Machine Learning Algorithms to Classify Water Levels for Smart Irrigation Systems

Tahani M Allam, Shrouk Ezzeldin Ali*, Elsayed A. Sallam

Computer and Control Engineering Department, Faculty of Engineering, Tanta University, Tanta, Egypt
Shrouk.ezz1992@yahoo.com

Abstract- **Agriculture is the main source of food. With the passing of time, there are dangers in order to preserve on the freshwater in agriculture sector. Thus, one of solutions to save the freshwater is enhancing the wastewater. Machine learning (ML) algorithms are used in several applications, such as smart irrigation, to reduce freshwater loss via building high-performance ML algorithms. This paper proposes four algorithms: support vector machine (SVM), decision tree (DT), SVM with Adaboost, and DT with Adaboost to classify water levels of sprinklers for smart irrigation. Here, five levels of water are classified– Max, High, Medium, Low, and Stop. The proposed algorithms are tested to obtain which algorithm achieves better performance and higher accuracy. Five steps sequentially are implemented on the used dataset via Pandas and Scikit-learn frameworks. The steps are preprocessing data, feature selection, feature scaling, training, and classification; to analyze the performance of the algorithms. The results showed that the DT algorithm with Adaboost is the best algorithm compared to the rest of the algorithms. The DT algorithm achieves an accuracy score of 0.912 with a shorter testing time of 2.2 seconds and mean square error (MSE) of 0.08.**

Keywords: Multiple algorithms; Smart Irrigation; Machine learning; Freshwater; SVM; DT; Adaboost

I. INTRODUCTION

Machine learning (ML) depends on computational statistics, the main idea of ML is making predictions using computers. Machine learning algorithms create a mathematical predictive model that depends on a sample of data, known as "training dataset" [1]. Also, predictions or decisions made without explicit programming is another benefit of machine learning. Machine learning algorithms are used in many different applications, such as intelligent irrigation [2], healthcare [3], speech recognition [4], smart manufacturing [5], and human activity recognition [6]. The problem of over irrigation occurs due to poor distribution or lack of water management, while the under irrigation provides sufficient water to the plant [7]. Thus, this problem leads to poor crops. To save the freshwater with high quality of crops, a smart irrigation system is developed to classify the level of the water of the irrigation with the help of ML algorithms.

In the literature, a number of machine learning algorithms have been introduced in smart irrigation with ML. Researchers [8] introduced SVM and random forest (RF) algorithms to decide the amount of the irrigation required by crops. They evaluate the RF and SVM algorithms compared to previous algorithms, reached an accuracy of 81.6% for RF. In [9], authors proposed support vector regression (SVR) and k-means clustering ML algorithms to forecast the soil moisture. The algorithms are applied on online data using multiple sensors. These algorithms decide whether there will be irrigation or non-irrigation. S. Ramya and Swetha [10] presented SVR and bagging ML algorithms with internet of things (IoT) to develop a smart irrigation system. These algorithms help in effective decision making for smart irrigation. Also, the algorithms depended on data acquisition with online weather data collection. In [11], researchers implemented a sustainable irrigation system based on a random forest (RF) algorithm. They evaluated the performance of the algorithm via a confusion matrix, and achieved an accuracy of 84.6%. Authors [12] proposed a DT algorithm based on a fully automated system which fills a tank of the water for agriculture irrigation. This system includes a wireless network and a temperature sensor with a soil moisture sensor positioned on the plant. In [13], H. Chen et al. implemented a convolutional neural network (CNN) deep learning algorithm with a DT to evaluate and classify levels of water pollution with an analysis of chemical oxygen. Researchers [14] presented k-nearest neighbor (KNN) and SVM algorithms to predict water quality status. The KNN algorithm used 10-fold cross validation method to achieve a maximum accuracy and provides highest-f1-score compared to the SVM.

Looking at the reviewed literature, we found new ML algorithms that can be applicable to different datasets. In this paper, four ML algorithms are proposed to classify water levels for irrigation systems, the algorithms are support vector machine (SVM), decision tree (DT), SVM with Adaboost, and DT with Adaboost. The levels of the water are "Max, High, Medium, Low, and Stop". The algorithms are applied to the dataset acquired from the Climate Toolbox [13]. The DT with Adaboost achieve the best accuracy with high performance among the other algorithms, it reached a maximum accuracy of 0.912. This result is achieved via the fine-tuning of the hyper parameters of the proposed algorithms. These algorithms are evaluated using assessing metrics of accuracy score and mean square error (MSE). The main contribution of the paper is achieving high accuracy to classify different water levels of a smart irrigation system.

The rest of the paper is structured as follows: Section II explains the proposed machine learning algorithms. Section III presents the experimental results. Section IV demonstrates the discussion of the results. Finally, conclusion of the paper is listed in Section V.

II. PROPOSED ALGORITHMS

This work has three main parts: (i) Data pre-processing with normalization such as rescaling (ii) training of four machine learning algorithms with fine-tuned parameters (iii) Evaluating the performance of each algorithm and classifying water irrigation levels. Figure 1 shows the suggested algorithms' framework. These algorithms are implemented via the Scikit-learn library. The methodology includes five steps: the first step is the pre-processing by cleaning the missing values using the dropna function, while the split function is used to split the dataset, and data visualization to discover the relations between the features. The second step is the rescaling on the dataset to improve the performance of the algorithms. For instance, obtaining normalized data, which falls in the range between 0 and 1, see Equation (1) [14]. The third step is the training of the ML algorithms on the used dataset. The fourth step is testing the performance of the proposed algorithms through learning curves. The fifth step is the classification of the water irrigation levels. The classification of the algorithms whether maximum level or high level, medium level, low level, and stop. A regularization technique with a Gridsearch is used to optimize the hyper-parameters of the algorithms for preventing the over-fitting occurrence. The proposed algorithms are implemented with Python programming language using Jupter Notebook environment.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

A. Support Vector Machine (SVM)

The SVM algorithm splits the dataset input variable space into a number of classes in order to identify the greatest marginal hyperplane, which is a line.to split the input variable space. The hyperplane could be generated in an iterative approach through the SVM algorithm that can reduce errors [15-17]. Equations (2) and (3) represent the mathematical relations of the hyperplane distance $d_H\,(x_n)$, where w is the weight from the SVM algorithm, while b represents the bias of the algorithm. SVM has several advantages; it is relatively memory efficient and it performs well when number of samples less than number of dimensions.

$$w* = arg_w max\,[min_n d_H(\phi(x_n))] \qquad (2)$$

$$d_H(\phi(x_0)) = \frac{w^T(\phi(x_0)) + b}{w^2} \qquad (3)$$

B. Decision Tree (DT)

The DT algorithm is used for both classification and regression problems. In most cases, it is used in classification issues. The DT algorithm is a tree-established classifier, in which inner nodes constitute the features of a dataset, with branches that constitute the choice guidelines and every leaf node represents the outcome [18,19]. The DT algorithm is based on the entropy $H(S)$ and the information gain (I_G), which are calculated using Equations (4) and (5). Regarding the DT, the features are selected from the dataset via calculating entropy and information gain. one of the benefits of this algorithm needs less effort during preprocessing of the data and doesn't require scaling or normalization. Also, Gini is a method used to split the DT algorithm. Gini is the probability of correctly labelling a randomly selected element if it was labelled randomly based on the distribution of labels in the node. Equation (6) describes the *Gini*.

$$H(S) = \sum_{i=1}^{c} - p_i \, log_2 \; p_i \qquad (4)$$

$$I_G = \sum_{i=1}^{c} log_2 \; \frac{1}{p_i} \qquad (5)$$

$$Gini = \sum_{i=1}^{n} p_i 2 \qquad (6)$$

C. Adaboost

It is an ensemble method usually produces more accurate solutions, which could use for several applications. It produces improved results, the basic concept of the ensemble methods is to train more than one algorithm and combine their results into a single result to obtain reasonable and appropriate performance [20]. The Adaboost function is described in Equation (7). There are several advantages for the gradient boosting: thus, it handles the dataset that are missed and also this algorithm doesn't require a pre-processing for the data. In addition, it is more flexible, its optimization can be performed on various loss functions, and it has many hyper-parameters for tuning.

$$F(x) = \sum_{i=1}^{i} \alpha_i h_i(x) \qquad (7)$$

D. Dataset Description

The used dataset is available [13] to classifiy the water levels of irrigation. It comprises 743 samples with different water irrigation levels "Max, High, Medium, Low, and Stop". The dataset is gathered for coordinates: 30.8125 North and 31.0208 East for a position inside the University of Tanta. The dataset has seven features which are soil (Soil Moisture), t_{min} (Minimum Temperature), t_{max} (Maximum Temperature), ppt (Precipitation), Ws (Wind Speed), aet (Actual Evapotranspiration), and pdsi (Palmer Drought Severity Index). The dataset is splitted intoan 80% training set and a testing set of 20%. The testing set is utilized to evaluate the performance of the proposed algorithms.

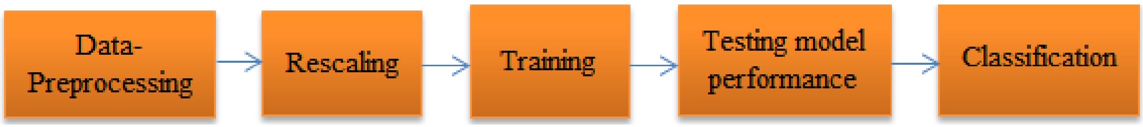

Figure 1. Framework of the proposed algorithms.

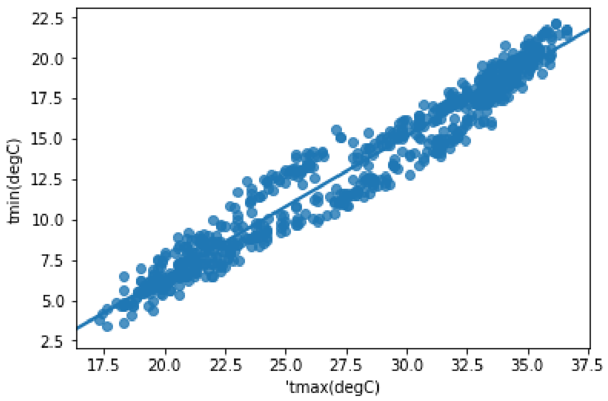

Figure 2. The relations between the t_{max} and t_{min} features of dataset.

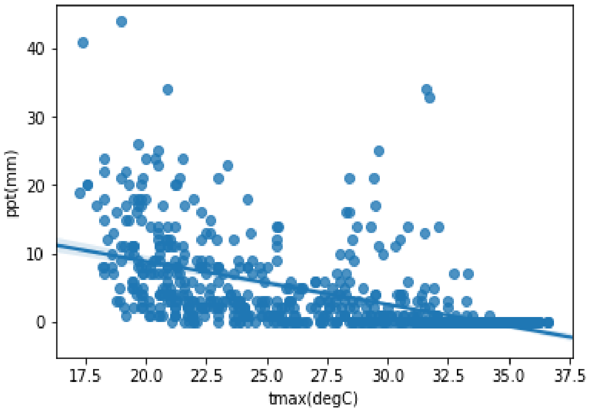

Figure 3. The relations between the ppt and t_{max} features of dataset.

III. Results

Figure 2 and 3 demonstrate the relations between the features of dataset. In figure 2, there is a symmetrical relationship between the t_{min} and t_{max}, while in figure 3, an inversely proportional relationship among the ppt and t_{max}.

Table 1 and figure 4 illustrate the testing accuracy and mean square error (MSE) of the proposed algorithms. The results show that the best algorithm in terms of the performance is the DT with Adaboost. It achieved an accuracy score of 0.912 and MSE 0.08. From Equation (8), the accuracy is calculated. The MSE is determined form Equation (9). The least accuracy is 0.879 for the SVM, while the least MSE is 0.1208. Also, the testing time of the DT with Adaboost is 0.18 seconds.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (8)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - e_i)^2 \quad (9)$$

Table 1. The testing accuracy and mean square error for the proposed algorithms

Algorithm	Accuracy	MSE	Time (s)
SVM	0.879	0.1208	2.89
DT	0.906	0.093	2.8
SVM with Adaboost	0.893	0.103	1.9
DT with Adaboost	0.912	0.08	0.18

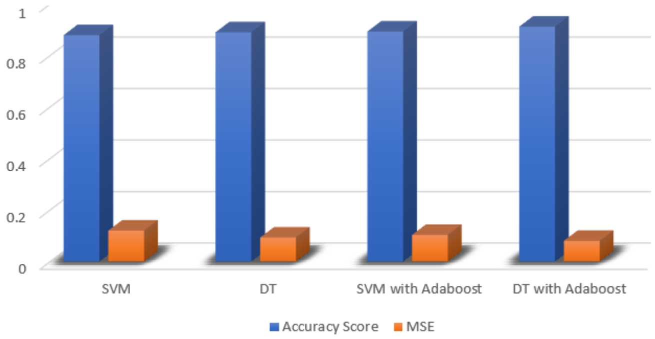

Figure 4. The testing accuracy and mean square error for the proposed algorithms

Leaning curves as result of a *k* cross validation method, is used to diagnose problems with learning such as over-fitting, under-fitting or well-fit model through particular graphs. The process of the learning is divided into training and cross validation. The dataset is divided into *k* partitions, where *k* is selected to 10 with ten iterations of training. One partition for testing and while nine partitions for training. The second iteration is wrapped with the next partition as testing data while the remaining *k*-1 is utilized as training data and so on. Figure 5 demonstrates the learning curves of the DT algorithm with setting the max depth of 3. It is clear that the score of the training curve gradually decreases with increasing the number of training examples, thus the error of the training is increased, while the cross-validation score doesn't increases than 0.6. So, there is an under-fitting. Also, the learning curves of the DT with max depth of 5 are shown in Figure 6.

Moreover, by increasing the max-depth hyperparameter, the performance of the DT algorithm is outperformed and became more accurate after applying the cross-validation method. So, there is no over-fitting or under-fitting. Further, Figure 7 illustrates the DT's learning curves with 8 max-depth. The training score is slightly decreased, while the validation score is gradually increased. Thereby, with increasing the training examples, the gap between the training score and validation score is decreased. So, this causes a well-fit. The Gridsearch technique is applied to the DT algorithm to select the optimum max-depth hyperparameter. This technique obtains that the best max-depth is 9.

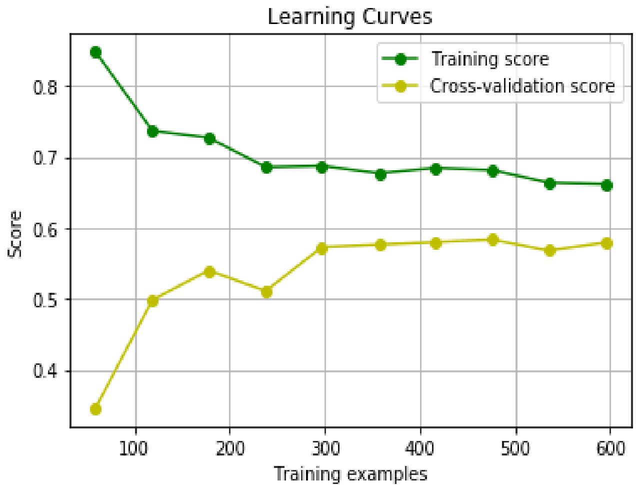

Figure 5. Learning curves for DT algorithm with max depth of 3.

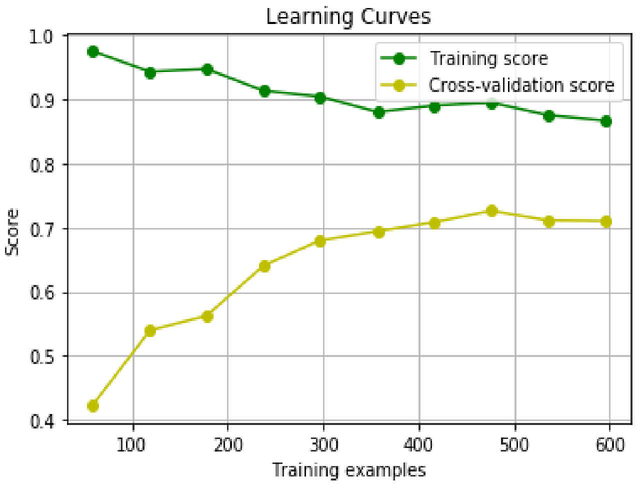

Figure 6. Learning curves for DT algorithm with max depth of 5.

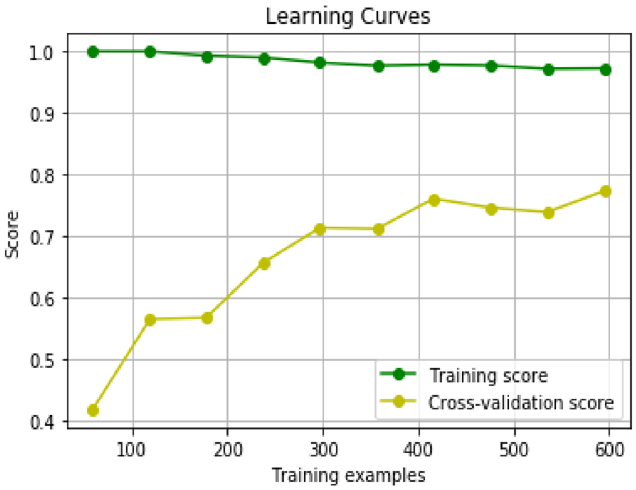

Figure 7. Learning curves for DT algorithm with max depth of 8.

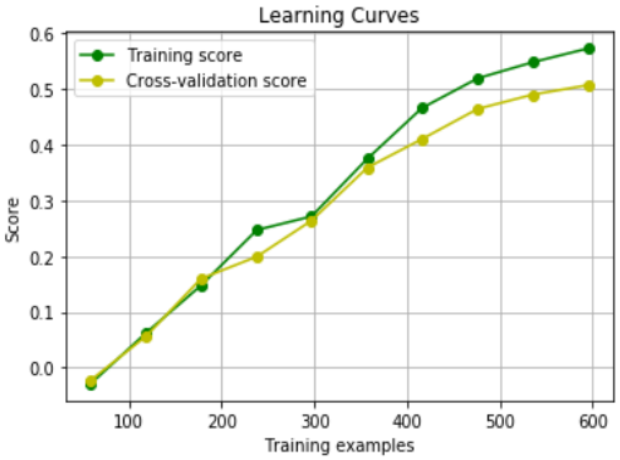

Figure 8. Learning curves for SVM algorithm with C of 1

Figures 8 and 9 demonstrate the learning curves for the SVM algorithm. In figure 8, the score of training reached to 0.58 with 600 examples of training, while the cross validation score reached to 0.5. The regularization parameter 'C' is chosen to be 1. Also, the learning curves of the SVM are shown in Figure 9 with C of 2. It is clear that the training score is increased to 0.7, while the cross-validation score slightly increased.

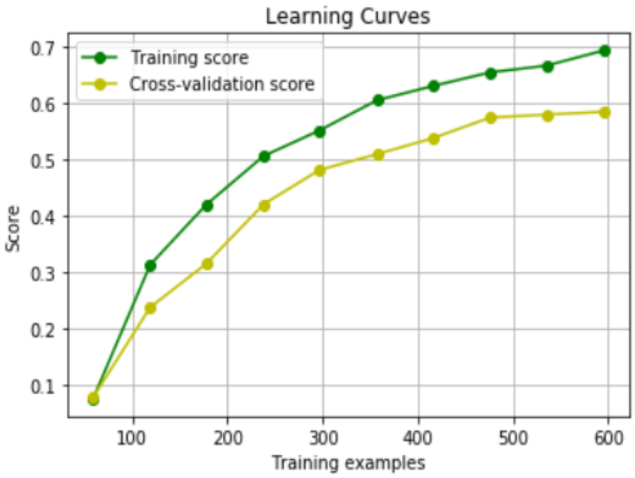

Figure 9. Learning curves for SVM algorithm with C of 2.

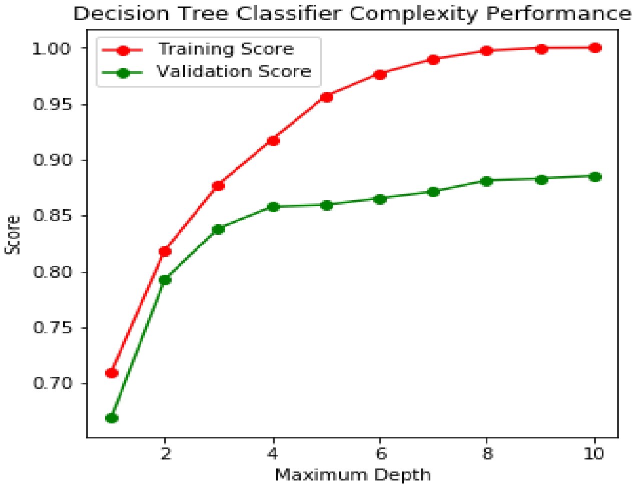

Figure 10. Complexity curves for DT algorithm.

Figure 10 shows the complexity curves for DT algorithm. These curves as result of the training and validation process of the DT with different maximum depths In Figure 10, two complexity curves: One for training and the other for validation. The complexity curves are similar to the learning curves, and the algorithm is scored on both the training and validation sets using the performance metric function.

IV. Discussion

The results show that the introduced algorithms achieved high accuracy and the less error. The decision tree algorithm with Adaboost reached to an accuracy of 0.912 based on the max-depth hyperparameter. When the max-depth is configured to 8, the DT achieved accuracy of 0.899, when the max-depth is set to 5, the testing accuracy was 0.892, and the accuracy was 0.852 with max-depth of 3. The SVM with Adaboost algorithm achieved an accuracy of 0.893 at C of 1.0 with degree = 3. When the testing accuracy was 0.892 at C=2.0 with degree of 2. Further, the mean square errors of the algorithms are 0.1208, 0.093, 0.103, and 0.08 for the SVM, DT, SVM with Adaboost, and DT with Adaboost, respectively. The testing times of the algorithms are 2.89 s, 2.8 s, 1.9 s, and 0.18 s for the SVM, DT, SVM with Adaboost, and DT with Adaboost, respectively. The DT with Adaboost achieved highest accuracy due to it includes the advantages of the DT and Adaboost. The proposed algorithms identify the testing dataset to five classes of the water irrigation levels. Furthermore, these ML algorithms are considered as high-performance benchmark in smart irrigation to classify levels of water. Finally, Table 2 shows a comparison in terms the accuracy of the proposed algorithms and previous works.

Table 2. Comparison among the proposed algorithms and previous works

Algorithm	Accuracy
[8]	0.816
[11]	0.846
SVM	0.879
DT	0.906
SVM with Adaboost	0.893
DT with Adaboost	**0.912**

V. Conclusion

This paper presented an implementation of four machine learning algorithms for classification five levels of the water. The implemented algorithms are DT, SVM, DT with Adaboost, and SVM with Adaboost. The algorithms are applied to a dataset available on the Climate Toolbox. The used dataset is based on seven features: soil moisture, minimum temperature, maximum temperature, precipitation, wind speed, actual evapotranspiration, and palmer drought severity index. Five phases are executed i.e. preprocessing data, one hot encoding, rescaling data, training, and testing. The DT algorithm with Adaboost achieved the best accuracy with 91.2%, the best MSE is 8%, and the training time is 0.18 seconds. These results are obtained with the optimum fine-tuning of max-depth, which is 9 with the best complexity rate, compared to the SVM with Adaboost algorithm which achieved 89.3% with MSE 10.3%, and training time 1.9 seconds. In the future work, a number of deep learning (DL) algorithms can be applied to the used dataset. Also, one could use a huge dataset contains more classes of water levels.

Funding:

This research has not received any type of funding.

Conflicts of Interest:

The authors declare that there is no conflict of interest.

References

[1] B. Genc, and H. Ü. S. E. Y. İ. N. Tunc, "Optimal training and test sets design for machine learning," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 2, pp. 1534-1545, 2019.

[2] H. Nandanwar, A. Chauhan, D. Pahl, and H. Meena, "A survey of application of ML and data mining techniques for smart irrigation system," *In 2020 Second International Conference on Inventive Research in Computing Applications*, pp. 205-212, July 2020.

[3] R. Bhardwaj, A. R. Nambiar, and D. Dutta, "A study of machine learning in healthcare," *In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, pp. 236-241, July 2017.

[4] D. Yu and L. Deng, "Automatic speech recognition: A deep learning approach," *Springer Publishing Company*, Incorporated, 2014.

[5] S. Mohsen, A. Elkaseer, and S. G. Scholz, "Industry 4.0-oriented deep learning models for human activity recognition," *IEEE Access*, vol. 9, pp. 150508-150521, November 2021.

[6] S. Mohsen, A. Elkaseer, and S. G. Scholz, "Human activity recognition using k-nearest neighbor machine learning algorithm," *Proceedings of the 8th International Conference on Sustainable Design and Manufacturing (KES-SDM)*, Split, Croatia, pp. 304-313, September 2021.

[7] L. S. Pereira, T. Oweis, and A. Zairi, "Irrigation management under water scarcity," *Agricultural water management*, vol. 57, no. 3, pp. 175-206, 2002.

[8] A. Vij, S. Vijendra, A. Jain, S. Bajaj, A. Bassi, and A. Sharma, "IoT and machine learning approaches for automation of farm irrigation system," *Procedia Computer Science*, vol. 167, pp. 1250-1257, 2020.

[9] A. Goap, D. Sharma, A.K. Shukla, and C. R. Krishna, "An IoT based smart irrigation management system using machine learning and open source technologies," *Computers and Electronics in Agriculture*, vol. 155, pp. 41-49, 2018.

[10] S. Ramya, A. M. Swetha, and M. Doraipandian, "IoT framework for smart irrigation using machine learning technique," *Journal of Computer Science*, vol. 16, no. 3, pp. 355-363, 2020.

[11] A. Glória, J. Cardoso, and P. Sebastião, "Sustainable irrigation system for farming supported by machine learning and real-time sensor data," *Sensors*, vol. 21, no. 9, pp. 3079, 2021.

[12] M. P. Ebin, R. Kavitha Nair, and J. K Mathew, "Automated irrigation system based on machine learning concept," *International Journal of Information Systems and Computer Sciences*, 2019.

[13] H. Chen, A. Chen, L. Xu, H. Xie, H. Qiao, Q. Lin, and K. Cai, "A deep learning CNN architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources," *Agricultural Water Management*, vol. 240, pp. 106303, 2020.

[14] A. Danades, D. Pratama, D. Anggraini, and D. Anggriani, "Comparison of accuracy level K-Nearest Neighbor algorithm and Support Vector Machine algorithm in classification water quality status," *2016 6th International Conference on System Engineering and Technology (ICSET)*, 2016, pp. 137-141, 2016.

[15] "Dataset" Climate Tool Box, https://climatetoolbox.org/.

[16] S. Akshay and T. K. Ramesh, "Efficient machine learning algorithm for smart irrigation," *In 2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 867-870, July 2020.

[17] A. S. Galathiya, A. P. Ganatra, and C. K. Bhensdadia, "Improved decision tree induction algorithm with feature selection, cross validation, model complexity and reduced error pruning," *International Journal of Computer Science and Information Technologies*, vol. 3, no. 2, pp. 3427-3431, 2012.

[18] A. Goldstein, L. Fink, A. Meitin, S. Bohadana, O. Lutenberg, and G. Ravid, "Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge," *Precision agriculture*, vol. 19, no. 3, pp. 421-444, 2018.

[19] J. Zubek and D. M. Plewczynski, "Complexity curve: a graphical measure of data complexity and classifier performance," *PeerJ. Computer Science*, 2, e76, 2016.

[20] A. Raghuvanshi, U. K. Singh, G. S. Sajja, H. Pallathadka, E. Asenso, M. Kamal, A. Singh, and K. Phasinam, "Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming," *Journal of Food Quality*, 2022.

[21] C. K. Albuquerque, S. Polimante, A. Torre-Neto, and R. C. Prati, "Water spray detection for smart irrigation systems with Mask R-CNN and UAV footage," *In 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, pp. 236-240, November 2020.

[22] E. A. Abioye, O. Hensel, T. J. Esau, O. Elijah, M. S. Z. Abidin, A. S. Ayobami, O. Yerima, and A. Nasirahmadi, "Precision irrigation management using machine learning and digital farming solutions," *AgriEngineering*, vol. 4, no. 1, pp. 70-103, 2022.