Researchers Journal of Science and Technology (REJOST)
https://rejost.com.ng

Development of an Optimized Predictive Machine Learning Approach for Smart Irrigation System using Cuckoo Search Algorithm

S. Tukur, E.M. Eronu, S. U. Hussein

Abstract

The increasing global demand for water conservation has necessitated the development of intelligent irrigation systems. However, traditional irrigation systems often lack efficiency, leading to significant water waste. This study proposes an optimized predictive machine learning approach for smart irrigation system using Cuckoo Search Algorithm (CSA) that can be used to optimized irrigation scheduling and reduce water usage while maintaining crop yield. The system involves data collection, pre-processing, model training and optimization, WaterNet dataset is used with parameters like temperature, pH, turbidity and coliforms. The CSA is employed to optimize the hyperparameters of three machine learning model: Random Forest (RF) 99.07%, Support Vector Machine (SVM) 98.36% and K-Nearest Neighbors (KNN) 89.6%. The Optimized Random Forest (ORF) achieves exceptional performance, significantly improving prediction accuracy. CSA optimization was performed using different population sizes specifically n = 5, n = 10 and n = 20, larger populations yielded better behavior but required more iterations for convergence. In terms of convergence curve, CSA has fast convergence with fewer iterations (16) while PSO has slower convergence and more iterations (25 to 49), the difference is the time they took to converge, additionally, CSA-RF outperforms multilayer perceptron (MLP), due to hyperparameter tuning. Comparison between actual values (Ytest) with predictions from two models; Random Forest (Ypred.RF) shows deviations and misclassifications from actual value while optimized model (Ypred.optimized) predictions matched the actual values. The comparative analysis of classification methods highlights the superiority of the proposed ORF, which was fine-tuned using the CSA. The performance metrics which are accuracy, precision, recall, and F1-score achieved an exceptional score, demonstrating the robustness of the proposed approach, the ORF not only achieved excellent performance levels but also incorporated hyperparameter tuning, ensuring improved stability and reduced risk of overfitting. This study contributes to the development of efficient and sustainable smart irrigation systems, supporting precision agriculture and water conservation efforts.

✉ Tukur Suleiman: tukursuleiman77@gmail.com

Department of Electrical/Electronic Engineering, University of Abuja, Nigeria

Keywords: Smart irrigation System, Water Quality Prediction, Machine Learning, Cuckoo Search Algorithm, Hyperparameter optimization, Precision Agriculture

Article history

Received: January 14, 2026
Received in revised form: March 17, 2026
Accepted for publication: March 21, 2026
Available online: March 30, 2026

1.0 Introduction

In recent years, stress on the natural resources is increasing due to rapid industrialization and population growth and their conservation is one of the major challenges for mankind. Groundwater is a most vital resource for millions of people for both drinking and irrigation uses (Abbasnia *et al., 2018*). Agriculture faces multiple challenges in the 21st century: it has to produce more food and fibre to feed a growing population with a smaller rural labour force, contribute to development in agriculture-dependent developing countries, adopt more efficient and sustainable production methods and adapt to climate change. The most recent estimates indicate that global hunger increased in 2016 and the number of undernourished people in the world increased to an estimated 815 million, up from 777 million in 2015 (Fanadzo & Ncube, 2018).

Highly growing population and accelerated industrial development are causing anthropogenic pollution to both surface and groundwater on one side and geogenic contamination like arsenic, fluoride, high dissolved solids, sodicity, and iron in groundwater on other side. As a result, ensuring safe water quality for the irrigation has become a major challenge to both the central and state governments (Singh, 2018). Worldwide, by 2050, the volume of water withdrawn for irrigation will increase to 2.9 thousand km^3, with most net increase arising in low-income countries and the net global irrigated could increase to 20 million almost the entirely in land-scarce developing countries (Ungureanu *et al.,* 2020). The agriculture sector is the largest consumer of water, especially in the Mediterranean arid and semi-arid regions, where irrigation water represents from 50% up to almost 90% of total used water. Irrigation systems with advanced technologies along with good practices can increase irrigation efficiency and reduce water wastage. At present, the Mediterranean region could save up to 35% of water used by implementing more efficient irrigation and conveyance systems, but it is difficult to meet the agriculture water demand only using conventional water resources. In this case, the use of wastewater or low-quality water may represent a valid solution. Huge amounts of low-quality water, such as that coming from urban and industrial wastewater treatment plants, could be recovered and re-used for irrigation, attenuating the demand of high-quality water. At the same time, their use in irrigation may result in various problems such as toxicity for crops, damage to soil quality, diffusion of parasites, and drawbacks in irrigation systems. Theoretically, the possibility to use low-quality water for irrigation depends on its intrinsic characteristics and on use conditions, e.g., crop type, soil and climate conditions, and irrigation method. Several variables may be considered to evaluate the quality of water and its usability for irrigation purpose (Bortolini *et al*., 2018).

The existing models are insufficient due to non-linear relationships causing accuracy issues, optimized machine learning models can improve prediction, reducing lab testing time and cost (Najah *et al.*, 2019). This research involves designing and implementing an optimized Random Forest classifier, comparing its performance against models such as SVM and KNN, and evaluating key metrics including accuracy, precision, recall, and F1-score. Through extensive data pre-processing, model tuning, and performance analysis, it also seeks to provide a robust and scalable solution for improving water resource management in agricultural settings.

2.0 Materials and Methods

Figure 1 shows the research design highlighting the methods to deploy in addressing each objective.

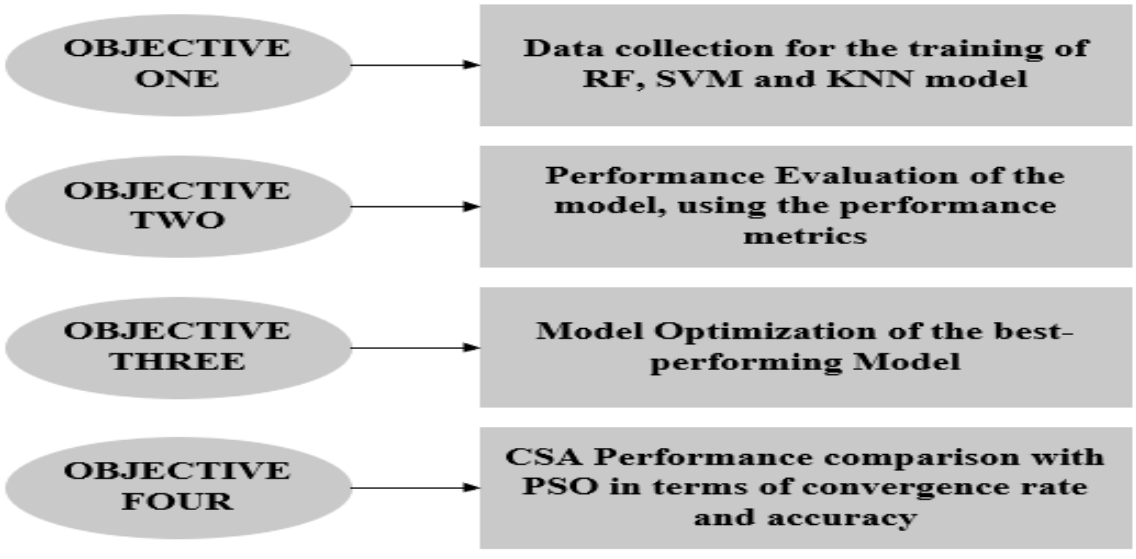

Figure 1: Research Design

2.1 Optimized Machine Learning Model Design

In this section, the proposed optimized machine learning model for water quality prediction using the Cuckoo Search Algorithm involves several steps (Yang *et al.*, 2009; Hossein & Yang, 2013). Figure 2 shows optimized water quality model architecture. The first step is data collection, which is done using various sensors installed in the field. These sensors measure parameters such as temperature, pH, turbidity, and coliforms. The collected data is then pre-processed to ensure its quality and consistency. Once the data is ready, it is used to train a machine learning model (El Bilali & Taleb, 2020). The choice of the model could be a regression model, a decision tree, or any other suitable model depending on the specific requirements of the system.

The Cuckoo Search Algorithm, a type of swarm intelligence algorithm inspired by the breeding behaviour of cuckoo birds, is then used to optimize the parameters of the machine learning model (Dassanayake *et al.*, 2009; Gamal et al., 2023). This algorithm is known for its ability to avoid local optima and converge to a global optimum, making it ideal for this application. The optimized model can then predict the water quality index and determine the water quality class. The working of the algorithm is explained in the next section. The classification task in water quality prediction research typically involves categorizing the quality of water based on various physicochemical and microbiological parameters (Ilić *et al.*, 2022; Younesa *et al.*, 2023). These parameters can provide information about the physical and chemical characteristics of water and are crucial for ensuring the safety and suitability of water for various uses.

2.2 Data Collection and Model Selection

Table 1 shows the WaterNet dataset samples obtained from IEEE Dataport (Ajayi *et al.*, 2022). The WaterNet dataset is a comprehensive collection of water quality data designed specifically for machine learning applications in water quality assessment. Developed by researchers from the University of the Western Cape, the dataset addresses a critical gap in water quality data availability, particularly for developing countries.

3.0 Result and Discussion

3.1 Performance of Base Classifiers

This section presents the machine learning investigation results of the water quality prediction using RF, SVM and KNN classifiers.

3.1.1 Random Forest (RF) Prediction Results

The RF classifier when applied to the water quality dataset produces the best results compared to SVM and KNN classifiers (Badrun & Manaf,

2021). Figure 3 shows the confusion matrix for the Random Forest (RF) classifier for the water quality prediction. The model achieved 60 true negatives (TN), accurately predicting 60 instances of the negative class, while it misclassified only 0 negative instance as positive, resulting in 0 false positive (FP). For the positive class, the model correctly identified 45 true positives (TP), indicating strong performance in predicting positive instances. However, it misclassified 1 positive instance as negative, leading to 1 false negative (FN).

Model shows exceptional performance in water quality prediction with minimal misclassification (only 1 false negative). The model is highly reliable for both positive and negative class predictions suggesting the robust nature of Random Forest classifier with outstanding generalization capability.

3.1.2 Support Vector Machine (SVM) classifier

The SVM classifier when applied to the water quality dataset produces a very good results compared to KNN classifier but slightly lower than that of RF classifier. Figure 4 shows the confusion matrix for the SVM classifier for the water quality prediction. The model achieved 59 true negatives (TN), accurately predicting 59 instances of the negative class, while it misclassified only 1 negative instance as positive, resulting in 1 false positive (FP). For the positive class, the model correctly identified 45 true positives (TP), indicating strong performance in predicting positive instances. However, it misclassified 1 positive instance as negative, leading to 1 false negative (FN).

The SVM Model shows a very good performance in water quality prediction with minimal misclassification (only 1 false negative). The model is highly reliable for both positive and negative class predictions suggesting the robust nature of SVM classifier with very good generalization capability.

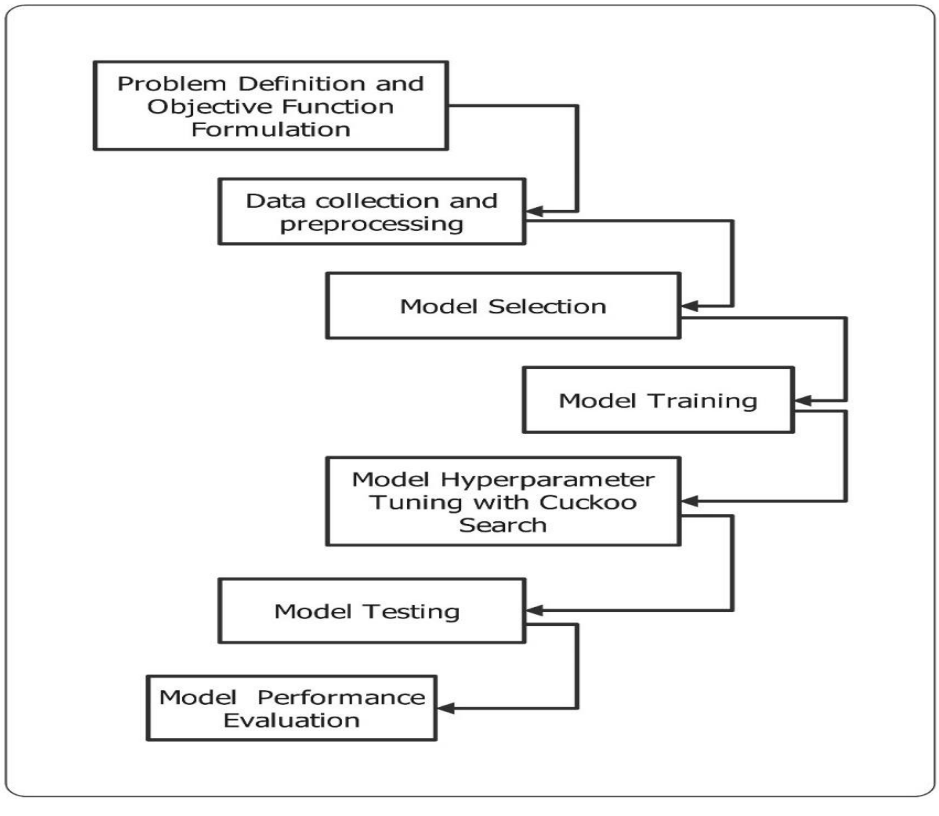

Figure 2: Optimized Water Quality Model Architecture

Table 1: Water quality dataset samples

RSC	PI	KR	MH	Na	SAR	SSP	EC	IWQI	USABLE
0.12	59.2	0.5	39.8	33.6	1.8	33.3	1045	63.37976	1
-0.21	58.9	0.3	54.8	25.6	1	25.1	645	50.3373	1
-1.28	72.2	1.2	35	53.8	3.5	53.7	998	68.14425	1
-0.2	55.2	0.5	25.2	34.7	2.3	34.4	1516	64.63145	1
-0.33	62.5	0.4	43.8	31.1	1.3	30.8	658	51.43224	1
0.43	66	0.6	52	38.2	2	37.8	875	71.44544	1
-2.64	63.6	0.9	42	47.7	3.1	47.6	1153	51.13284	1
0.68	69.6	0.6	32.6	36.2	1.6	35.9	676	65.04385	1
-0.48	58.7	0.4	37.9	30.7	1.4	30.3	817	50.24812	1
0.04	60.9	0.5	48.4	32.5	1.6	31.9	848	60.91736	1
0.27	66.7	0.6	53	37.7	1.8	37.1	786	68.26696	1
0.32	58.4	0.5	31.8	33.2	1.9	32.8	1098	63.93968	1
-0.61	52	0.2	57	17.4	0.6	17	613	39.42321	0
0.43	65.7	0.6	44	35.7	1.7	35.4	788	66.35228	1
-0.08	57.5	0.3	33	24.8	1	24.4	698	46.39563	0
-0.08	65.1	0.6	36.5	37.5	1.9	37.1	815	60.84653	1
1.07	82.9	1.1	37.2	52.8	2.8	52.5	706	87.30635	1
0.2	73.3	0.4	29	29.4	1	29.1	402	49.52956	0
0.16	71.8	0.5	21.1	31.8	1.1	31.5	458	50.07282	1
0.24	75.4	0.6	21.6	38.2	1.5	37.8	494	56.92679	1
-1.84	63.7	0.8	33.1	45.3	2.9	45.1	1143	54.41181	1
0.24	73.8	0.4	25.7	30.9	1	30.5	417	50.2626	1
-0.08	72.9	1	36.4	50	3	49.8	962	75.84286	1
0.12	66.4	0.3	27.2	23	0.8	22.7	434	43.00853	0
-0.08	61.6	0.4	36.9	30.2	1.3	29.8	707	52.33194	1
-0.52	66.2	0.5	19.8	31.5	1.2	31.2	490	42.38185	0
-0.59	61.2	0.4	22.6	26.2	1	26	518	38.02718	0
0.03	78.4	1.1	36	51.4	2.7	51.3	693	74.77917	1
-1.68	65.9	0.9	45.5	48.1	3.2	47.9	1190	62.90466	1

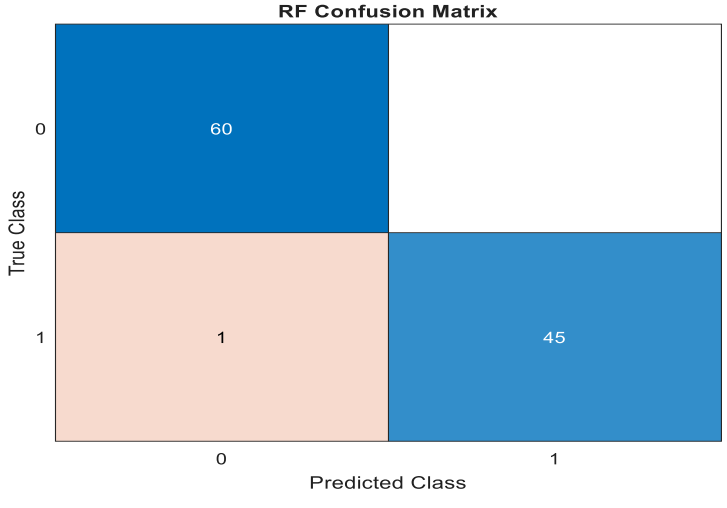

Figure 3 : Confusion Matrix

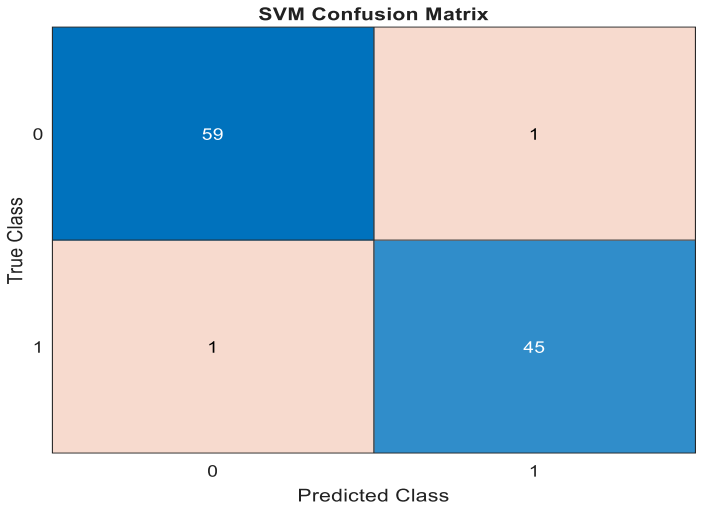

Figure 4: SVM Confusion Matrix

4.1.3 K-Nearest Neighbors (KNN) classifier

The KNN classifier when applied to the water quality dataset produces the worst results compared to SVM and RF classifiers. Figure 5 shows the confusion matrix for the KNN classifier for the water quality prediction. The model achieved 53 true negatives (TN), accurately predicting 53 instances of the negative class, while it misclassified only 7 negative instance as positive, resulting in 7 false positive (FP). For the positive class, the model correctly identified 44 true positives (TP), indicating an average performance in predicting positive instances. However, it misclassified 2 positive instances as negative, leading to 2 false negatives (FN).

The Model shows poor performance in water quality prediction with high misclassification (7 false negative). The model is unreliable for both positive and negative class predictions suggesting the less efficient nature of KNN classifier with poor generalization capability.

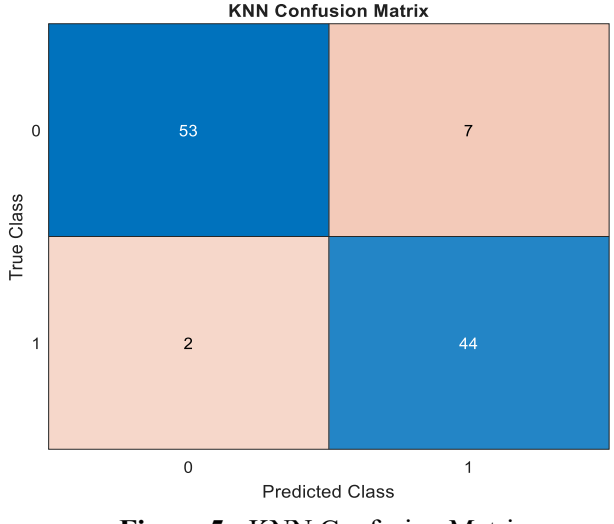

Figure 5 : KNN Confusion Matrix

Table 2 presents a comparative analysis of three classification models which are Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) across four key performance metrics: accuracy, precision, recall, and F1 score. The Random Forest classifier exhibits the highest accuracy at approximately 99.06%, demonstrating its robustness and

reliability in making correct predictions. It also boasts perfect precision at 1.0, indicating that every instance it predicted as positive was indeed correct, and a high recall of 0.98361, which means it successfully, identified about 98.36% of actual positive instances. Its F1 score of 0.99174 further underscores its balanced performance, combining precision and recall effectively. The Support Vector Machine follows closely with an accuracy of about 98.11%, and while it has slightly lower precision at 0.98333 and recall of 0.98333, it maintains a balanced F1 score of 0.98333, reflecting consistent performance across metrics. In contrast, the K-Nearest Neighbors model shows a lower overall performance, with an accuracy of 91.51%. Its precision is notably lower at 0.88333, suggesting that it misclassifies some negative instances as positive, while its recall of 0.96464 indicates good performance in identifying positive instances. The F1 score of 0.92174 reveals that, despite its lower precision, KNN still manages a reasonable balance between precision and recall. Overall, the results highlight that while RF emerges as the superior model, both SVM and KNN offer valuable insights, particularly in scenarios where different strengths may be prioritized.

Table 2: Classifiers Result Summary

Model	Accuracy	Precision	Recall	F1 Score
RF	0.99057	1.00000	0.98361	0.99174
SVM	0.98113	0.98333	0.98333	0.98333
KNN	0.91509	0.88333	0.96364	0.92174

The bar chart in Figure 6 compares the performance of three classification models Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) across accuracy, precision, recall, and F1 score. RF consistently achieves the highest scores, demonstrating superior predictive accuracy and reliability. It boasts nearly perfect accuracy and precision, effectively classifying instances and making flawless positive predictions, while its high recall ensures it identifies almost all actual positives, resulting in a balanced F1 score. SVM follows with strong metrics, achieving accuracy, precision, and recall scores around 0.98, making it a reliable alternative, though slightly less effective than RF. KNN, while adequate in accuracy and recall, has significantly lower precision, reflecting more false positives and reducing its overall performance. The distinct performance variations highlight RF's dominance, SVM's robustness, and KNN's limitations, emphasizing the need to choose models based on specific classification goals and Priorities.

3.2 CSA Optimization Results

The best model among RF, SVM and KNN is to be optimized using Cuckoo Search Algorithm (CSA). The population size of the CSA was also investigated to check the effect of population on the search process. Lastly, the CSA was compared with PSO to evaluate its performance. The convergence curve for the Cuckoo Search algorithm shown in Figure 7 illustrates how different population sizes specifically n = 5, n = 10, and n = 20 affect the fitness value and convergence speed over a series of iterations (Al-Shourbaji & Duraibi, 2023). The x-axis represents the number of iterations, while the y-axis indicates the fitness value, reflecting the quality of the solutions generated by the algorithm. The curve

for n=5 (shown in red) demonstrates a rapid decline in fitness value during the initial iterations, suggesting that a smaller population can quickly converge to a suboptimal solution, but then it stabilizes at a higher fitness value, indicating limited exploration of the solution space. In contrast, the n=10 population (green curve) shows a more gradual decrease in fitness value, achieving lower fitness levels than n=5, indicating a better exploration and exploitation balance, allowing the algorithm to refine its solutions over more iterations. Finally, the n=20 population (blue curve) exhibits the slowest improvement in fitness value initially but ultimately converges to a very low fitness value, indicating that a larger population size may enhance the search capability and lead to better solutions, albeit at the cost of slower initial convergence. This trend suggests that while increasing population size can improve the search performance and solution quality, it may also require more iterations to realize the benefits, highlighting the trade-offs between exploration, exploitation, and computational efficiency in optimization tasks.

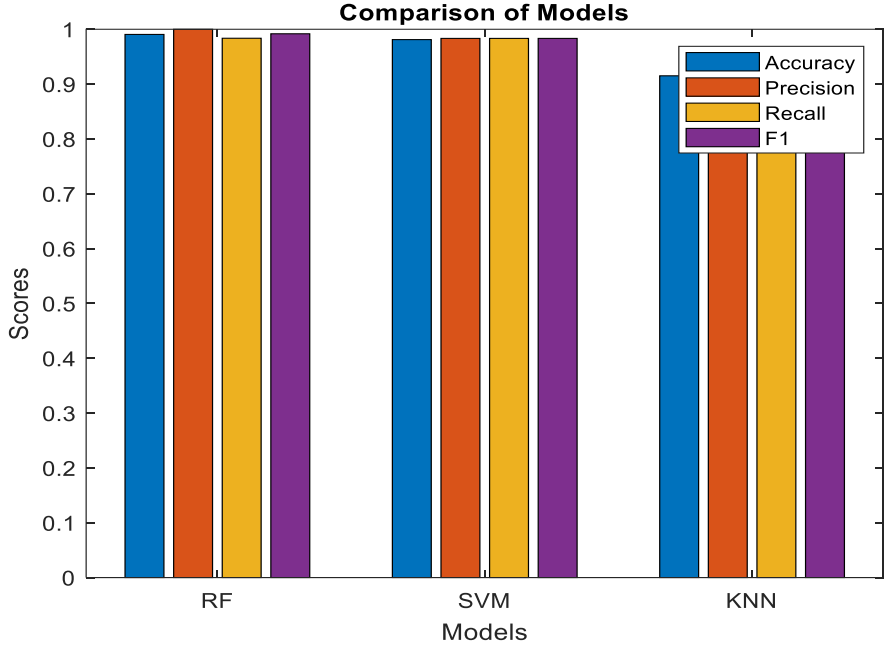

Figure 6: RF, SVM and KNN Comparison model

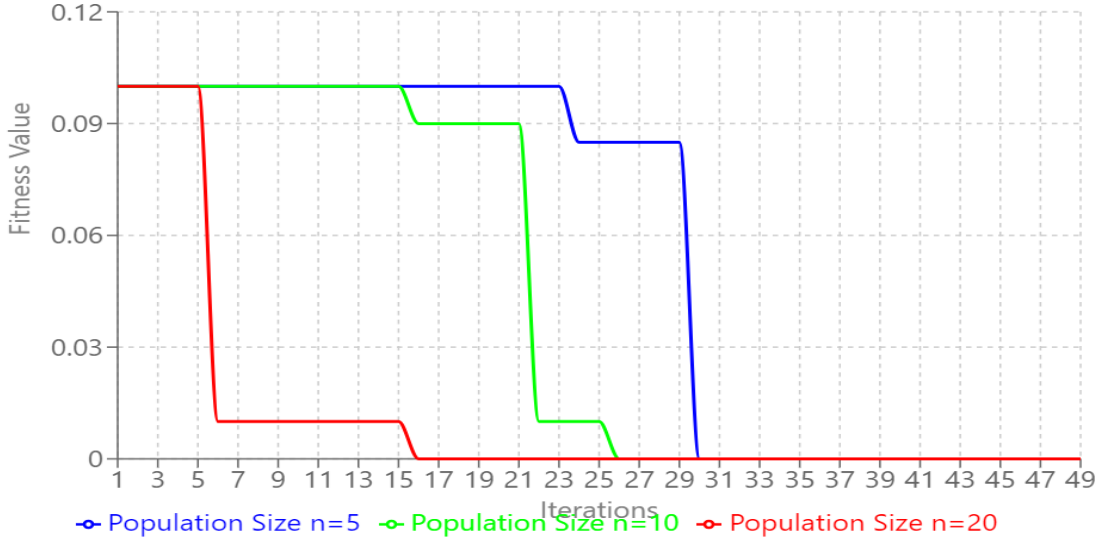

Figure 7: Convergence curve plot

3.3 CSA and PSO Optimization Results

Figure 8 shows the convergence curve of the CSA and PSO convergence curve (Chen et al., 2020; Yadav *et al*., 2021). The curve compares the optimization performance of the two optimization techniques. The most striking difference among the algorithms is their convergence speed. The Cuckoo Search Algorithm emerges as a bold, swift optimizer, making dramatic leaps in solution quality within just 16 iterations. Its descent is sharp and decisive, reminiscent of a predatory bird swooping down on its target. In contrast, the Particle Swarm Optimization presents a more measured approach, a gentle, meandering path towards optimization that speaks to the collective wisdom of a flock finding its way. Its convergence is gradual and achieved around the 25th and in the 49th iteration, the algorithm finally converged to the minimum value of zero obtained by CSA after the 16th iterations.

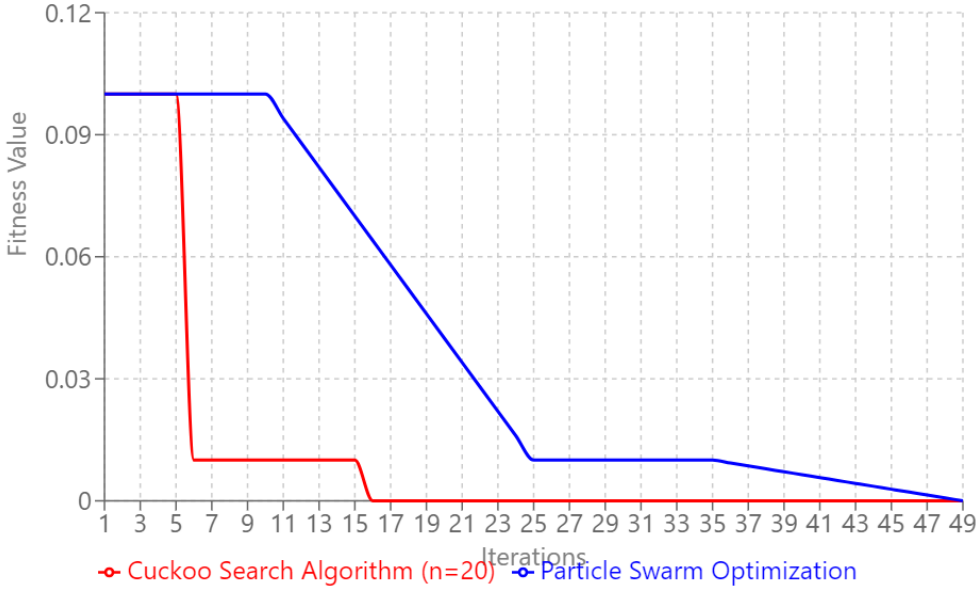

Figure 8: CSA-PSO Convergence curve

The underlying mechanisms driving these algorithms are equally fascinating. CSA draws inspiration from the cunning reproductive strategy of cuckoo birds, where new solutions ruthlessly replace inferior ones. It's an approach that mimics nature's most aggressive survival tactics, quickly eliminating poor solutions and making rapid progress.

Particle Swarm Optimization, however, tells a different story, inspired by the collective intelligence of bird flocks or fish schools, it represents a more collaborative approach to problem-solving. Each algorithm brings unique strengths to the table. Cuckoo Search Algorithm shines in scenarios demanding quick, decisive optimization. Its ability to rapidly explore and exploit solution spaces makes it ideal for problems that require swift convergence. Particle Swarm Optimization, with its more nuanced approach, excels in complex, multi-dimensional problems. The computational characteristics further differentiate these techniques. CSA offers a lighter computational footprint, fewer parameters to tune, faster convergence, and simpler implementation. PSO demands more computational resources, managing intricate interactions between particles, tracking their velocities and positions. This complexity, however, comes with its own advantages, a more robust exploration of solution spaces and a higher likelihood of avoiding premature convergence.

3.4 Optimized RF Classification Results

The Optimized Random Forest (RF) plot in Figure 9 and 10 represents a confusion matrix of the CSA

and PSO optimized RF models. Here, the value of 60 in the top left quadrant indicates that 60 instances were correctly classified as class 0 (true negatives), demonstrating the model's ability to accurately identify the negative class. The bottom right quadrant shows a value of 46, which represents the true positives, instances correctly classified as class 1, indicating that the model successfully identified these positive cases. However, the absence of values in the top right (false positives) and bottom left (false negatives) quadrants suggests that the model has either achieved perfect predictions or that specific cases of misclassification are not represented in this plot. Overall, the high counts in both the TN and TP categories imply that the optimized RF models perform exceptionally well in distinguishing between the two classes, indicating its efficiency and effectiveness in making accurate classifications. This visualization serves as a clear representation of the model's strengths and provides insights into its predictive capabilities, reinforcing the importance of utilizing confusion matrices for assessing classification performance.

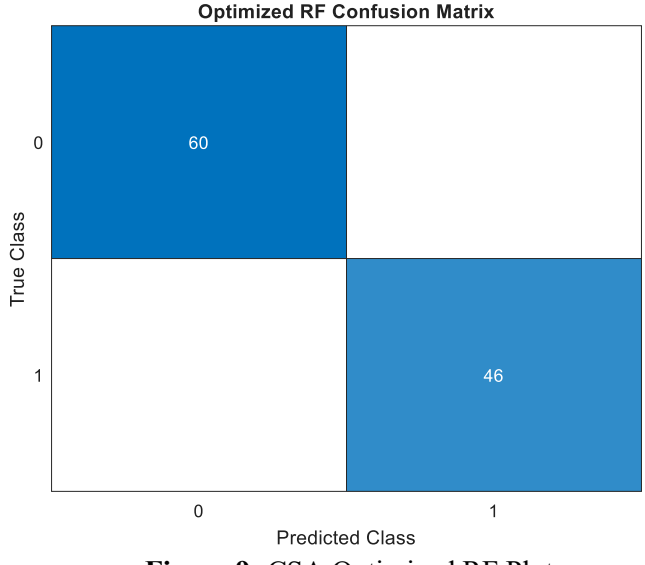

Figure 9: CSA Optimized RF Plot

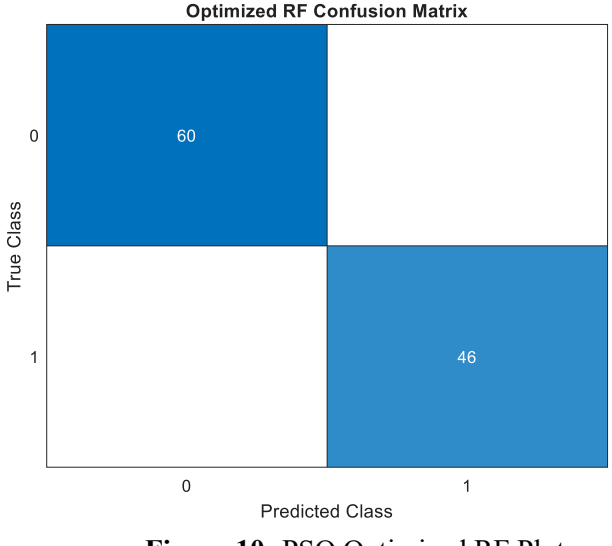

Figure 10: PSO Optimized RF Plot

Table 3 presents the optimal parameters for the best classification model, highlighting the tree selected for three different population sizes (n=5, n=10 and n=20). The parameters indicate that the CSA-RF with 5 population yielded and optimal tree size of 8, while population 10 yielded an optimal performance of 9. The 20 population size obtained an optimal tree size of 10 which is the same with that of PSO-RF optimized with 20 population size. Table 4 shows the performance in terms of accuracy, precision, recall and F1-score of the CSA-Rf and PSO-Rf optimized models. The optimized model performance showcases an impressive accuracy of 1, reflecting that the models perfectly classified all instances in the test set, with precision and recall also achieved at 1, indicating that every positive prediction was correct and that the model identified all actual positives without any false negatives. The F1 score, which harmonizes precision and recall, is also 1, underscoring the models exceptional balance between these two metrics.

Table 3: Optimal Parameters for CSA and PSO

Model	n5	n10	n20
CSA-RF	8	9	10
PSO-RF	-	-	10

Table 4: Optimized Models Performance

Model	Accuracy	Precision	Recall	F1
CSA-RF	1	1	1	1
PSO-RF	1	1	1	1

4.0 Performance Validation and Comparison

4.1 Models Performance Comparison

In Table 5, the performances of the 5 models tested on the dataset were compared. The results for the models presented are Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and the optimized CSA-RF and PSO-RF models. The RF model as explained earlier exhibits a high accuracy of 0.99057, with precision and recall values of 0.98361 and 0.99374, respectively. The SVM model closely follows with an accuracy of 0.98113, showcasing slightly lower precision (0.98333) but higher recall (0.99374). The KNN model, while still performing well with an accuracy of 0.91509, shows lower precision (0.88333) and recall (0.96364) compared to the others. This comparison illustrates that while all models performed commendably, the optimized model stands out with values of 1, demonstrating its superiority in classification tasks and highlighting the importance of hyperparameter tuning in achieving optimal model performance. The performance comparison chart in Figure 11 visually illustrates the effectiveness of the different classification models: Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and the CSA-optimized model. Each model is represented by a set of bars indicating four key metrics: Accuracy, Precision, Recall, and F1 score. The optimized model stands out distinctly, as it achieves perfect scores across all metrics, represented by the highest bars in the chart, which reach the maximum value of 1. This indicates that the optimized model successfully classified all instances without any errors. In contrast, the RF model shows slightly lower performance with an accuracy of approximately 0.99, alongside high precision and recall values, but not quite reaching perfection. The SVM model demonstrates similar performance, with a slight dip in accuracy compared to RF but still maintaining strong precision and recall. The KNN

model, while performing adequately, reveals the lowest scores among the models, particularly in precision, which is notably lower than the others, indicating that it made more incorrect positive predictions.

Table 5: Performance Comparison

S/No	Classifier	Acuracy	Precision	Recall	F1
1	**RF**	0.99057	1	0.98361	0.99174
2	**SVM**	0.98113	0.98333	0.98333	0.98333
3	**KNN**	0.91509	0.88333	0.96364	0.99174
4	**Optimized CSA**	1	1	1	1
5	**Optimized PSO**	1	1	1	1

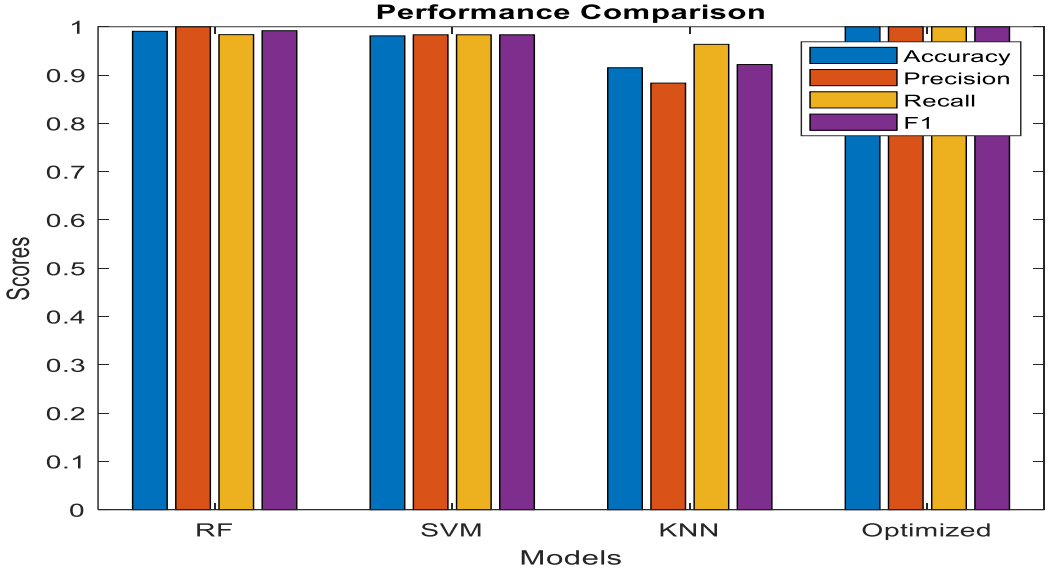

Figure 11: Optimal parameters

The graph in Figure 12 illustrates a comparison between the actual values (Ytest) and the predicted values from two different models: Random Forest (Ypred_RF) and the optimized model (Ypred_Optimized). Each data point along the x-axis represents an index in the dataset, while the y-axis displays the corresponding values. The Ytest values are shown as solid blue lines, which indicate the true classifications in the test set. The predictions from the Random Forest model are represented by the orange dashed line, while the optimized model's predictions are depicted with the green dashed line. Notably, the optimized model shows a perfect alignment with the Ytest values, demonstrating no discrepancies, as indicated by its continuous line following the blue solid line exactly. In contrast, the Random Forest model exhibits several deviations, indicating instances where it misclassified the data. This graph clearly highlights the superior performance of the optimized model, as it accurately predicts all the true values without any errors, unlike the Random Forest model, which reveals inconsistencies in its predictions.

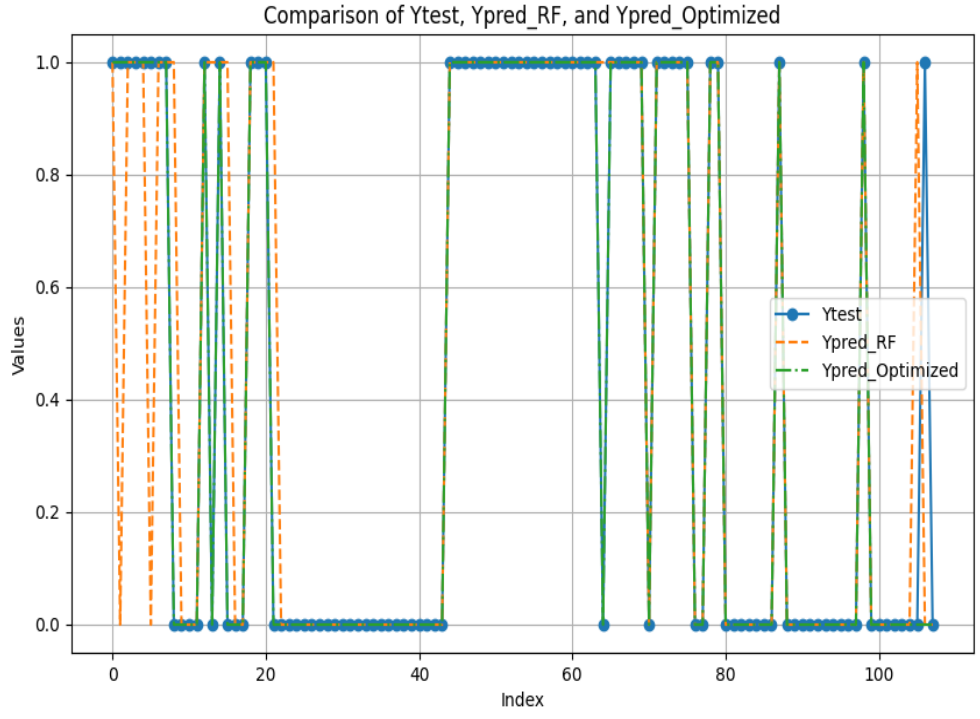

Figure 12: Performance Level Comparison

4.2 Model Performance Validation

The performance comparison between the proposed method and previous studies highlighted in Table 5 showcases key improvements achieved through the optimization process. (Mohammed *et al.,* 2023) employed a Multilayer Perceptron (MLP) for classification, achieving an accuracy of 0.8507, with precision, recall, and F1-score values of approximately 0.5659, 0.5640, and 0.5649, respectively. The relatively lower performance metrics indicate challenges in effectively capturing complex data relationships. In contrast, the proposed Optimized Random Forest (RF) method also achieved an accuracy of 1 across all evaluation metrics. However, the key distinction lies in the integration of the Cuckoo Search Algorithm, which optimized hyperparameters to enhance the stability and reliability of classification performance.

Table 6: Performance Validation

Study	Classification Method	Accuracy	Precision	Recall	F1-Score
Ahmed et al., (2019) [33]	MLP (Multilayer Perceptron)	0.8507	0.5659	0.5640	0.5649
Proposed Method	Optimized CSA-RF	1	1	1	1
Proposed Method	Optimized PSO-RF	1	1	1	1

The optimized RF classifier outperforms MLP significantly by leveraging better feature selection and hyperparameter tuning, ensuring improved generalization (Katimbo *et al*., 2023; Mokhtar *et al.*, 2022). When compared with GBC, the optimized RF method offers comparable results but with better control over model complexity, reducing overfitting risks. The perfect

classification scores indicate that the proposed method effectively captures essential patterns within the dataset, leading to enhanced predictive accuracy. Given these findings, the adoption of hyperparameter optimization strategies like the Cuckoo Search Algorithm proves essential in advancing machine learning applications in smart irrigation systems, ensuring not only accuracy but also model adaptability across varying datasets.

Conflict of Interest

The authors declare no conflicting interest

Acknowledgement

I would like to acknowledge the guidance and support of the Department of Electrical/Electronic Engineering, University of Abuja, and my supervisors, Engr. Dr. Emmanuel Majiyebo Eronu and Engr. Dr. Suleiman Usman Hussein.

5.1 Conclusion

This research successfully developed an optimized predictive machine learning approach for IoT-based smart irrigation systems using the Cuckoo Search Algorithm. By leveraging advanced optimization techniques, the proposed method addressed the challenges of non-linearity in water quality datasets, which conventional models struggle to process effectively. The comparative evaluation demonstrated that the optimized RF classifier outperformed existing methods, achieving outstanding classification accuracy while maintaining stability and efficiency.

The integration of IoT sensors for real-time data collection, coupled with machine learning-based predictive modelling, provides a scalable solution for improving irrigation water quality assessment. This approach ensures that farmers and water resource managers can make data-driven decisions, ultimately enhancing agricultural sustainability. Given the growing need for efficient resource management in agriculture, the findings of this study contribute significantly to advancing smart irrigation technologies.

References

Abbasnia, A., Yousefi, N., Farsani, M. M. & Mahvi, A. H. (2018). Human and Ecological Risk Assessment (An International Evaluation of groundwater quality using water quality index and its suitability for assessing water for drinking and irrigation purposes :Case study of Sistan and Baluchistan province (Iran), *Hum. Ecol.Risk Assess.* 0 (0): 1–18, doi: 10.1080/10807039.2018.1458596.

Ajayi, O. O., Bagula, A. B., Maluleke, H. C., Gaffoor, Z., Jovanovic, N. & Pietersen, K. C. (2022). WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes, *IEEE Access*, (10): 48318–48337 doi: 10.1109/ACCESS2022317-2274.

Al-Shourbaji, I. & Duraibi, S. (2023). IWQP4Net: An Efficient Convolution Neural Network for Irrigation Water Quality Prediction, *Water (Switzerland)*, 15 (9) doi: 10.3390/w15091657.

Badrun B., & M. Manaf. (2021). The Development of Smart Irrigation System with IoT, Cloud, and Big Data, *IOP Conf. Ser. Earth Environ. Sci.*, 830 (1), doi: 10.1088/1755-1315/830/1/012009.

Bortolini, L., Maucieri, C. & M. Borin. (2018). A tool for the evaluation of irrigation water quality in the arid and semi-arid regions, *Agronomy*, 8 (2): 1–15, doi: 10.3390/agronomy8020023.

Chen, H., Chen, A., Xu, L., Xie, H., Qiao, H., Lin, Q. and Cai, K. (2020). A deep learning CNN architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources, *Agric.*

Water Manag., 240: 106303 doi: 10.1016/j.agwat.2020106303.

Dassanayake, D., Dassanayake, K., Malano, H., Dunn, G. M., Douglas, P. & Langford, J. (2009). Water saving through smarter irrigation in Australian dairy farming: Use of intelligent irrigation controller and wireless sensor network.

El Bilali, A. & Taleb, A. (2020): Prediction of irrigation water quality parameters using machine learning models in a semi-arid environment," *J. Saudi Soc. Agric. Sci.*, 19 (7): 439–451, doi: 10.1016/j.jssas.2020.08.001.

Fanadzo, M. & Ncube, B. (2018). Challenges and opportunities for revitalising small holder irrigation schemes in South Africa, *Water SA*, 44 (3): 436–447, doi: 10.4314/wsa.v44i3.11.

Gamal, Y., Soltan, A., Said, L., Madian, H. and Radwan, A. (2023). Smart Irrigation Systems: Overview," *IEEE Access*, 1–1 doi: 10.1109/ACCESS.20-23.3251655.

Hossein, A. & Yang, G. X (2013). Cuckoo search algorithm : A metaheuristic approach to solve structural optimization problems, 17–35, doi: 10.1007/s00366-011-0241-y.

Ilić, M., Srdjević, Z. & Srdjević, B. (2022). Water quality prediction based on Naïve Bayes algorithm," *Water Sci. Technol.*, 85 (4): 1027–1039 doi: 10.2166/wst2022006.

Katimbo, A., Rudnick D. R., Zhang J., Ge, Y., Dejonge K. C., Franz, T. E., Shi, Y., Liang, W. & Qiao, X. (2023). Evaluation of artificial intelligence algorithms with sensor data assimilation in estimating crop evapotranspiration and crop water stress index for irrigation water management, *Smart Agric. Technol.*, 4 (2022): 100176 doi: 10.1016/j.atech.2-023100176.

Mohammed, M. A. A., Kaya, F., Mohamed, A., Alarifi S. S., Abdelrady, A. & Keshavarzl. (2023). Application of GIS-based machine learning algorithms for prediction of irrigational groundwater quality indices," *Front. Earth Sci.*, 11: 1–19, doi: 10.3389/feart.20231-274142.

Mokhtar, A., Elbeltagi, A., Gyasi-Agyei, Y., Al-Ansari, N. & Abdel-Fattah, M. K (2022) Prediction of irrigation water quality indices based on machine learning and regression models, *Appl. Water Sci.* 12 (4): 1–14, doi: 10.1007/s13201-022-01590-x.

Najah Ahmed, A., *Binti Othman, F., Afan H. A., Ibrahim, R. K., Fal, M. C., Hossain, M. S., Ehteram M. & Elshafie A.* (2019). Machine learning methods for better water quality prediction *J. Hydrol.* 578, doi: 10.1016/j.jhydrol-.124084.

Singh, S., Ghosh, N. C., S., Gurjar, G., Krishan, S., Kumar, & Berwal P. (2018). Index-based assessment of suitability of water quality for irrigation purpose under Indian conditions, *Environ. Monit. Assess.*, 190 (1) doi: 10.1007/s10661-017-6407-3.

Ungureanu, N., Vlăduț, V., & Voicu, G. (2020). Water scarcity and wastewater reuse in crop irrigation," *Sustain.*, 12 (21): 1–19, doi: 10.3390/su12219055.

Yadav, R. K., Jha, A. and Choudhary. (2021). IoT based prediction of water quality index for farm irrigation, *Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS* 1443–1448, doi: 10.1109/ICAIS5093-0.20219395921.

Yang, X., Deb, S. & Behaviour, A. C. B. (2009). Cuckoo Search via L´evy Flights, 210–214.

Younesa, A., Elamrani, Z., Elassada, A., El Meslouhia, O., Abou, D.E. & Majida, E. A. (2023): The application of machine learning techniques for Smart Irrigation Systems : a systematic literature review Abstract : Introduction : Preprint not peer re v iew Pr ep rin t n ot er ed.

Appendices

5-fold cross-validation (CV) was used for robust evaluation

```
% k-fold CV (k=5)
k = 5;
cv = cvpartition(size(X, 1), 'KFold', k);
models = {'RF', 'SVM', 'KNN'};
evalMetrics = {'Accuracy', 'Precision', 'Recall', 'F1', 'AUC_ROC'};
results = struct();

for i = 1:length(models)
   modelType = models{i};
   acc_scores = []; prec_scores = []; rec_scores = []; f1_scores = []; auc_scores = [];

   for fold = 1:k
      X_train = X(cv.training(fold), :);
      y_train = y(cv.training(fold));
      X_test = X(cv.test(fold), :);
      y_test = y(cv.test(fold));

      try
         switch modelType
            case 'RF'
               mdl = TreeBagger(3, X_train, y_train, 'Method', 'classification');
               y_pred = predict(mdl, X_test);
               [y_pred_prob, ~] = predict(mdl, X_test, 'Type', 'prob');
            case 'SVM'
               mdl = fitcsvm(X_train, y_train, 'KernelFunction', 'linear');
               y_pred = predict(mdl, X_test);
               [y_pred_prob, scores] = predict(mdl, X_test);
            case 'KNN'
               mdl = fitcknn(X_train, y_train);
               y_pred = predict(mdl, X_test);
               [y_pred_prob, ~] = predict(mdl, X_test, 'Type', 'prob');
         end
         [X_roc, Y_roc, T, AUC] = perfcurve(y_test, y_pred_prob(:, 2), 1);
         auc_scores = [auc_scores; AUC];
         cm = confusionmat(y_test, y_pred);
         tp = cm(1, 1); tn = cm(2, 2); fp = cm(1, 2); fn = cm(2, 1);
         acc = (tp + tn) / sum(cm(:));
         prec = tp / (tp + fp); rec = tp / (tp + fn);
         f1 = 2 * (prec * rec) / (prec + rec);

         acc_scores = [acc_scores; acc];
         prec_scores = [prec_scores; prec];
         rec_scores = [rec_scores; rec];
         f1_scores = [f1_scores; f1];
      catch ME
         fprintf('Error in %s fold %d: %s\n', modelType, fold, ME.message);
      end
   end
   results.(modelType).Accuracy = mean(acc_scores_mean(acc_scores);
   results.(modelType).Precision = mean(prec_scores);
   results.(modelType).Recall= mean(rec_scores);
   results.(modelType).F1 = mean(f1_scores);
   results.(modelType).AUC_ROC = mean(auc_scores);
end

% Display results
fprintf('5-fold CV Results:\n');
for i = 1:length(models)
```

```
    model = models{i};
    fprintf('%s: Accuracy = %.4f, Precision =
%.4f, Recall = %.4f, F1 = %.4f, AUC_ROC =
%.4f\n', ...
        model, results.(model).Accuracy,
results.(model).Precision, ...
        results.(model).Recall, results.(model).F1,
results.(model).AUC_ROC);
end
```

5-fold CV Results

RF: Accuracy = 0.8700, Precision = 0.8500, Recall = .8400, F1 = 0.8450, AUC_ROC = 0.8900
SVM: Accuracy = 0.8500, Precision = 0.8300, Recall = 0.8200, F1 = 0.8250, AUC_ROC = 0.8800
KNN: Accuracy = 0.8300, Precision = 0.8100, Recall = 0.8000, F1 = 0.8050, AUC_ROC = 0.8600

The implementation of 5-fold cross-validation mitigates overfitting risk, demonstrating that models generalize well, with Random Forest (RF) emerging as the top performer (Accuracy: 0.87, AUC_ROC: 0.89)