Developing a Smart, Integrated Irrigation Systems Based on Machine Learning (ML) and Internet of Things (IoT) by Employing the K-Nearest Neighbour (KNN) Algorithm

Aakash Kharb
Maharshi Dayanand University, Rohtak

[1]*Received: 28 Nov 2023; Accepted: 15 Jan 2024; Published: 05 Feb 2024*

ABSTRACT

Rapid population growth, freshwater scarcity and climate variability are putting severe pressure on conventional irrigation practices. Smart irrigation systems that combine Internet of Things (IoT) sensing with machine-learning (ML) prediction provide a promising pathway to improve water-use efficiency, crop yield and operational reliability. This paper presents an IoT and ML-based irrigation system in which a k-nearest neighbors (KNN) model recommends irrigation actions (irrigate now, delay, or skip) based on soil, weather and crop features. We first review recent work on IoT-enabled smart irrigation and ML-driven decision support, highlighting the growing role of KNN in irrigation scheduling. We then propose a layered system architecture integrating low-cost field sensors, an edge gateway, cloud storage and a KNN-based analytics service. The methodology section details feature engineering, KNN formulation, and a representative evaluation protocol. Using an illustrative scenario grounded in ranges reported in recent literature, we compare KNN with logistic regression, support vector machines (SVM) and random forests in terms of accuracy, F1-score, latency, implementation complexity and interpretability. The comparative analysis shows that KNN achieves competitive accuracy with simple implementation and robust performance for small to medium datasets, making it attractive for resource-constrained deployments. We conclude with a discussion of practical challenges such as sensor noise, model update strategies and scalability, and outline future extensions including hybrid KNN–deep learning architectures and federated learning for cross-farm collaboration.

Keywords: *Smart irrigation; Internet of Things; Machine learning; k-nearest neighbors; Precision agriculture; Soil moisture; Water-use efficiency*

1. Introduction

Agriculture accounts for more than 70% of global freshwater withdrawals, and in many arid and semi-arid regions irrigation is the dominant consumer of water resources. Efficient irrigation scheduling is therefore critical both for food security and for sustainable water management. Smart irrigation systems that combine IoT sensors (soil moisture, temperature, humidity, rainfall) with ML models are increasingly being deployed to automate and optimize irrigation decisions.

Recent work has demonstrated that IoT-based smart irrigation, when paired with ML, can reduce water consumption by 20–40% compared with fixed-schedule irrigation while maintaining or improving yield. Within this space, KNN stands out as a simple yet powerful non-parametric algorithm with strong performance for classification and regression tasks on tabular sensor data.

However, most published systems still focus on neural networks, fuzzy logic or ensemble methods, and often treat KNN only as a baseline rather than as the core decision engine. There is still room to design and analyze an end-to-end IoT irrigation system explicitly built around KNN, including a clear discussion of trade-offs against alternative models.

[1] **How to cite the article**: Kharb A (2024) Developing a Smart, Integrated Irrigation Systems Based on Machine Learning (ML) and Internet of Things (IoT) by Employing the K-Nearest Neighbour (KNN) Algorithm; *International Journal of Technology, Science and Engineering*; Vol 7 Issue 1; 25-33

Table 1. High-level challenges motivating IoT+ML-based irrigation

Challenge	Impact on irrigation	Role of IoT+ML
Water scarcity & drought	Need to minimize waste	Predict soil moisture & optimal irrigation timing
Climate variability	Uncertain weather, ET_0	Fuse sensor & weather data for adaptive decisions
Labour constraints	Manual monitoring is costly	Remote sensing & automated scheduling
Heterogeneous soils & crops	One schedule doesn't fit all	Field-specific models based on local sensor data
Limited farmer analytics	Reliance on intuition	Transparent recommendations & decision support dashboards

The objective of this paper is to design and analyze an IoT and ML-based irrigation system with KNN as the core decision algorithm, and to compare its performance and suitability against alternative approaches.

2. Background and Related Work

2.1 IoT-Based Smart Irrigation

IoT-enabled irrigation uses distributed sensor nodes to monitor soil, plant and atmospheric conditions in real time, transmitting data to edge or cloud platforms for decision-making. A comprehensive review of smart irrigation systems shows increasing use of wireless sensor networks, low-power radios, and cloud services to implement precision irrigation at scale.

Tace et al. proposed a "smart irrigation system based on IoT and machine learning," using soil moisture, temperature and rainfall sensors connected to Node-RED and MongoDB, and evaluated several ML algorithms (KNN, logistic regression, neural networks, SVM, Naïve Bayes). KNN achieved the highest recognition rate of 98.3% with a low RMSE, validating its suitability for this domain.

Abioye et al. surveyed precision irrigation management using machine learning and digital farming solutions, documenting multiple IoT architectures that combine sensor networks, cloud platforms and ML models for irrigation scheduling and fertigation.

2.2 Machine Learning for Irrigation Scheduling

The literature reports diverse ML models for irrigation management, including artificial neural networks (ANNs), SVMs, decision trees, random forests, gradient boosting and fuzzy logic controllers. KNN appears across several studies as either a soil moisture predictor or as a classifier for irrigation timing. A systematic review notes systems where KNN uses discrete soil moisture "states" (dry to wet) and nearby sensor patterns to determine whether to trigger irrigation.

Ramya et al. proposed an "IoT framework for smart irrigation using machine learning technique" where sensor data is transmitted to a cloud backend, and ML models are trained to recommend irrigation actions. Abo-Zahhad designed an IoT-based automated irrigation management system using soil moisture data and weather forecasting, adopting ML techniques to optimize irrigation decisions under forecast uncertainty.

2.3 K-Nearest Neighbours Algorithm

KNN is a non-parametric, instance-based learning method. For a new observation, KNN finds the K training samples that are closest in feature space (based on a distance metric such as Euclidean distance) and predicts a label by majority vote (classification) or average (regression). KNN's simplicity makes it well-suited to embedded or edge deployments, particularly where datasets are moderate in size and the feature set is small and well-engineered.

Table 2. Selected ML approaches for smart irrigation from literature

Study / Year	**IoT Components**	**ML Method(s)**	**Role of ML**
Touati et al., 2013	WSN, data logger	Fuzzy logic	Valve control based on soil moisture
Ramya et al., 2020	Sensors + cloud backend	KNN, others	Smart irrigation decisions
Tace et al., 2022	Node-RED, MongoDB, sensors	KNN, SVM, NN, NB	Model comparison; KNN best accuracy
Abioye et al., 2022	Various architectures surveyed	Multiple ML models	Review of precision irrigation ML
Abo-Zahhad, 2023	IoT soil moisture + weather	Supervised ML	Automated irrigation management
Vallejo-Gómez et al., 2023	Multiple systems reviewed	ANN, KNN, RF, etc.	Systematic review of smart irrigation

These works motivate a focused design that explicitly builds IoT irrigation around KNN while comparing it rigorously with alternative ML models.

3. System Architecture

We consider a generic architecture for an IoT and ML-based irrigation system using KNN, comprising three primary layers:

1. **Sensing & Actuation Layer:** Field-deployed sensor nodes and actuators.
2. **Edge & Network Layer:** Gateway nodes that aggregate data, perform basic preprocessing and relay data to the cloud.
3. **Cloud & Application Layer:** Storage, ML services (including KNN), irrigation scheduling logic and user dashboards.

3.1 Sensing & Actuation Layer

Each field block is equipped with:

- Soil moisture sensors at multiple depths
- Temperature and humidity sensors
- Light or solar radiation sensors
- Rain gauge or rainfall proxy (optional)
- Solenoid valves controlling drip or sprinkler lines

Low-power microcontrollers (e.g., Arduino, ESP32) periodically read sensor values, timestamp them, and send packets via Wi-Fi, LoRa or cellular to an edge gateway.

3.2 Edge & Network Layer

The edge gateway (e.g., a Raspberry Pi) aggregates readings from multiple sensor nodes, performs basic data validation (range checks, missing value handling) and may execute lightweight analytics (e.g., rolling averages) to reduce noise. Data packets are forwarded to the cloud via REST APIs or MQTT.

3.3 Cloud & Application Layer

In the cloud, a time-series database stores sensor readings and derived features. A model service exposes a REST endpoint that runs the KNN classifier/regressor on the most recent window of features for each field block and returns an irrigation recommendation. A scheduler component translates these recommendations into actuation commands (open/close valves for specified durations), with safety constraints and farmer overrides.

A web or mobile dashboard displays:

- Current and historical sensor readings
- Irrigation recommendations and actions taken
- Alerts (e.g., sensor failure, abnormal patterns)

Table 3. Example hardware components for the proposed system

Layer	Component type	Example choice	Notes
Sensing	Soil moisture probe	Capacitive or TDR sensor	2–3 depths per plot
Sensing	Temp/humidity	DHT22 / SHT31	Protected enclosure
Sensing	Rain gauge (optional)	Tipping-bucket sensor	For rainfall-aware scheduling
Actuation	Valve control	12V DC solenoid valves	Controlled by relay board
Edge gateway	Compute + comms	Raspberry Pi / industrial gateway	Runs local services, buffering
Network	Connectivity	Wi-Fi / LoRa / 4G	Depends on field layout
Cloud	Storage & compute	Time-series DB + ML service	E.g., InfluxDB + containerized API

This layered design keeps field hardware simple while centralizing most of the ML complexity in the cloud or a powerful edge node.

4. Methodology: KNN-Based Irrigation Decision Model

4.1 Data and Feature Engineering

The KNN model operates on fixed-length feature vectors representing the recent state of the field and the near-term weather. In a typical setup, training data are collected over one or more growing seasons during which irrigation decisions are made either manually by experts or using a baseline scheduling strategy. These historical records provide labeled examples of “irrigate now” vs. “wait” vs. “no irrigation.”

Core features (per sample) might include:

- Current soil moisture at multiple depths
- Short-term moisture trends (e.g., 3-hour delta)
- Air temperature and relative humidity
- Solar radiation or light intensity
- Forecast rainfall (0–24 hours)
- Crop type and phenological stage
- Soil texture class
- Time of day and day of season

Features are normalized to comparable ranges to ensure Euclidean distance is meaningful. Categorical variables such as crop type are encoded (e.g., one-hot).

Table 4. Typical feature set for KNN-based irrigation decision

Feature group	**Example features**	**Rationale**
Soil moisture	θ_0–10cm, θ_{10}–20cm, $\Delta\theta$ (last 3 h)	Direct measure of water availability
Weather	Temp, RH, wind speed, solar radiation	Drives evapotranspiration
Forecast	Rainfall (0–6, 6–24 h)	Avoid irrigating before rain
Crop & soil	Crop type, growth stage, soil class	Different water requirements
Time context	Hour of day, day after planting	Captures diurnal & seasonal patterns

Labels can be defined either as discrete classes (e.g., 0 = no irrigation, 1 = irrigate, 2 = irrigate heavily) or as regression targets (e.g., mm of water to apply). In this paper we focus on a 3-class classification setup.

4.2 KNN Formulation

Let each training example be a pair (x_i, y_i), where $x_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{0, 1, 2\}$ is the irrigation class. For a new observation x, KNN proceeds as follows:

1. Compute the distance between x and each x_i.
2. Identify the K nearest neighbors $\mathcal{N}_K(x)$ based on distance.
3. Predict $\hat{y}$ as the majority class among neighbors:
$$\hat{y} = \arg\max_{c \in \{0,1,2\}} \sum_{(x_i,y_i) \in \mathcal{N}_K(x)} \mathbf{1}(y_i = c)$$

Weighted KNN variants can assign higher influence to closer neighbors using inverse distance weights. For our irrigation scenario, moderate K values (e.g., 5–15) often provide a good balance between robustness and local sensitivity.

4.3 Training and Evaluation Protocol

A typical evaluation protocol might use:

- **Train/validation/test split:** e.g., 60/20/20 across time, ensuring the test set corresponds to a later portion of the season or a subsequent season.
- **Cross-validation on training+validation:** To tune K and other hyperparameters (distance metric, weighting).
- **Baselines:**
  - Fixed-time irrigation schedule (rule-based)
  - Threshold-based moisture control (simple control rule)
  - Other ML models: logistic regression, SVM, random forest

Key metrics include overall accuracy, per-class F1-score, confusion matrices, water-use efficiency (WUE), and potential yield impact.

Table 5. Example evaluation protocol for comparative analysis

Item	**Configuration / choice**
Dataset span	2–3 seasons, multiple plots
Target	3-class irrigation decision

Item	Configuration / choice
Split strategy	Time-aware 60/20/20 (season-based)
Hyperparameters	K ∈ {3, 5, 7, 9, 11, 13}, distance: Euclidean
Baseline 1	Fixed time-of-day irrigation
Baseline 2	Moisture threshold rule
ML comparators	Logistic regression, SVM, random forest
Metrics	Accuracy, F1, WUE, decision latency

The next section provides an illustrative comparative analysis using this setup.

5. Results and Comparative Analysis

The exact numerical results depend on the specific farm, soil, crop and data. Here we present an illustrative scenario consistent with performance ranges reported in the literature for KNN-based irrigation and related ML approaches.

5.1 Evaluation Metrics

- **Accuracy:** Proportion of correctly predicted irrigation decisions.
- **Macro F1-score:** Average of F1 across the three classes.
- **Water-Use Efficiency (WUE):** Ratio of yield to water applied, normalized to a baseline.
- **Latency:** Average time from sensor data arrival to decision.
- **Complexity:** Qualitative implementation and maintenance complexity.

5.2 KNN Performance

In many smart irrigation studies, KNN has reached accuracy levels around 90–98% for classification tasks when trained on well-curated sensor datasets. In our illustrative scenario, after tuning K using validation data, K = 7 is selected.

KNN shows:

- High accuracy for “irrigate now” vs “wait” decisions when soil moisture and short-term trends are good predictors.
- Slight confusion between “wait” and “no irrigation” in edge cases, especially under unusual weather conditions.
- Stable performance across different plots as long as the feature distribution is similar to training data.

5.3 Comparison with Other ML Models

Table 6 summarizes a representative comparison among KNN and alternative models. Numbers are indicative, not tied to a single real dataset, but reflect relative behaviour commonly reported in literature.

Table 6. Illustrative comparison of ML models for irrigation decision

Model	**Accuracy (%)**	**Macro F1**	**Relative WUE gain vs. rule-based**	**Latency (ms)***	**Implementation complexity**	**Notes**
Rule-based schedule	~70	0.68	1.00× (baseline)	5–10	Very low	Fixed schedule, no adaptation
Moisture threshold	~78	0.75	1.08×	5–10	Low	Single-sensor threshold
Logistic regression	~86	0.84	1.15×	10–20	Low–medium	Linear decision boundary
SVM (RBF kernel)	~90	0.88	1.20×	50–100	Medium–high	Good accuracy, harder to tune
KNN (K = 7)	**92–95**	**0.90**	**1.22×**	20–60	Low–medium	Strong performance on tabular sensor data
Random forest	93–96	0.92	1.24×	30–80	Medium–high	Higher accuracy, more complex to deploy

*Latency for a single prediction on typical embedded or cloud setups.

From this comparative perspective:

- KNN achieves accuracy close to or slightly below random forests but is simpler to implement and interpret.
- Compared with SVM, KNN may be easier to tune since it requires only K and a distance metric.
- Logistic regression underperforms KNN when the decision boundary is non-linear, which is common given interactions among soil, weather and crop features.

5.4 Comparison with Rule-Based Irrigation

Compared to conventional rule-based or fixed-schedule irrigation, the KNN-based system can:

- Reduce unnecessary irrigation events when soil moisture remains sufficient, especially following unanticipated rainfall.
- Increase irrigation frequency during heat waves when evapotranspiration spikes.
- Adapt to differences between plots with varying soil textures and crop varieties without manually re-tuning rules.

These advantages translate into better WUE and more stable yields under variable climatic conditions.

Table 7. Indicative impact of decision strategy on water and yield

Strategy	**Water used (normalized)**	**Yield (normalized)**	**WUE (Yield / Water)**
Fixed schedule	1.00	1.00	1.00
Moisture threshold	0.88–0.92	1.00–1.02	1.10–1.15
KNN-based ML	**0.75–0.85**	**1.02–1.05**	**1.25–1.35**

The ranges in Table 7 are consistent with reported water savings (20–40%) and yield preservation or improvement in ML-assisted irrigation studies.

6. Discussion

6.1 Strengths of KNN for IoT-Based Irrigation

KNN offers several attractive properties for smart irrigation:

- **Simplicity:** No model training in the conventional sense; the "model" is the stored dataset. This reduces the barrier for deployment on resource-constrained platforms.
- **Local adaptivity:** Decisions are based on local neighborhoods in feature space, enabling the model to adapt to micro-climatic and soil differences captured in the data.
- **Interpretability:** It is straightforward to explain recommendations to farmers by showing similar past situations and their outcomes.
- **Incremental updates:** New labeled examples can be incorporated simply by adding them to the dataset.

6.2 Practical Considerations and Challenges

Despite these strengths, several practical issues must be addressed:

1. **Computational cost at scale:** KNN inference time grows with the size of the training set. This can become problematic as multi-season, multi-farm datasets grow. Approximate nearest neighbor indexing or prototype selection techniques can mitigate this.
2. **Data quality and sensor reliability:** Faulty sensors, communication glitches and calibration drift can introduce noise and outliers, which directly degrade KNN performance because of its reliance on raw distances. Robust preprocessing, outlier detection and redundancy in sensing are important.
3. **Domain shift:** If a KNN model trained on one farm or season is deployed in a significantly different environment (new soil types, crop varieties, or climate), performance may degrade. Periodic re-collection of local labeled data and domain adaptation strategies are required.
4. **Label acquisition:** Supervised training requires ground-truth labels (e.g., expert decisions or measured optimal irrigation). Capturing these labels consistently can be challenging, especially for smallholder farms.

Table 8. Strengths and limitations of KNN-based irrigation

Aspect	KNN strengths	KNN limitations
Implementation	Easy to implement, minimal tuning	Needs efficient indexing for large datasets
Data requirements	Works well with small–medium datasets	Sensitive to noise and irrelevant features
Performance	Competitive accuracy, good WUE gains	Latency increases with dataset size
Usability	Transparent, example-based reasoning	Harder to generalize across very different sites

6.3 Future Extensions

Potential directions to enhance the proposed system include:

- **Hybrid models:** Combining KNN with deep learning for feature extraction (e.g., using LSTM to capture temporal patterns and KNN for final decision) in line with emerging smart irrigation architectures that mix multiple ML techniques.
- **Federated and collaborative learning:** Sharing model insights across farms without centralizing raw data, particularly useful where data privacy or connectivity limit centralized training.
- **Explainable decision support:** Visual tools that show similar historical states and outcomes to build farmer trust.
- **Integration with broader farm management:** Coupling irrigation decisions with fertigation, disease risk prediction and energy management for pump operations.

7. Conclusion

This paper presented an IoT and ML-based irrigation system centered on the KNN algorithm. We reviewed current literature on smart irrigation, highlighting the growing yet still under-exploited role of KNN in irrigation scheduling. We then outlined a layered system architecture connecting field sensors, edge gateways and a cloud-hosted KNN decision service. A detailed methodology for feature engineering, KNN formulation and evaluation was provided, followed by an illustrative comparative analysis.

The comparative results, aligned with trends in existing studies, suggest that KNN:

- Achieves accuracy and water-use efficiency comparable to or better than more complex models in many scenarios.
- Is easier to implement and update than deep neural networks or sophisticated ensembles.
- Provides transparent, example-based reasoning that can improve farmer acceptance and trust.

At the same time, KNN's dependence on high-quality labeled data and its computational scaling challenges mean that careful system design, robust preprocessing and incremental model management are essential. Future work should explore hybrid and federated approaches that retain KNN's interpretability while improving scalability and robustness across diverse agro-ecosystems.

References

1. Abioye, E. A., Hensel, O., Esau, T. J., Elijah, O., Abidin, M. S. Z., Ayobami, A. S., Yerima, O., & Nasirahmadi, A. (2022). Precision irrigation management using machine learning and digital farming solutions. *AgriEngineering, 4*(1), 70–103. https://doi.org/10.3390/agriengineering4010006
2. Abo-Zahhad, M. M. (2023). IoT-based automated management irrigation system using soil moisture data and weather forecasting adopting machine learning technique. *Sohag Engineering Journal, 3*(2), 122–140. https://doi.org/10.21608/sej.2023.209528.1037
3. Ramya, S., Swetha, A. M., & Doraipandian, M. (2020). IoT framework for smart irrigation using machine learning technique. *Journal of Computer Science, 16*(3), 355–363. https://doi.org/10.3844/jcssp.2020.355.363
4. Tace, Y., Tabaa, M., Elfilali, S., Leghris, C., Bensag, H., & Renault, E. (2022). Smart irrigation system based on IoT and machine learning. *Energy Reports, 8*, 10263–10276. https://doi.org/10.1016/j.egyr.2022.07.088
5. Touati, F., Al-Hitmi, M., Benhmed, K., & Tabish, R. (2013). A fuzzy logic based irrigation system enhanced with wireless data logging applied to the state of Qatar. *Computers and Electronics in Agriculture, 98*, 233–241. https://doi.org/10.1016/j.compag.2013.08.018
6. Vallejo-Gómez, D., Osorio, M., & Hincapié, C. A. (2023). Smart irrigation systems in agriculture: A systematic review. *Agronomy, 13*(2), 342. https://doi.org/10.3390/agronomy13020342
7. Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine, 4*(11), 218. https://doi.org/10.21037/atm.2016.03.37