IOT_Phase2 **SAKTHI**

Machine Learning Algorithm to Analyse Water Consumption Patterns and Suggestions

Machine learning as a subset of AI is widely used these days in various fields such as industry, health, environment, energy, and municipal utilities. Machine learning is quite well-known as an efficient technology in future prediction because of its ability to find data patterns from past data. Self-learned and automatic improvement through experience are two main remarkable features of machine learning: working with various types of data, applying different algorithms and statistical techniques, big data handling, data analysis, and future prediction. Figure 2.4 shows the machine learning lifecycle. First, we define the business problem and specify the prediction aim. Then we prepare the data collected and select the appropriate data in the analysis step for utilizing in machine learning. In the Model step, we try to build a model based on our target variable and select features that affect the target value and the prediction. In the next step of the machine learning process, this model makes a pattern from the sample or primitive data. When new data enters, the model trains the data to test the relation between new data and the primitive pattern for prediction.

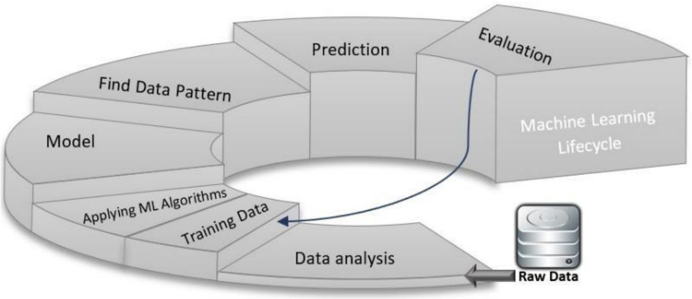

Figure 2.4.Machine Learning Lifecycle

2.7.1 Types of Machine Learning Algorithms

The first step in this section is about which type of machine learning tasks is suitable for our study for selecting the proper machine learning algorithms. Indeed, machine learning problems are divided into two tasks: supervised learning that works with the labeled data and unsupervised learning that works with unlabeled data (Figure 2.5). Labeled data means the input data or samples come with a label (tag) such as name, number, or type. Therefore, in this study, we describe the supervised learning task because we had labeled data. Supervised learning investigates the raw input data. When new data enters as an input, supervised learning algorithms try to produce the correct label for new data. Indeed, the supervised learning algorithms carry out this through the training data analysis and create a labeled output. This model predicts the future output based on available evidence. The evidence is available raw input or primitive sample of the labeled dataset that the predictive model has shaped based on them. Therefore, based on our dataset in this study, we had a supervised machine learning task that makes a predictive model.

Some combined regression models have been used in most of the research as a statistical tool because of their ability to forecast the target value with continuous values. The regression model is based on supervised learning. There are different types of regression like Linear regression, Support Vector regression (SVR), Decision Tree regression, KNN regressor, AdaBoost regressor, Ridge regressor, and Random Forest regression.

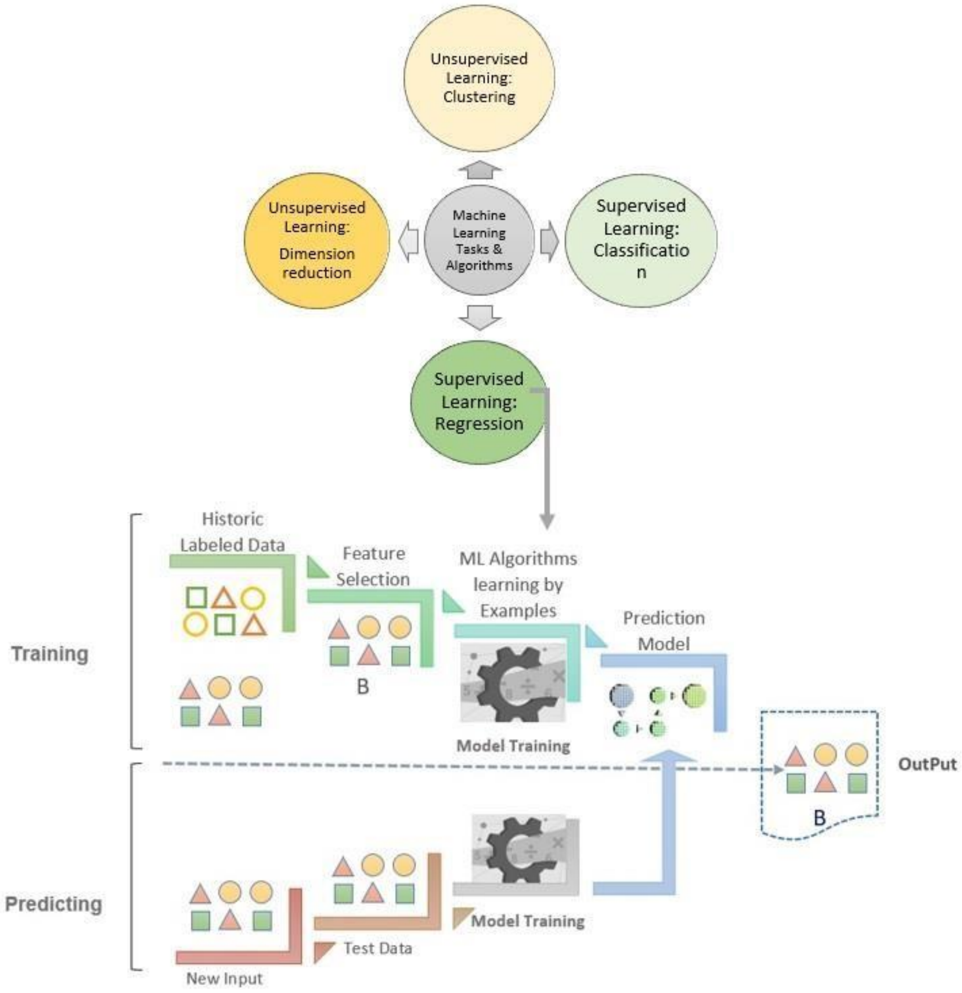
Unsupervised Learning: Clustering
Unsupervised Learning: Dimension reduction
Machine Learning Tasks & Algorithms
Supervised Learning: Classificatio n
Supervised Learning: Regression
Training
Historic Labeled Data
Feature Selection
B
ML Algorithms learning by Examples
Model Training
Prediction Model
OutPut
B
Predicting
New Input
Test Data
Model Training

Figure 2.5. Machine Learning tasks and Supervised Learning process

Models and Algorithms

In this section, we describe some of the machine learning algorithms that we decided to apply for our study after first evaluating our dataset and investigating the result of different algorithms. It is mentioned that this is just a brief description of each algorithm because our study is not an Systematic Literature Review (SLR) on the functionality of each algorithm or its advantages or disadvantages. Therefore, we provide a short explanation based on their ability to give a generic perspective about what algorithms we used in this study based on our study approach or problem statement.

Support Vector Machine (SVM): One supervised machine learning algorithm is the SVM model that can analyze the data for both regression and classification. Although the

SVM is a linear model, it can be used for both non-linear and linear models. This analysis is done by SVM through a technique that is called Kernel technique. Kernel as a mathematical function is one of the SVM hyperparameters that try to find out the most optimal and efficient separating line or boundary by transforming the input dataset into two phases or dimensions [14]. When we can separate the input data into two sections, and they are separable, we utilize a hyperplane line to create two classes, as is shown in Figure 2.6.

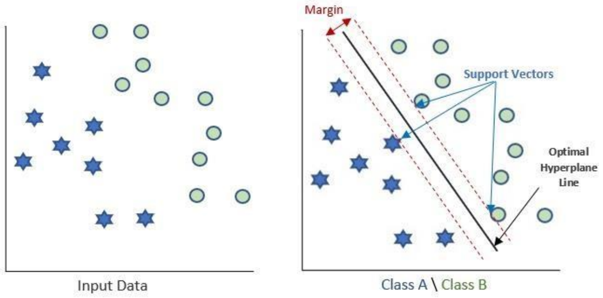

Figure 2.6.The input data is separated by hyperplane line

As is shown in the Figure 2.6, the solid black line is an optimal hyperplane line that the distance between two dotted black lines, and the optimal hyperplane line is called margin. The two dotted lines are two hyperplane lines that move between the nearest and optimal hyperplane lines. The closest data to the hyperplane lines are support vectors, and it can be claimed that often there is no data in the margin area when we use this method. But if the raw data or input data is not separable, the data is divided into two-dimension like illustrated in Figure 2.7.

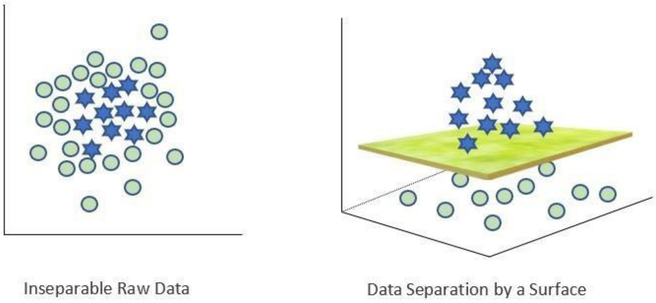

Figure 2.7.Data separation into two dimensions by a decision surface

The Kernel type can be Radial Basis Function (RBF: for non-linear problems), Polynomial Kernel Function, Linear Kernel, Sigmoid, Precomputed, Gaussian Radial Basis Function, and Gaussian Function. If we do not determine a specific type for the Kernel, the default type for the Kernel is considered RBF. The SVM parameters are the Kernel, degree, gamma, coef0, tol, C, epsilon, shrinking, cache_size, verbose, max_iter that can be modified or changed based on our dataset or model function [21].

Random Forest (RF): Another most popular machine learning model and algorithm is RF, a supervised learning and a tree-based algorithm (Figure 2.8). "Random" means this algorithm uses many different decision trees made randomly, and this huge number of trees creates a "Forest" of trees. One decision tree has a high level or amount of variance in the training set. At the same time, the RF uses several decision trees on one sample of the dataset, that the result of all the decision trees is the low level of variance. Indeed, the collection of confluences and the production of the decision trees in each sub-branch improve the algorithm performance. So, the result or output is gained based on the combination of multiple decision trees, not one decision tree.

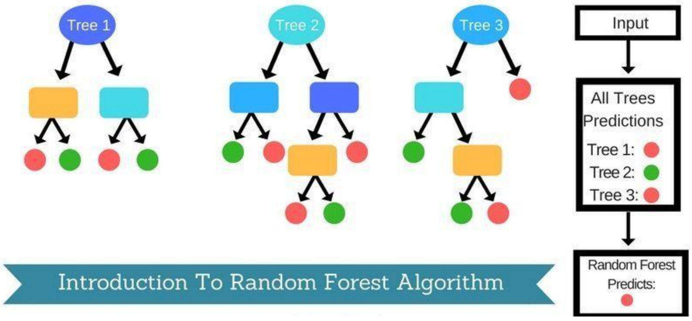

Figure 2.8.The functionality of Random Forest. The final result is the majority of voting (red ball) [22]

The method used by RF is the Bagging technique that includes Bootstrap and Aggregation phases (Figure 2.9). Each tree in the training phase is build based on learning from one sample of data points that are randomly selected. Bootstrap does resample through replacement which means every sample replaces with a random sample selected. Sometimes one sample can be repeated or used many times in the replacement process. RF in the regression model considers the mean of all the outputs as a final result or output that this process is called Aggregation (Figure 2.10). RF in the classification model produces the final output based on the majority vote.

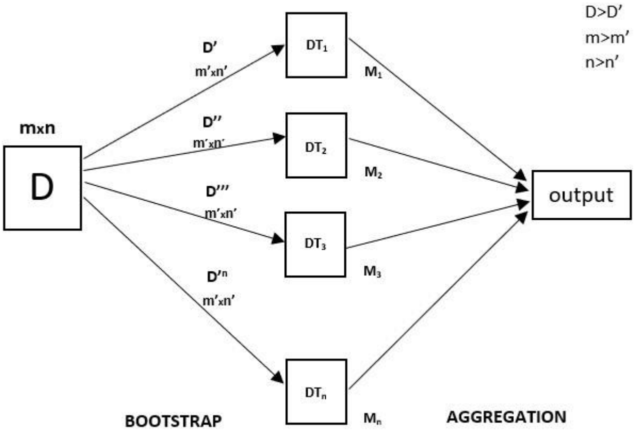

Figure 2.9. Bootstrap and Aggregation in the Random Forest [23]

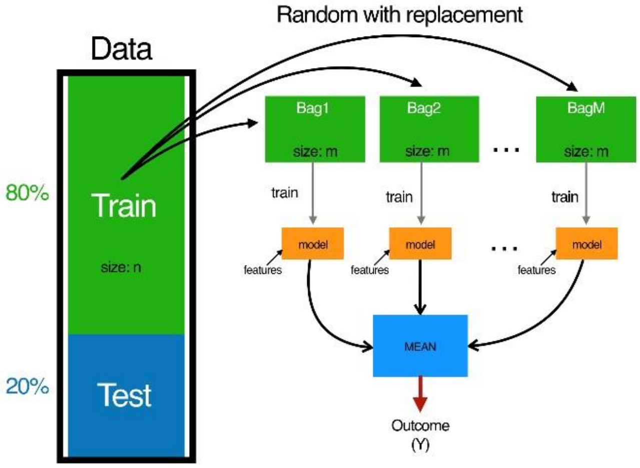

Figure 2.10.The process of Bagging regression model. The Mean means Aggregation [24]

The RF is a fast model in the data training phase because of its great number of decision trees, but it is known as a slow algorithm in prediction when the dataset is trained. Therefore, we should maybe choose other algorithms for run-time performance and real-time prediction. RF is a widely used model for most machine learning approaches. Some algorithms like the neural network algorithm can be better in some features such as better performance compared with the RF algorithm. But the neural network algorithm is time-consuming, while RF, with easy and quick development, is an efficient algorithm for various features like categorical, numerical, and binary, making it a flexible algorithm. Overall, RF is a fast, simple, robust, and diverse algorithm with easy and quick development that we can apply for both regression and classification tasks [25].

XGBoost: When we want to talk about performance and speed for supervised learning tasks, XGBoost (Extreme Gradient Boosting) is another efficient algorithm that is a tree-based algorithm. XGBoost can be used for classification and regression tasks in machine learning challenges when we have a structured dataset with small or medium size. For example, the countless of decision trees causes to overfitting issue and model complexity. XGBoost algorithm can eliminate these problems through Ridge regression and Lasso regression [26]. This algorithm is capable of managing missing values by understanding the missing values' trend. This trend is gained through automatic "learning" from the best missing values in the "training" phase of the XGBoost algorithm. Using the automatic learning ability of XGBoost can also help to fix the problem of raw data sparse.

Furthermore, the XGBoost structure includes a Cross-Validation (CV) function that this ability means we do not need to import the CV function from Scikit-Learn library [27]. XGBoost algorithm follows the ensemble learning [28] method (Figure 2.11) to predict the distance between the predicted values and the actual values. In contrast with machine learning method that uses one hypothesis based on each data training phase (base learners or individual models), the ensemble learning method uses several learners and make a combination of hypothesis to create a sample for more precise prediction or predictive model. The ensemble models include several base learners in which both training and testing phases are performed. In fact, because the base learners work based on a random guess, the XGBoost algorithms extract the poor performance of base learners from a combination of prediction of ensemble learners to gain excellent and precise final prediction.

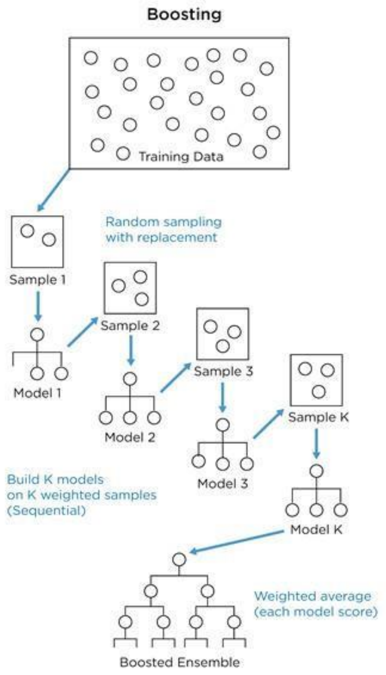
Boosting
Training Data
Random sampling with replacement
Sample 1
Sample 2
Sample 3
Sample K
Model 1
Model 2
Model 3
Model K
Build K models on K weighted samples (Sequential)
Weighted average (each model score)
Boosted Ensemble

Figure 2.11.An ensemble learning method example [29]

AdaBoost Regressor: An ensemble method and Boosting are two essential features of the AdaBoost algorithm. This algorithm uses the ensemble method to grow trees in regular series in the training phase and tries to improve the weak classifications by using the Boosting feature (Figure 2.12). It does this by Boosting the combination of previous weak classifications and trying to set a new strong combination of previous weak classifications into the new classification to alleviate the problems of the previous poor classification in the new sample. Decision trees that grow using the Boosting method and form new classifications are called "stump". In this case, each tree is trained so that it pays particular attention only to the weaknesses and challenges of its previous tree. This model works based on this hypothesis that making a new model from the previous weak models can create a new powerful model that ensemble learning produces sequentially. In the regression problems, the AdaBoost algorithm computes and applies the Mean of these models made by Boosting and ensemble method [30].

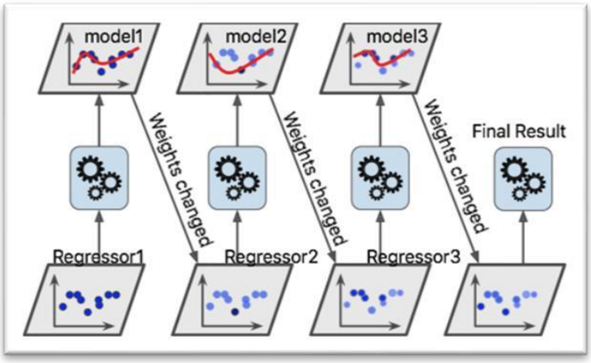

Figure 2.12.The new strong models are made by Boosting and ensemble learning techniques and the average of these models is used for regression as a final result or output [31]

Indeed, the output or final predictor is a combination of all several predictors, including their knowledge about the previous models or predictors. With this approach, each new model is more efficient than the previous model.

Ridge Regressor: The regularized shape of linear regression is Ridge regression, one of the supervised learning algorithms. Ridge is a model tuning algorithm that can analyze every dataset that has a multicollinearity problem. When there is a high correlation between some input variables with other variables in the regression model, the dataset has the multicollinearity problem. This algorithm uses the L2 penalty technique (adding a squared magnitude of the coefficient to the loss function) to shrink some parameters like coefficient for those input variables that do not influence the model prediction [25], [32]. By limiting the size of all coefficients, the L2 penalty method tries to make these ineffective parameters smaller and makes them zero or omitted. Also, it decreases the complexity of the model because of coefficient shrinkage. So, the Ridge algorithm with the L2 penalty method can prevent the multicollinearity problem [17], [33]. This method is useful for feature selection when we have a great number of features in the input dataset because it declines or removes ineffective features.

K- Nearest-Neighbors Regressor (KNN): One of the non-parametric algorithms initiated by Fix et al., 1951 [34] and then developed by Cover et al., 1967 [35] is the KNN algorithm (Figure 2.13) which is used for both classification and regression problems. Based on the performance of this algorithm, every data point gets a value or a weight. When a new data point is entered, the algorithm tries to find out how similar the new data point is to the training dataset points and assign a new value to this new input based on this similarity [36]. The KNN calculates the distance between the data points in the training set and the new input data point that is a new input or observation. This algorithm is sensitive to the scale of the dataset because it works based on distance. Therefore, before using this algorithm, we should consider the scale of our dataset. Because on a larger scale, it calculates the higher distances leading to the poor result. In this algorithm, the K is an integer value and parameter that points to the number of all nearest neighbors in the most of voting process steps.

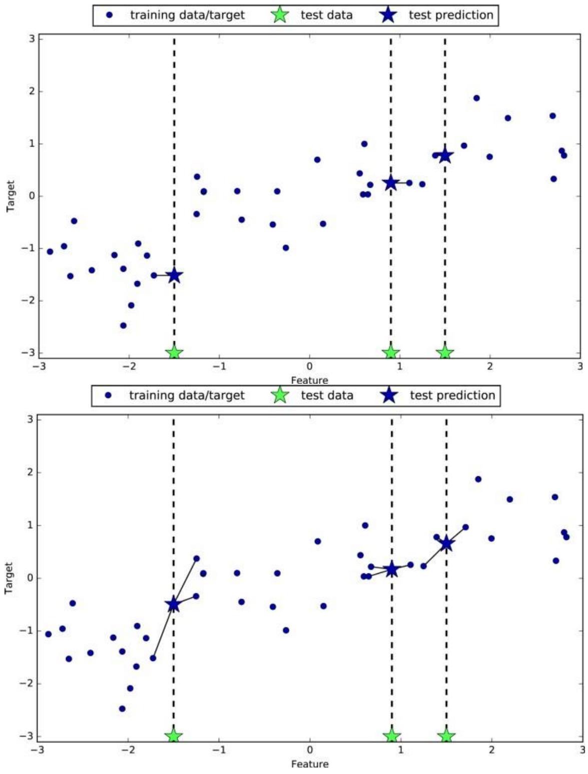

Figure 2.13. KNN regression plot by using n_neighbors = 1 / using more closest neighbor and prediction by computing the Mean of the relevant neighbors [37]

The important thing in the KNN for classification is that it calculates the Mode of nearest K neighbors, while in the regression, it computes the Mean of nearest K neighbors. The KNN can store all the training samples and forecast numerical target values based on distance functions. In fact, in both regression and classification models, KNN works based on the distance functions. The simple functionality of the KNN for regression is to compute the mean of the numerical target values of the KNN. As mentioned above, this algorithm stores all training instances in memory because it does not have any special training phase. This can be a great advantage for this algorithm that can make predictions without using the training phase. But the problem arises when this algorithm is computationally costly if the data is too large. Because this requires a lot of memory space and time to store all the training samples, it is also called a lazy algorithm due to not having a particular training phase and storing all the training samples. The lack of a special training phase and a non-parametric algorithm makes the KNN an efficient algorithm for non-linear datasets [38].

Long Short-Term Memory cells (LSTM): LSTM has been introduced to improve Recurrent Neural Networks (RNN). Therefore, first, we introduce the concept behind RNNs. RNNs are other types of Artificial Neural Network (ANN) in which the neurons have connections to subsequent steps. Figure 2.14 demonstrates a simple RNN layer architecture. Like other ANNs, RNNs can have many hidden layers, or connections can have complex behaviors.

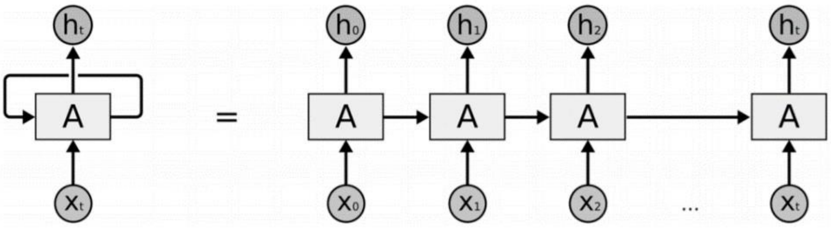

Figure 2.14.The RNN architecture. The figure shows an RNN layer (left) and its unfolded schema (right) [39]

In a standard ANN, the data goes through the input, hidden, and output layers, respectively. While in RNN, the hidden layer receives information from both the current time step input layer and the prior time step hidden layer. In this way, the RNN can keep the past or historical information [36]. Recurrent networks are widely used in sequential data like time-series problems because this kind of network can consider the nonlinearity of sequences, preserve the previous state, and remember past events by connecting past and current neurons. This characteristic makes the RNN models very appropriate for time-series prediction problems [40].

By training RNNs using backpropagation, through time, the vanishing and exploding gradient problems will happen. The exploding case occurs when the gradient factor increases exponentially, making the model unstable because of a large change in the weights. On the other hand, the vanishing case is when the component decreased enormously. The weight coefficients become very small, near-zero in this condition, and the model does not learn anything during the training. For tackling these problems and improving the RNNs, some solutions were introduced; among them, LSTM was a successful approach [41]. The structure of LSTM is depicted in Figure 2.15, and the abbreviations are described in Table 2.2.

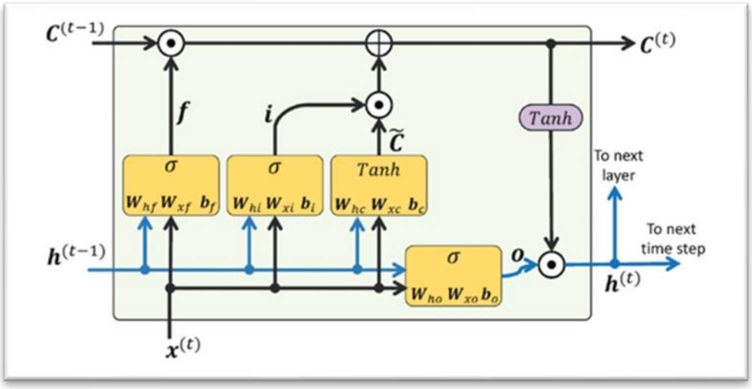

Figure 2.15. The LSTM structure [36]

Table 2.2. The abbreviations' description of LSTM

$\boldsymbol{C}^{(t-1)}$: the cell state from the previous time step (t-1)	$\oplus$: element-wise addition
$\boldsymbol{C}^{(t)}$: the cell state from the current time step (t)	$\odot$: element-wise multiplication
$\boldsymbol{x}^{(t)}$: input data at current time step (t)	σ: sigmoid function
$h^{(t-1)}$: hidden units' activation at previous time step (t-1)	Tanh: hyperbolic tangent function
	W: weight matrix b:
	bias vectors

The LSTM is composed of three computation units called gates:

- The input gate (i) is responsible for allowing the signal to update the “cell state” or not.
- The forget gate (f) makes the cell keep its past state or ignore it.
- The output gate (o) permits the cell state to influence other nodes in the layer or prevent that [42].

Regression Evaluation Metrics

There is the fact that the ability of machine learning models in future prediction should be evaluated by some statistical metrics or measurements. We can use various metrics in the regression models to estimate prediction accuracy. In our regression models, we used some metrics like:

Mean Absolute Error (MAE): measures the errors (differences) between predicted variables and the target, then calculates the absolute value of the average of the total errors of the predicted set [43]. Our study first estimates the MAE for each Device-ID based on time duration changing and choosing minimum MAE. After collecting all minimum MAE of all Device-IDs, we compare them and choose the least minimum MAE to find the best time duration for prediction. The lower the MAE value and the closer to zero, means that our model works better. Also, it is mentioned that we applied another type of MAE called RMAE (Root Mean Absolute Error), which is the value of the root of MAE.

Mean Squared Error (MSE): Another popular regression metric that we use in our machine learning models is MSE that calculates the sum of square differences (error) between predicted values and target variables [44]. In summary, the purpose of training the machine learning model is to reduce the amount of loss function to gain a prediction that is precisely equal to the actual value.

Root Mean Square Error (RMSE): Using RMSE (the root of MSE) helps find and handle the larger errors. Indeed, RMSE indicates how much our regression line is fit with the data points. The lower values of this measurement indicate better fit and higher accuracy for our predictive model [45].

Correlation Coefficient: This indicates how much variables relate to each other or how they relate. This value is a statistical measure and is always between (-1, +1). If the linear correlation is very weak or the variables do not correlate, the Correlation Coefficient becomes (0). When the variables often move in the same direction, the Correlation Coefficient is (+1) because there is a perfect relationship and positive connection between variables, while (-1) shows the variables have a strong negative correlation or negative relationship [46]. This metric describes the dependency between our variables that prove how much of a change in one variable causes a change in another variable.

Variance score: There are three kinds of variance: residual, regression, and total variance. We utilize regression variance to investigate the degree of difference between actual data and our model. The goal of using this metric is to find the value error or difference of actual value from the mean of predicted data points through using the regression line rather than the mean to make the prediction. The Best possible value or score for variance is 1.0 and more than 60%. Lower values are worse and show that the data collected should be investigated or collected again. Perhaps some extra factors should be removed from the predictive model [47].

R-Squared (R^2 or coefficient of determination): R^2 calculates the proportion of variance to a dependent variable that is defined by variables in the regression model or independent variables. R^2 for the multiple regression represents how much the data points are close to the regression line. This statistical measurement describes how the variance of one variable can explain the variance of another variable. The R^2 value is the target variable variation value in the supervised learning that the linear model defines. This value is between 0 and 100%. The zero value means the model does not explain any variability of the target data. The 100% value shows that the model explains all variability of target value around its mean [48].

Hyperparameters Optimize Machine Learning Models

Hyperparameters are anything that is set before the training of the machine learning method begins. They are different from inner parameters. For example, in a neural network model, the weights are not hyperparameters because they are set and updated in the training process. The batch size or optimizer functions are hyperparameters since they are placed before training begins and do not change during the model training phase. Since they control the training algorithm behaviors directly, they are crucial in machine learning studies. Also, they have a fundamental impact on the model performance [49], [50]. Some simple machine learning models do not require any hyperparameters. While in some other algorithms, there are many hyperparameters, some may be dependent on the other ones. The execution time of model training and testing may depend on its hyperparameters configuration [51].

Hyperparameter Tuning (HPT): In machine learning, the process of finding hyperparameter values that have the highest performance concerning the execution time is called hyperparameter tuning (HPT) or optimization. This process is done before the training phase begins. There are a wide variety of hyperparameter iterations and combination options. In this regard, the HPT may be an exhaustive and time-consuming task [49]. Two main HPT methods exist: manual and automatic. Manual search performance depends on the professional knowledge and experience of performers and should be done by expert users. This method cannot be applied when encountering high dimensional data or algorithms with many hyperparameters, and it is not reproducible easily. Automatic search methods are good choices to overcome these drawbacks. Among automatic search methods, Grid Search is a popular method. It is an exhaustive search and trains the machine learning algorithm with every possible value set of defined hyperparameters and provides the best combination with the best performance by evaluating the performance of models according to the predefined metric [50].

ReactJS

ReactJS is an open-source and frontend JavaScript library that is utilized to create a user interface. ReactJS is efficient and worthwhile due to its benefits and attributes. Some of its useful attributes are being declarative, fast, simple, flexible, scalable, building a web application, ability to communicate with old web servers like NGINX or Apache, ability to communicate with the backend like Rails, PHP, and letting you create a reusable and complex user interface from small parts of code (components) [52]. These remarkable traits lead every data scientist researcher to apply this frontend library to visualize JSON's data. Research Questions and Methodology

In this study, the research section includes two parts. The first one comprises our research approach and methodology about the study's machine learning part. The second one is a brief literature review addressing the studies using ReactJs for JSON data visualization. In the first section, we use the new research methodology that is a combination of two methodologies, as we explain in the following.

Machine Learning Research Approaches

To achieve our goal of predicting the amount of water consumption, we shaped our research by investigating many studies about energy consumption in both IoT technology and machine learning techniques. Finally, we decided not to talk about both technologies because this study is not just the Systematic Literature Review. It is unnecessary to focus on all techniques to deal with this issue. So, we continued our study toward concentrating on the machine learning models and algorithms.

2.8.1.1 Research Question *(*RQs)

Research Question 1 What are the characteristics of the dataset used in the energy and water consumption studies?

Research Question 2 Which types of machine learning algorithms or models are efficient for analyzing water datasets and predicting water consumption?

Research Question 3 What are the other possible methods used in addition to Artificial Intelligence (AI) algorithms in energy and water studies?

Research Question 4 Which variables are influencing water consumption?

Research Question 5 What are the evaluation metrics for measuring the performance of models in water consumption studies?

2.8.1.2 Scholarly Sources and Search Strategy

2.8.1.2.1 Data Resources

In this section, the academic resources as are mentioned below were the basis of our research.

- ScienceDirect
- ACM Digital Library
- Springer Link
- IEEE Xplore Digital Library
- Hindawi
- Journal of Algorithms & Computational Technology

2.8.1.2.2 Search Term

After rounds of initial searches with various combinations of search terms, finally, we formulated the following search term, which was an efficient term for searching:

("SENSOR" AND ("BIG DATA" OR "MACHINE LEARNING") AND ("CONSUMPTION" AND "ENERGY" AND ("WATER" OR "ELECTRICITY")) AND ("CITY" OR "MUNICIPALITY"))

2.8.1.2.3 Search Process

Our search process is a combination of two techniques and includes four phases that results from the phases are described in Search Execution:

Phase 1. First, we reviewed the abstract, introduction, and summary of the related articles to our study. Then, we separated those papers that were more relevant to the subject matter studied. We finally transferred them into a reference manager known as Zotero.

Phase 2. Then for scrutiny review, we scanned all the resources obtained from the first phase accessible to explore the studies' details further. Due to a more precise investigation in this step, we reviewed a few resources related to our field of research.

Phase 3. Then we reviewed the remaining resources from previous phases based on the Systematic Literature Review (SLR) methodology to review and to perform our results. The results of this phase have been categorized in a data extraction form generated in an Excel file for streamlined accessibility.

Phase 4. In the final evaluation, we finalized our review by the combination of two techniques. To get closer to the studies that were precisely relevant to our research topic, we utilized the results of the SLR for doing Snowballing. As a result, we achieved exactly related studies in this area by searching for a few references from the previous phase.

2.8.1.3 Criteria as a Selection Tool

This step presents our criteria for selecting and choosing resources and categorising them into two sections: Inclusion and Exclusion Criteria (Table 2.3).

Inclusion Criteria	□ Exclusion Criteria

The studies which investigated the big data management obtained from sensor networks The papers which referred to at least one machine learning algorithms Studies related to sustainable smart cities Focus on energy consumption, especially water consumption	□ The studies before 2009 □ Papers in a language other than English □ Thesis, reports, books □ The studies that are not relevant to our research like investigation security and Sensors' function □ The studies that are not defined as reliable (such as web pages) □ The inaccessible studies

Table 2.3. Inclusion and Exclusion criteria for Machine Learning studies

2.8.1.4 Research Methodology (SLR + Snowballing)

2.8.1.4.1 Search Execution

As Figure 2.16 shows, all the results have been achieved through the combined methods include the SLR and Snowballing technique on academic resources that we describe in the continuation of this section.

The important and time-consuming part of search execution was the 2nd step results (Full-text Scanning for Literature Review) that include choosing one technology between machine learning and IoT technologies. After investigating some IoT scientific papers, we decided to focus on machine learning methods in the SLR technique. Therefore, the number of results decreased because of removing IoT studies. The Snowballing technique helped us utilize the references of the most relevant studies to find other related studies in this area based on our research criteria. After further review by Snowballing method ability, 15 related sources were added to our resources to get closer to the subject under study (Table 2.4). The results of these 27 scientific papers focusing on the hourly water consumption prediction are fully described in chapter 3.

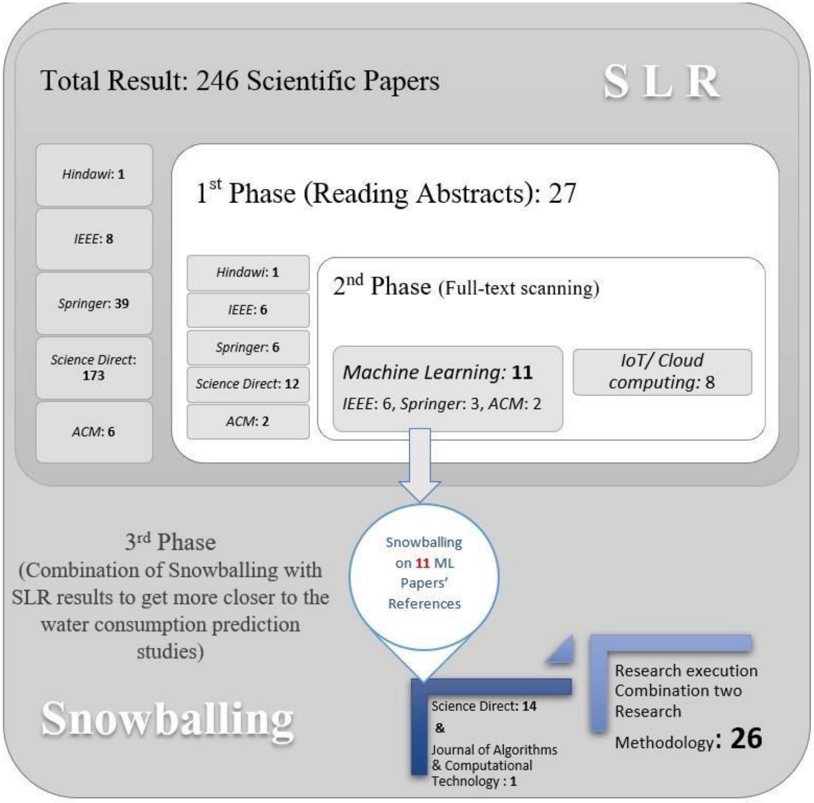
Total Result: 246 Scientific Papers
SLR
Hindawi: 1
IEEE: 8
Springer: 39
Science Direct: 173
ACM: 6
1st Phase (Reading Abstracts): 27
Hindawi: 1
IEEE: 6
Springer: 6
Science Direct: 12
ACM: 2
2nd Phase (Full-text scanning)
Machine Learning: 11
IEEE: 6, Springer: 3, ACM: 2
IoT/ Cloud computing: 8
3rd Phase
(Combination of Snowballing with SLR results to get more closer to the water consumption prediction studies)
Snowballing on 11 ML Papers' References
Snowballing
Science Direct: 14
&
Journal of Algorithms & Computational Technology : 1
Research execution Combination two Research
Methodology: 26

Figure 2.16. Our research methodology is a combination of SLR and Snowballing

Library	Full-Search Result	Abstract and Title Scanning	Full-text Scanning		Result of Snowballing on Machine Learning Papers' References
			Machine Learning Literature Results	IoT/ Cloud Computing Literature Results	
Hindawi	1	1	0	1	0
IEEE	8	6	6	0	0
Springer	39	6	3	3	0
Science Direct	173	12	0	4	14
ACM	6	2	2	0	0
Journal of Algorithms & Computational Technology	0	0	0	0	1
246		27	**11** ⇩	8	**15** ⇩
Research Execution			**26**		

Table 2.4. The result of Research Execution

Second Section: The ReactJs Research Approaches

2.8.2.1 Data Flow Display by the ReactJs

Facebook has developed ReactJs in JavaScript. That is a frontend web application and JavaScript library used as a graphical interface to display data. It has reusable components, which mean it can accept different arbitrary inputs and then show a React component as an output on the screen. Scalable framework, reusable UI components, stable code with regular updates are just some of the functional characteristics of ReactJs that make it an efficient interactive web app for users.

2.8.2.2 Scholarly Sources and Search Strategy

2.8.2.2.1 Data Resources

The results were collected from well-known academic research sources such as:

- ScienceDirect
- ACM Digital Library
- Springer Link
- IEEE Xplore Digital Library

2.8.2.2.2 Search Term

After trying different search terms, we reached desired results by this search terms about using the ReactJs functionality in data visualization.

("ReactJs "AND" DATA VISUALIZATION" AND "SENSOR" AND "MACHINE LEARNING" AND ("TIME-SERIES DATA" OR "JSON") AND "ENERGY" AND ("WATER" OR "ELECTRICITY") AND" CONSUMPTION")

2.8.2.2.3 Search Process

Phase 1. Among 41 studies found, we tried to select the papers relevant to our study's aims with a brief overview. Then we transferred articles with the relevant topics, abstracts, or introduction to our research to the Zotero.

Phase 2. We scanned all the relevant studies from the previous phase that we accessed to examine the obtained resources. Therefore, we reviewed a few numbers of studies to get closer to useful information and data.

Phase 3. The extracted data from relevant studies were transferred to the Excel sheets for quick access.

2.8.2.2.4 Criteria as a Selection Tool

Table 2.5 outlines our criteria for selecting and choosing resources related to the ReactJs studies.

Inclusion Criteria	□ Exclusion Criteria

The studies which investigated the ReactJs functionality The papers which referred to the sensor networks in smart cities Focus on JSON data visualization	□ The studies before 2009 □ Discard papers in a language other than English □ Thesis, reports, books □ The studies that are not defined as reliable (such as web pages) □ The inaccessible studies

Table 2.5. Inclusion and Exclusion criteria for ReactJs studies

2.8.2.3 Research Methodology (SLR)

2.8.2.3.1 Search Execution

To prove the ReactJS capabilities, we applied the SLR methodology to find papers with a similar context to our studies. Therefore, we achieved several efficient and persuasive studies to use ReactJS to visualize the JSON data we provide in section 3.2.

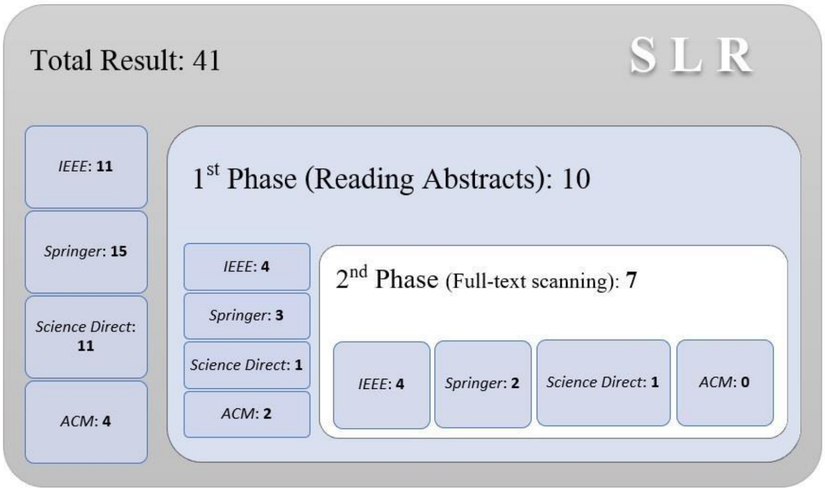

Figure 2.17. SLR methodology for ReactJs studies