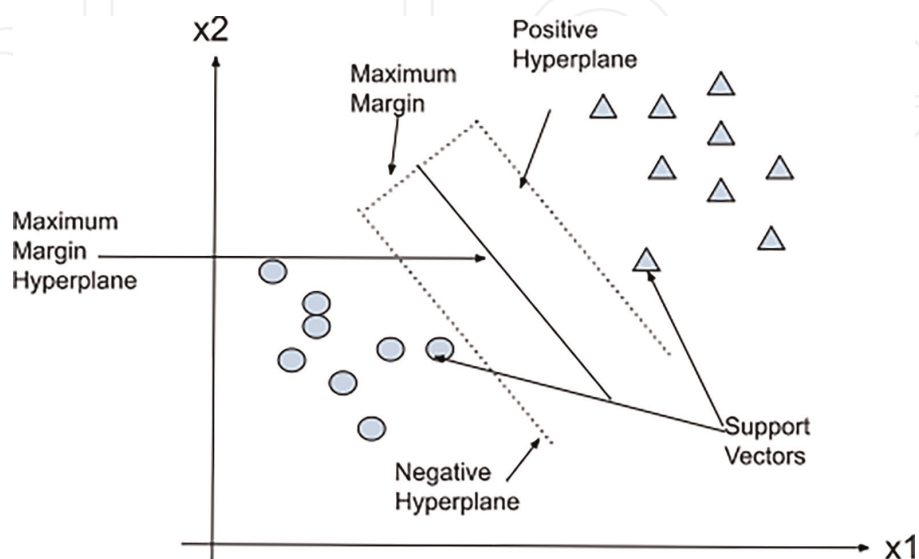


Figure 12.
Support Vector Machine (SVM).

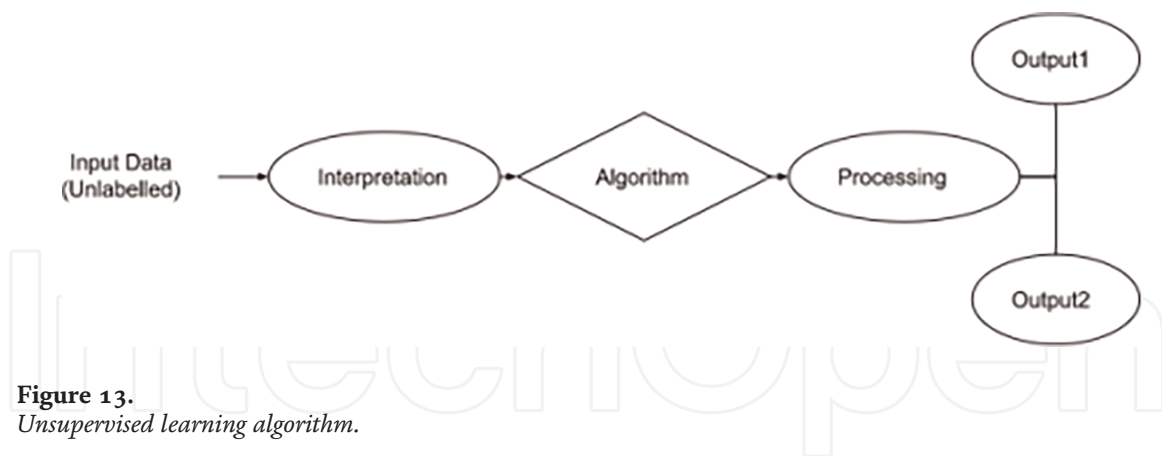


Figure 13. Unsupervised learning algorithm.

vectors are the names for these points. Margin is the distance between the hyperplane and the vectors. Maximising this margin is the aim of SVM. The ideal hyperplane is the one with the largest margin.

SVM can be categorised in two different ways:

Linear SVM: Linear SVM is used for linearly separable records, which means that if a dataset may be categorised into instructions by the usage of a single straight line, then such facts is called as linearly separable statistics, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated information, which means that if a dataset cannot be categorised by way of the use of a directly line, then such data is termed as non-linear statistics and classifier used is referred to as Non-linear SVM classifier.

Face identification, image classification, text categorisation, grouping of portrayals, bio-informatics, handwriting remembrance, etc. may all be done using the SVM method.

Some pros of using the SVM algorithm:

1. Support vector machines perform similarly well when there is a discernible margin of class dissociation;
2. High dimensional rooms are more productive;
3. When the quantity of dimensions exceeds the quantity of examples, it works well;
4. This algorithm is very much similar to memory systematic.

Some cons of using this algorithm:

1. For huge data sets, the support vector machine approach is unacceptable;
2. When the target classes overlap and the data set has more sound, it does not operate very well;
3. The support vector machine will perform poorly when there are more attributes for each data point than there are training data specimens;

4. There is no classification error since the support vector classifier places data points above and below the classifying hyperplane.

2.1.2.8 K-Nearest Neighbour (K-NN)

K-Nearest Neighbour is one of the most effective machines gaining knowledge of algorithms based totally on Supervised getting to know method. The K-NN algorithm makes the assumption that the new case and the existing cases are comparable, and it places the new instance in the category that is most like the existing categories. A new data point is classified using the K-NN algorithm based on similarity after all the existing data has been stored. This means that utilising the K-NN method, fresh data can be quickly and accurately sorted into a suitable category. Although the K-NN approach is most frequently employed for classification problems, it can also be utilised for regression. Since K-NN is a non-parametric technique, it makes no assumptions about the underlying data [18].

It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to perform an action when classifying data. The KNN method simply saves the information during the training phase, and when it receives new data, it categorises it into a category that is quite similar to the new data.

The K-NN algorithm is excellent for WSN query processing jobs because of its simplicity.

The following algorithm can be used to describe how the K-NN works:

Step 1: Decide on the neighbours K-numbers;

Step 2: Calculate the Euclidean distance (or Hamming Distance) between K neighbours. The distance between two points, which we have already examined in geometry, is known as the Euclidean distance;

E.g.: Let there be two points $A(x_1, y_1)$ and $B(x_2, y_2)$. Now the Euclidean distance between them can be calculated as: $ED = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Step 3: Based on the determined Euclidean distance, select the K closest neighbours;

Step 4: Count the number of data points in each category among these k neighbours;

Step 5: Assign the fresh data points to the category where the neighbour count is highest;

Step 6: Model is complete.

Some pros of using the KNN algorithm:

1. It is straightforward to put in force;
2. It's far strong to the noisy schooling records;
3. It can be extra powerful if the training facts are huge.

Some cons of using the KNN algorithm:

1. K's value must always be determined, and sometimes that can be difficult;

2. The high computation cost is caused by the need to determine the separation between each data point for each training sample.

2.2 Unsupervised learning

In supervised machine learning, models are trained on labelled data while being watched over by training data. However, there may be several instances when lacking labelled data and need to identify hidden patterns in the supplied dataset. Therefore, one needs unsupervised learning strategies to handle these kinds of problems in machine learning. Unsupervised learning is a subcategory of machine learning wherein models are trained using unlabelled datasets and are free to operate on the data without being checked by a human observer.

Because unlike supervised learning, one has the input data but no corresponding output data, unsupervised learning cannot be used to solve a regression or classification problem directly. Finding the underlying structure of a dataset, classifying the data based on similarities, and representing the dataset in an unsupervised way and in the compressed format are the objectives of unsupervised learning [19].

The following are a few key arguments for the significance of unsupervised learning:

1. Finding valuable insights from the data is made easier with the aid of unsupervised learning;
2. Unsupervised learning is considerably more like how humans learn to think via their own experiences, which brings it closer to actual artificial intelligence;
3. Unsupervised learning is more significant because it operates on unlabelled and uncategorized data;
4. Unsupervised learning is necessary to handle situations when the input and output are not always the same in the real world;

According to the **Figure 13**, input data is unlabelled, which means that neither its category nor any associated outputs are provided. Now, the machine learning model is being trained using the unlabelled input data. It will first evaluate the raw data to identify any hidden patterns in the data before applying the appropriate algorithms. The unsupervised learning algorithm has two subtypes: Clustering and Association.

2.2.1 Clustering

Clustering is a way of organising items into clusters so that the items that share the most similarities stay in one group and share little to none with the objects in another group. The data items are classified based on the existence or lack of commonalities discovered during the cluster analysis.

2.2.2 Association

A rule of association is used to uncover links between variables in a sizable database using unsupervised learning techniques. It establishes the group of items that co-occur in the collection. Marketing strategy is more effective because to the association

rule. Those who purchase X (let us say, bread) also frequently buy Y (let us say, butter or jam). Market Basket Analysis is an illustration of an association rule in action.

Popular unsupervised learning algorithms are listed below:

A. K-means clustering; and

B. Principal component Analysis.

2.2.2.1 K-means clustering

The unlabelled dataset is divided into k different clusters using an iterative process. Each cluster comprises just one dataset and has a unique set of properties. The data objects are divided into separate clusters using an unsupervised learning algorithm, which also serves as a useful technique for automatically identifying group categories in unlabelled datasets without the need for training. As each cluster is linked to a centroid, the method is centroid-based. The main objective of this approach is to minimise the overall distances between the data points and the clusters they belong to. Because it is straightforward and linear in complexity, the K-means clustering algorithm is used for clustering WSN sensor nodes and is useful for finding the cluster heads as well.

The K-means method is demonstrated in the phases below:

1. Choose K to determine the total number of clusters;
2. Choose K points or the centroid at random in next step;

(That may not be the dataset that was provided.)

1. Assign each data point to its nearest centroid, which will produce the predetermined K clusters;
2. Locate the centroid of each cluster and calculate the variance;
3. Repeat the third step to assign each data point to its new centroid in this step;
4. Go to step 4 if a reassignment is necessary; otherwise, move to final stage;

Some pros of using K-means:

1. Easy to put into practice;
2. Big data sets are scaled;
3. Ensures convergence;
4. Can warm-up the centroids locations;
5. Adapts readily to new examples; and
6. Broadens to include clusters of various sizes and shapes, such as elliptical clusters.

Some cons of using K-means:

1. It is challenging to estimate the k-value, or the number of clusters;
2. Initial inputs like the number of clusters in a network have a significant impact on output (value of k);
3. The order in which the data is entered significantly affects the result;
4. Rescaling makes it pretty sensitive. Using normalisation or standards to rescale our data will result in a very different result. Final outcome; and
5. When clusters have a complex geometric shape, clustering activities should not be performed.

2.2.2.2 Principal component analysis (PCA)

Principal component analysis is the most effective unsupervised learning technique for reducing the dimensionality of data. It simultaneously reduces information loss while increasing interpretability. It facilitates the identification of the dataset's most crucial qualities and makes data easier to plot in 2D and 3D. PCA facilitates the discovery of a series of linear combinations of variables. The Main Components are the names given to these newly altered functions. One of the most well-known pieces of equipment for exploratory information evaluation and predictive modelling is this [20].

Typically, PCA looks for the surface with the lowest dimensionality onto which to project the high-dimensional data. PCA functions by taking into account each attribute's variance since a high attribute demonstrates a solid split between classes, which results in low dimensionality.

Since it uses a feature extraction technique, it keeps the crucial variables and discards the unimportant ones.

Some of the important additives of principal components are given below:

- The number of these components is either equal to or less than the original functions gift inside the dataset;
- The major element ought to be the linear combination of the unique capabilities;
- These components are orthogonal, i.e., the correlation between a couple of variables is zero; and
- The significance of every element decreases while going to 1 to n, it manner the 1 pc has the maximum importance, and n laptop will have the least significance.

2.2.2.3 Steps for PCA algorithm

Step 1: To obtain the dataset: First, split the input dataset into two halves, X and Y, where X represents the training set and Y represents the validation set;

Step 2: Putting information into a structure: Now create a structure to represent dataset, and use the two-dimensional matrix of independent variable X as an

example. Here, each row represents a data item and each column represents a feature. The dataset's dimensions are determined by the number of columns;

Step 3: Data standardisation: Normalise the dataset in this stage. For instance, in a given column, features with higher variation are more significant than features with smaller variance. Split each piece of data in a column by the column's standard deviation if the importance of features is independent of the variance of the feature. The matrix in this case is called Z ;

Step 4: Determining Z 's covariance: Transpose the Z matrix in order to determine Z 's covariance. Transpose it first and then multiply it by Z . The Covariance matrix of Z will be the output matrix;

Step 5: Making the Eigen Values and Eigen Vectors calculations: The resulting covariance matrix Z 's eigen values and eigenvectors must now be determined. The high information axis' directions are represented by eigenvectors or the covariance matrix. Additionally, the eigen values are defined as the coefficients of these eigenvectors;

Step 6: Sorting of the Eigen Vectors: This phase involves taking all of the eigen values and sorting them from largest to lowest in a decreasing order. Additionally, in the eigen values matrix P , simultaneously sort the eigenvectors in accordance. The matrix that results in is known as P^* ;

Step 7: Principal Components or the new features calculation: Compute the new features here, and then multiply the P^* matrix by Z to achieve this;

Step 8: Eliminate less significant or irrelevant features from the new dataset: Decide here what to keep and what to eliminate now that the new feature set has been implemented. Retain relevant or significant features in the new dataset and exclude irrelevant information.

The PCA algorithm allows the minimal variance components to be dropped because they simply have the least amount of information and reduce dimensionality. This could reduce the amount of data being communicated between sensor nodes in WSN scenarios by obtaining a small pair of uncorrelated linear combined innovative readings. By permitting the selection of only significant principle components and eliminating other lower order inconsequential components from the model, it can also turn the problem of vast data into one of tiny data.

2.3 Reinforcement learning

The enormous amount of data that models need to train on is a recurring problem in machine learning. A model may need more data the more complicated it is. After all of this, the information received might not be trustworthy. It could be inaccurate, missing, or compiled from unreliable sources. Data acquisition is solved through Reinforcement Learning, which virtually eliminates the requirement for data.

A subfield of machine learning called reinforcement learning develops a model's ability to solve issues at their best on its own. With reinforcement learning, a computer learning model must analyse the issue and find the best solution on its own. This implies that we also come up with quick and original solutions that the programmer might not have even considered. A certain class of problems, such as those in robotics, gaming, and other long-term endeavours, are solved using RL.

Key principles of reinforcement learning

- In real life, the agent is not given instructions regarding the surroundings or what needs to be done;
- It is founded on the hit-and-miss method;
- The agent performs the subsequent action and modifies its states in response to feedback from the preceding action;
- The agent might receive a reward afterwards; and
- The agent must investigate the stochastic environment in order to maximise positive rewards.

Reinforcement learning's advantages

- Reinforcement learning can be used to resolve extremely complicated issues that cannot be resolved using traditional methods;
- To attain long-term effects, which are very challenging to achieve, this method is favoured;
- The way that humans learn is remarkably similar to this learning model;
- The model is capable of fixing mistakes that happened during training;
- The likelihood of experiencing the same error after a model has rectified one is quite low; and
- To tackle a certain issue, it might produce the ideal model.

The drawbacks of reinforcement learning

- In many ways, the framework of reinforcement learning is flawed, but it is precisely this flaw that makes it valuable.
- A state overload brought on by excessive reinforcement learning may have a negative impact on the outcomes.
- Using reinforcement learning to solve straightforward issues is not recommended.
- Both a lot of data and a lot of compute are required for reinforcement learning. It craves information. Because one can play video games repeatedly, gathering a lot of data looks doable; this is why it works so well in them.

Algorithms for reinforcement learning can be used by robots to teach themselves to walk. The mainly used algorithm in reinforcement learning is: **Q learning**.

2.3.1 Q learning

Given the agent's present state, Q-learning [21] is a model-free, off-policy reinforcement learning technique that will determine the appropriate course of action.

The agent will choose what to do next base on its location in the surroundings. The model's goal is to determine the optimum course of action given the situation as it is. In order to accomplish this, it might devise its own set of rules or might deviate from the prescribed course of action. This indicates that there is no real need for a policy, which is why it is referred to as off-policy. Model-free refers to the use of predictions about the environment's anticipated response by the agent in order to make decisions. It instead relies on trial and error learning rather than the reward system.

A recommendation system for advertisements is an illustration of Q-learning. The advertisements you see in a typical ad suggestion system are determined by your past purchases or websites you may have visited. In the event that you have already purchased a TV, various brand TVs will be suggested. A distributed architecture such as a wireless sensor network, where each node conducts activities that are anticipated to optimise its long-term benefits can readily implement this algorithm.

3. Conclusion

Due to the numerous ways that wireless sensor networks differ from regular networks, there is a need for protocols and tools that address particular problems and constraints. This chapter provided an empirical study of the wireless sensor network infrastructure, including its architecture, applications, and limitations. These networks consequently need novel approaches to routing, security, scheduling, localization, node clustering, data aggregation, fault detection, and data integrity that are both energy conscious and real-time. After that, the Chapter provided the taxonomy of ML algorithms that were applied to WSNs and performed a quantitative analysis of the algorithms that helped WSNs to overcome those limitations. Furthermore, varieties of methods are offered to improve a wireless sensor network's capacity to adjust to the changing behaviour of its environment. We also highlighted each ML algorithm's benefits and drawbacks.


However there is still a lot of work being done to address numerous outstanding issues because applications of machine learning methods to wireless sensor networks are still a relatively new area of study.

Author details

Rakesh Chandra Gangwar* and Roohi Singh
Department of Computer Science and Engineering, Sardar Beant Singh State
University, Gurdaspur, Punjab, India

*Address all correspondence to: rakeshgangwar@sbssugsp.ac.in

IntechOpen

© 2023 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Akyildiz IF, Weilian S, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. *IEEE Communications Magazine*. 2002;**40**(8): 102-114
- [2] Romer K, Mattern F. The design space of wireless sensor networks. *IEEE Wireless Communications*. 2004;**11**(6): 54-61
- [3] Kalantary S, Taghipour S. A survey on architectures, protocols, applications and management in wireless sensor networks. *Journal of Advanced Computer Science & Technology*. 2014;**16**:1-11
- [4] Đurišić MP, Tafa Z, Dimić G, Milutinović V. A survey of military applications of wireless sensor networks. In: *Proceedings of the 2012 Mediterranean Conference on Embedded Computing (MECO)*; Bar, Montenegro: IEEE (Piscataway); 2012. pp. 19-21
- [5] Bokareva T, Hu W, Kanhere S, Ristic B, Gordon N, Bessell T, et al. Wireless sensor networks for battlefield surveillance. In: *Proceedings of the Land Warfare Conference*. Brisbane, Australia: MDPI (Switzerland); 2006. pp. 1-8
- [6] Hii P, Chung W. A comprehensive ubiquitous healthcare solution on an android mobile device sensors. *Sensors (Basel)*. 2011;**11**(7):6799-6815
- [7] Nasir A, Soong BH, Ramachandran S. Framework of WSN based human centric cyber physical in-pipe water monitoring system. In: *Proceedings of the 2010 11th International Conference on Control Automation Robotics & Vision*; Singapore: IEEE (Piscataway); 2010. pp. 7-10
- [8] Mansour S, Nasser N, Karim L, Ali A. Wireless sensor network-based air quality monitoring system. In: *Proceedings of the 2014 International Conference on Computing, Networking and Communications (ICNC)*, 3–6 February; Honolulu, HI, USA: IEEE (Piscataway); 2014. pp. 545-550
- [9] Khedo KK, Bissessur Y, Goolaub DS. An inland wireless sensor network system for monitoring seismic activity. *Future Generation Computer Systems*. 2020;**105**:520-532
- [10] Kassim M, Harun AN. Applications of WSN in agricultural environment monitoring systems. In: *Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC)*, 19–21 October; Jeju, Korea: IEEE (Piscataway); 2016
- [11] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: A survey. *Computer Networks*. 2002;**38**(4): 393-422
- [12] Praveen KD, Amgoth T, Annavarapu CSR. Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion*. 2019;**49**: 1-25
- [13] Javaid A, Javaid N, Wadud Z, Saba T, Sheta OE, Saleem MQ, et al. Machine learning algorithms and fault detection for improved belief function based decision fusion in wireless sensor networks. *Sensors*. 2019;**19**:1334
- [14] Di M, Joo EM. A survey of machine learning in wireless sensor networks from networking and application perspectives. In: *6th International Conference on Information, Communications Signal Processing*. New York: Springer; 2007. pp. 1-5

[15] Sarker IH, Colman A, Han J, Khan AI, Abushark YB, Salah K. BehavDT: A behavioral decision tree learning to build user-centric context-aware predictive model. *Mobile Networks and Applications*. 2020;25:1151-1161

[16] Lee SM, Kim DS, Park JS. A hybrid approach for real-time network intrusion detection systems. *IEEE Transactions on Vehicular Technology*. 2011;60:457-472

[17] Pisner DA, Schnyer DM. *Machine Learning: Support Vector Machine*. Amsterdam: Elsevier; 2020. pp. 101-121

[18] Dey A. Machine learning algorithms: A review. *International Journal of Computer Science and Information Technologies (IJCSIT)*. 2016;7(3):1174-1179

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

[20] Feldman D, Schmidt M, Sohler C, Feldman D, Schmidt M, Sohler C. Turning Big Data into Tiny Data: Constant Size Coresets for k-Means, PCA and Projective Clustering. New Orleans, USA: SODA-2013; 2013. pp. 1434-1453

[21] Watkins C, Dayan P. Q-learning. *Machine Learning*. 1992;8(3-4):279-292