Received: February 25, 2025. Revised: April 10, 2025.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Deep Learning Architecture for Irrigation Prediction Using ID-CNN-Bidirectional GRU- Fusion Model

Fathima M. S[1]* **B. Booba[2]**

[1]*Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies, Chennai, India*
[2]*Department of Applied computing and Emerging Technologies,*
Vels Institute of Science, Technology & Advanced Studies, Chennai, India
* Corresponding author's Email: haqulfathima@gmail.com

Abstract: Irrigation management plays a crucial role in sustainable agriculture by optimizing water usage and minimizing resource wastage, especially in the face of global challenges like water scarcity and climate change. Traditional irrigation prediction models often lead to inefficient water use, either by under-irrigating or over-irrigating, thus affecting crop yield. This study proposes a novel hybrid deep learning (DL) model that combines a one-dimensional convolutional neural network (1D-CNN) and Bidirectional Gated Recurrent Units (Bi-GRU) to enhance irrigation prediction accuracy. The 1D-CNN excels at spatial feature extraction, enabling the model to detect localized patterns in environmental and soil moisture data, while the Bi-GRU captures temporal dependencies by processing sequential data in both forward and backward directions. This hybrid approach addresses the shortcomings of conventional models by effectively learning both spatial and temporal relationships within the data, leading to more accurate and adaptive irrigation predictions. The model is trained and evaluated using the Irrigation Scheduling for Smart Agriculture dataset, which includes various environmental and soil moisture parameters. The proposed hybrid model achieved an accuracy of 97.29%, outperforming traditional models and demonstrating its potential to optimize irrigation management. This study presents a scalable and adaptive solution for intelligent irrigation systems, offering a promising approach to reduce water wastage, enhance crop yield, and contribute to sustainable agricultural practices. The novel combination of CNN and Bi-GRU provides a significant advancement over existing techniques, making it a valuable contribution to the field of smart agriculture.

Keywords: Irrigation prediction, Convolutional layer, Bidirectional GRU, Deep learning, Soil moisture prediction.

1. Introduction

Water is one of the most essential natural resources that sustain life on Earth. It is indispensable for various human activities, with agriculture being the most significant consumer of freshwater resources. However, the availability of water is often limited and expensive, particularly in semi-arid and arid regions, where rainfall is sparse and inconsistent. In these regions, agricultural productivity heavily depends on irrigation to meet the water requirements of crops [1]. The growing global population and ongoing climatic changes have led to a significant surge in water demand, especially for agricultural purposes. The increase in food production requirements has placed immense pressure on existing water resources, necessitating efficient management strategies to ensure sustainable water use.

Water plays a crucial role in crop development and overall agricultural productivity. It is vital for various physiological processes such as photosynthesis, nutrient uptake, and temperature regulation in plants [2]. Without an adequate supply of water, crops fail to reach their full potential, leading to reduced yields and food shortages. Given the unpredictability of natural precipitation in many agricultural regions, irrigation is employed as a controlled means of delivering water to crops. Farmers rely on irrigation systems to supplement inadequate or inconsistent rainfall, ensuring that plants receive the necessary hydration for growth [3]. By maintaining optimal soil moisture levels, irrigation helps sustain crop health throughout the growing cycle and prevents the negative impacts of water stress, such as wilting, stunted growth, and reduced productivity.

In modern agriculture, irrigation management has evolved into a critical aspect of farming due to the increasing demand for food and the growing concerns over water scarcity [4]. The prediction of irrigation requirements plays a pivotal role in optimizing water consumption, enhancing crop yield, and promoting environmental conservation. Effective irrigation prediction enables farmers to apply the right amount of water at the right time, preventing the adverse effects of both over-irrigation and under-irrigation. Traditional irrigation practices, which often rely on fixed schedules or manual observations, are prone to inefficiencies. Over-irrigation results in excessive water wastage, leaching of essential nutrients from the soil, and increased risk of soil erosion. Conversely, under-irrigation leads to inadequate water supply, which negatively impacts plant growth and reduces crop yields [5]. To overcome these challenges, modern irrigation management integrates advanced technologies such as data analytics, weather forecasting, artificial intelligence (AI), and machine learning (ML) to accurately predict irrigation requirements [6].

Irrigation prediction is based on several key factors that influence water usage, including soil moisture levels, crop type, weather conditions, and the growth stage of plants. These factors are analyzed using advanced computational models that process large datasets to generate precise predictions. By leveraging these insights, farmers can make informed decisions regarding when and how much water to apply to their fields, ensuring that water resources are used efficiently. The integration of predictive models into irrigation systems enhances the accuracy of water distribution, minimizes waste, and reduces dependency on guesswork. This data-driven approach not only improves agricultural productivity but also contributes to the sustainability of water resources, addressing the long-term challenges of water scarcity and climate variability.

Beyond its environmental benefits, precise irrigation prediction offers substantial economic advantages. In regions where agriculture relies heavily on irrigation, the cost of water can be a significant burden on farmers. By minimizing water wastage, predictive irrigation management helps reduce operational costs associated with water procurement, pumping, and distribution. The efficient use of water translates into lower energy consumption, as less power is required to operate irrigation systems. This contributes to overall cost savings for farmers and promotes economic sustainability in agricultural practices [7]. Furthermore, optimized irrigation strategies improve the resilience of farming operations to climate change. As global temperatures rise and extreme weather events become more frequent, farmers must adapt to fluctuating environmental conditions to maintain crop productivity. Irrigation prediction allows for adaptive water management, helping farmers mitigate the effects of prolonged droughts, heatwaves, and irregular rainfall patterns [8]. By ensuring that crops receive an adequate and timely water supply, predictive irrigation enhances plant health, boosts yield, and maintains food security.

Another critical aspect of irrigation prediction is its role in maintaining soil health and long-term agricultural sustainability. Efficient water management prevents the degradation of soil structure and reduces the risks associated with excessive irrigation, such as soil salinization. When irrigation is applied excessively, the excess water percolates through the soil, dissolving and transporting salts to the surface. Over time, this leads to an accumulation of salts in the root zone, making it difficult for plants to absorb water and nutrients. By implementing precise irrigation techniques, farmers can maintain a balanced soil moisture level, preserving soil fertility and preventing land degradation. This not only benefits current crop cycles but also ensures that the land remains arable for future agricultural activities.

The integration of artificial intelligence (AI) into irrigation prediction has revolutionized the way water resources are managed in agriculture. AI-powered models analyze complex datasets, identify patterns, and generate highly accurate irrigation forecasts. These models utilize deep learning algorithms to process real-time data from various sources, such as satellite imagery, weather stations, soil sensors, and historical irrigation records. By continuously learning and adapting to changing environmental conditions, AI-driven irrigation prediction systems enhance decision-making and improve water use efficiency.

This study introduces a novel hybrid deep learning framework designed to leverage the strengths of both spatial and temporal analysis. The model integrates a 1D-CNN with Bi-GRU to form a synergistic architecture tailored for irrigation volume prediction. The 1D-CNN is particularly suited to identifying localized spatial patterns in multivariate input features. In agricultural settings, parameters such as soil moisture, temperature, and evapotranspiration often vary spatially across different locations or depths. By applying convolutional filters, the 1D-CNN effectively extracts these spatial dependencies, which are critical for detecting subtle environmental cues that inform irrigation needs.

To complement this, the Bi-GRU component captures temporal dynamics by processing input sequences in both forward and backward directions. This bidirectional learning allows the model to utilize both historical and forward-looking context, which is essential in understanding irrigation trends influenced by weather cycles, crop stages, and seasonal shifts. The GRU architecture also provides computational efficiency compared to LSTM models, making it suitable for real-time smart agriculture applications. By combining these two components, the model is capable of learning rich spatial-temporal representations from complex agricultural datasets. This integrated approach directly addresses the dual-dimensional nature of irrigation prediction and significantly enhances forecasting accuracy and system adaptability. The key contributions of the study are given below.

- To propose a novel hybrid DL model for irrigation prediction.
- To develop an intelligent irrigation prediction model that adapts to different environmental conditions.
- To assess the proposed model using different evaluation metrics.
- To compare the efficacy of the suggested model with conventional methods.

The remaining portion of the study is structured as: Section 2 gives a comprehensive review of existing studies in irrigation prediction and water management. Section 3 shows the proposed design of the hybrid DL method and Section 4 assesses the effectiveness of the suggested method. Finally, Section 5 concludes the key findings of the study.

2. Literature review

Sheline et al. [9] developed the Predictive Optimal Water and Energy Irrigation (POWEIr) controller, a precision irrigation system for solar-powered drip irrigation (SPDI) aimed at optimizing water and energy usage. The POWEIr controller utilized physics-based models and machine learning for crop water demand and energy predictions, improving irrigation scheduling and reducing overall system costs. The study validated the controller with a small-scale prototype tested in Cambridge, Massachusetts, over seven days. Results indicated that the POWEIr controller increased solar irrigation reliability by up to 46% on high water demand days while using six times less battery storage compared to conventional SPDI systems. Additionally, an economic analysis demonstrated that improved solar energy efficiency could save 18%–74% in solar pump lifetime costs while ensuring 31%–66% more reliable irrigation. A limitation of the study was the adaptation challenges when applying the POWEIr controller to different irrigation systems and variable power inputs.

Youssef et al. [10] investigated the use of machine learning (ML) algorithms to predict reference evapotranspiration (ETO), which is critical for optimizing irrigation management in the face of climate change. The authors employed three ETO calculation methods: Penman-Monteith (PM), Hargreaves (HA), and Blaney-Criddle (BC), and analyzed climate variables using the modified Mann-Kendall test and Theil Sen’s slope estimator to identify trends. Several ML models, including Support Vector Regression (SVR), Random Forest (RF), XGBoost, K-Nearest Neighbor (KNN), Decision Trees (DT), Linear Regression (LR), and Multiple Linear Regression (MLR), were applied for ETO prediction. The results showed strong predictive performance, with R^2 values ranging from 0.91 to 0.99, and low Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values. While the study demonstrated that ML algorithms can effectively predict ETO, a limitation lies in its application across varying climatic regions. The models' performance could fluctuate when applied to diverse geographical areas with distinct climate patterns.

Yan et al. [11] proposed an irrigation prediction method that integrates a bidirectional Long Short-Term Memory (BiLSTM) model, Convolutional Neural Networks (CNN), and an attention mechanism. Their approach aimed to improve irrigation volume prediction by leveraging spatio-temporal features and sequence dependencies within crop irrigation data. The study utilized historical irrigation data and meteorological variables such as temperature, precipitation, and wind speed, to train and test the BiLSTM-CNN-Attention model. The study achieved superior performance with an R^2 value of 0.9749. The study possessed challenges in parameter tuning and was limited to broader applicability testing across different contexts. The study also acknowledged that the increased complexity of the combined model structure, which integrates multiple components, could lead to computational inefficiencies and required extensive fine-tuning to optimize performance.

Kumar et al. [12] proposed an IoT-based sensor-integrated intelligent irrigation system for the agriculture industry, utilizing IoT-based humidity and soil sensors to collect soil-related data, which is stored in a centralized cloud. The authors employed the Correlation-based Feature Selection (CFS) algorithm to select relevant features and discarded irrelevant data. They applied the K-means algorithm for clustering the data, helping to group similar data together. The classification model was built using SVM, Random Forest, and Naïve Bayes algorithms, which were trained, validated, and tested using the collected data, including historical soil and humidity data. The outcomes demonstrated that the SVM hybrid classifier achieved superior accuracy. The hybrid classifier was also successful in predicting water demand, helping to save up to 20% of fresh water through intelligent irrigation. However, the study faced challenges related to scalability and long-term performance across different agricultural practices. These limitations were particularly evident in larger-scale implementations, where variations in soil conditions, sensor reliability, and environmental factors could affect the model's generalization. Additionally, the dependency on centralized cloud storage could pose risks related to data security and real-time processing for large agricultural areas.

Ndunagu et al. [13] proposed a smart irrigation system (SIS) using the drip method, integrating wireless sensor networks and an IoT platform (ThingSpeak) for data collection, analytics, and visualization. The system utilized soil sensors, weather forecasts, and real-time data updates every 15 minutes to automate irrigation based on predefined thresholds. Machine learning models processed CSV-formatted data containing over 143,000 entries, achieving 89% accuracy, 79% sensitivity, and 93% precision. The ESP8266 NodeMCU was used for efficient control, while RTC modules-maintained timekeeping. The prototype was tested on a small vegetable farm and successfully optimized irrigation efficiency and crop yield. However, limitations included reliance on stable internet and power sources, restricted scalability.

Singh et al. [14] proposed a deep learning approach to improve sprinkler irrigation by predicting the optimal irrigation time based on in-field soil moisture. The authors used a CNN integrated with depth-wise separable convolution and residual connections to predict soil moisture classes from in-field soil images. A mobile application was developed to estimate irrigation time by analyzing soil moisture, crop factors, and sprinkler system details. The CNN model achieved an impressive average classification accuracy of 97.10%, with high precision (85.50%) and recall (86.80%), leading to water and energy savings of 27.59% and 27.42%, respectively. The system was tested in an experimental field in Meghalaya, India, and demonstrated increased water productivity (32.75%) compared to conventional systems. However, the study faced challenges in predicting soil moisture due to horizontal heterogeneity in soil, which impacted model accuracy in certain conditions. Factors such as soil texture and structure occasionally resulted in misclassification, further limiting the model's performance.

Suresh et al. [15] proposed an IoT-enabled deep learning-based smart irrigation system (IoTDL-SIS) to optimize water usage in precision agriculture with minimal human intervention. The system uses various sensors such as soil moisture, temperature, air temperature, and humidity, which send data to an Arduino module for transmission to a cloud server. Data analysis at the cloud level is performed using three methods: deep support vector machine (DSVM) regression, clustering, and artificial immune optimization algorithm (AIOA) with a deep belief network (DBN) model for binary classification. Their experimental results demonstrated the effectiveness of IoTDL-SIS, achieving high accuracy (0.971) in predicting irrigation needs and reducing water usage. The system's performance was superior to existing methods, with precision, recall, and accuracy all above 0.96 in various test runs. While the IoTDL-SIS showed promising results in various test environments, its performance in more expansive agricultural settings or under diverse conditions remains unclear. Furthermore, the reliance on cloud servers for data analysis could introduce latency, potentially hindering real-time decision-making, particularly in remote agricultural areas where connectivity might be limited.

Alibabaei et al. [16] proposed DL models, like LSTM, Bi-LSTM and GRU, to forecast the agricultural yield from moisture content of the soil, irrigation prediction and climate data. The Bi-LSTM model surpassed others with an R^2 value of 0.97 to 0.99. However, the study had notable limitations. One key limitation was the absence of efficient preprocessing methods, which could have enhanced the model’s performance by improving the quality of the input data and reducing noise. Additionally, the training process for these models was time-intensive, particularly with large datasets, leading to increased computational costs and longer model convergence times.

Abuzanouneh et al. [17] suggested an IoT-ML based smart irrigation system (IoTML-SIS) for agriculture. Data were gathered using IoT sensors, which was achieved by the artificial algae algorithm to enhance the classification accuracy by least square-SVM (LS-SVM). The results demonstrated the efficacy of the system by attaining superior accuracy. However, a significant limitation of the study was its challenge in real-time application, particularly regarding the system's ability to handle dynamic environmental conditions. While the model performed well under controlled conditions, its scalability and adaptability in real-world, real-time irrigation settings remain uncertain.

Yonbawi et al. [18] improved intelligent agriculture by proposing a modified black widow optimization with a DBN based smart irrigation system (MBWODBN-SIS). Data were collected using IoT sensors and irrigation classes are classified into five categories using DBN and hyperparameter optimization was done using the MBWO algorithm. The outcomes demonstrated that, with 95.73% accuracy, this model outperformed other DL models. The study lacked efficient feature selection models, which limits its ability to identify the most relevant features for the system's performance.

Xu et al. [19] examined the soil moisture level and improved the irrigation system by integrating IoT with adaptive DL models. An Adaptive Hybrid Convolution-based ShuffleNetV2 (AHC-ShuffleNetV2) was employed for irrigation level prediction. Both 1D and 2D convolution layers are used to process the collected sensor data and crop field images. Piranha Foraging Optimization Algorithm (FPFOA) was used for optimization and the model achieved lower RMSE value. A notable limitation of the study was its inability to predict essential weather parameters such as temperature, UV rays, humidity, and wind, which are crucial for accurate crop field management and irrigation decisions. Benameur et al. [20] developed an intelligent irrigation system that identify the anomalies in sensor data by employing autoencoders (AE) and generative adversarial networks (GANs). The detected anomalies were replaced with the reconstructed output. The results demonstrated that the AE model outperformed GAN by 97% accuracy in detecting the air humidity. The study lacked autonomous calibration and remote management techniques, which are essential for improving sensor accuracy and ensuring the long-term sustainability of the system in real-world applications.

Kaur et al. [21] suggested an automated irrigation system with IoT and ML. The system was designed to automatically control irrigation by activating the motor when soil moisture dropped below a certain threshold. The collected data was analysed using ML models and demonstrated that RF and naïve bayes model achieved higher accuracy. The study's limitation lies in the complexity of sensor-based irrigation systems, which can be affected by sensor calibration, environmental variability, and the high costs associated with implementing a fully automated system. Moreover, the study did not consider the variability in soil properties, which could affect the system’s overall performance. Dhal et al. [22] introduced a Decision Support System (DSS) based on ML aimed at optimizing irrigation of turfgrass while reducing runoff. The system used a Radial Basis Function-SVM (RBF-SVM) model with artificial data produced by the Monte-Carlo method. When evaluated alongside a commercial irrigation controller, by 74% on average the system reduced runoff and maintained a high Green Cover (GC) with 87% accuracy. The study demonstrated that the ML-based irrigation systems optimized water usage and minimized environmental impact. The use of artificial data produced by the Monte Carlo method limits the generalization of the results, as the system's performance in real-world, variable environmental conditions was not fully addressed. Therefore, the model's applicability across diverse ecological contexts remains uncertain.

Rathore & Rajavat [23] employed ML models for irrigation water requirement prediction and real time disease detection. Three CNNs were employed for classify potato leaf disease and SVM and logistic regression (LR) model were used for real-time irrigation water requirement prediction. Parameters like humidity, soil moisture, temperature and crop age were measured as the factors to determine whether the water pump should be enabled or not. The SVM achieved 92% accuracy, while the LR model achieved 73%. However, the study faced limitations related to scalability, as it struggled to handle larger datasets and implement the models in real-time applications. This issue arises due to the inherent computational complexity of the employed models, especially when dealing with varying environmental factors and real-time data influx.

Despite significant advancements in smart irrigation systems, several research gaps remain. Many studies have demonstrated high accuracy in irrigation classification and prediction using ML and DL models, such as DBN and SVM, but challenges persist in scalability and adaptability to diverse agricultural practices [12]. Furthermore, while IoT-based solutions have been widely adopted, limitations in real-time application and broader context testing hinder their practicality [17]. Additionally, the lack of efficient preprocessing methods and feature selection models in several approaches impacts the optimization and performance of these systems [16,18]. Addressing

Received: February 25, 2025. Revised: April 10, 2025.

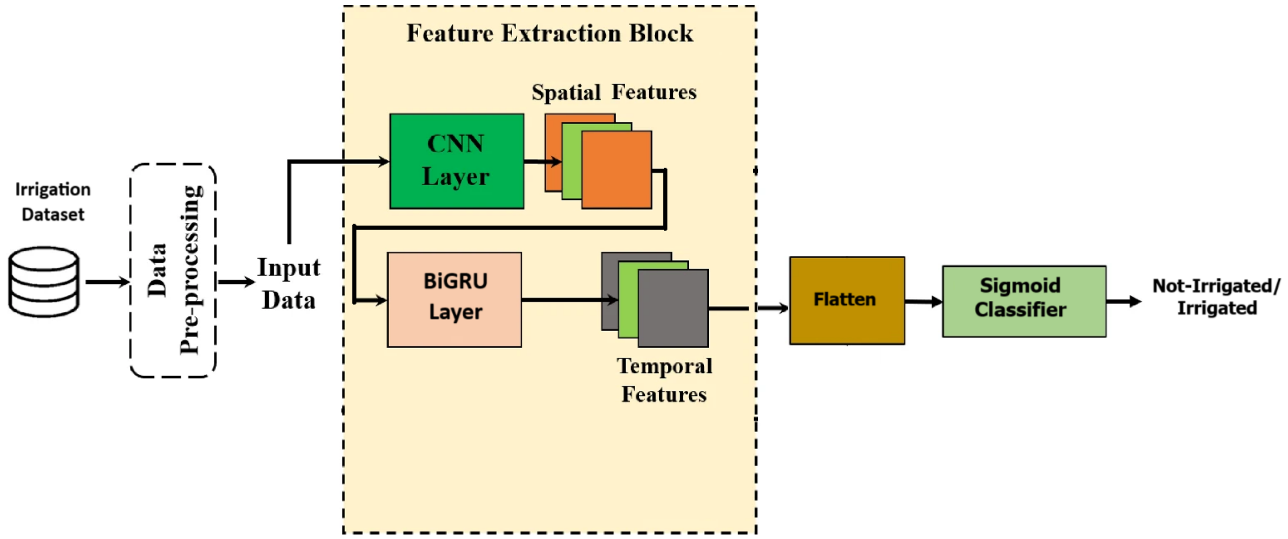

Figure.1 Suggested irrigation prediction method

these gaps could enhance the efficiency and reliability of smart irrigation technologies in varied and dynamic agricultural environments.

3. Proposed methodology

Irrigation prediction is crucial for sustainable agricultural practices by optimizing water usage. Accurate irrigation prediction enables farmers to determine the optimal water quality required for crops by reducing water loss and soil erosion. This study develops a novel DL model using 1D-CNN and Bi-GRU to address issues such as over-irrigation and under-irrigation. The spatial and temporal features are extracted from the preprocessed irrigation data. The workflow of the suggested model is illustrated in Fig. 1.

3.1 Dataset

The Irrigation Scheduling for Smart Agriculture dataset from the public repository, Kaggle serves as a prominent benchmark for optimized irrigation decisions in sustainable agricultural practices, considering environmental factors and soil moisture [24]. The dataset was made publicly available on Kaggle approximately 2 years ago. The dataset includes several thousand entries, each corresponding to a time-stamped observation across the mentioned features. Its tabular format allows for straightforward preprocessing and integration with deep learning models. This dataset serves as a credible and practical benchmark for developing intelligent irrigation systems, particularly in sustainable agriculture where environmental and soil dynamics play a crucial role in water resource optimization. The dataset captures a diverse array of environmental parameters critical for irrigation decisions.

One key feature is ‘temperature’, which represents the environmental temperature during data collection. This influences evaporation and transpiration rates, with higher temperatures typically increasing irrigation needs. Temperature values range from -10°C to 50°C, depending on geographical and climatic conditions. Another factor is ‘atmospheric pressure’ that influences weather conditions. Overall climate and its effect on plant health are assessed using pressure. Recorded pressure values vary between 900 hPa and 1050 hPa, relative to the standard atmospheric pressure (1013 hPa). The height of the place where the data is collected above sea level is referred to as altitude. This factor correlates with temperature and pressure variability, offering a broad representation of environmental conditions from sea level to mountainous terrain. Higher altitude locations generally exhibit cooler temperatures and lower atmospheric pressure.

The quantity of water present in the soil is indicated by the factor ‘soil moisture’, which is crucial for plant health. Low soil moisture levels indicate dryness and necessitate irrigation, while high soil moisture signifies adequate wetness. This factor is represented in percentage, where 0% represents completely dry soil and 100% represents saturated soil. This aids in determining the irrigation needs grounded on the current state of water content in the soil. The parameter ‘notes’ provides additional context or observations recorded during data collection. This includes weather conditions, unique plant requirements or other factors affecting irrigation decisions. The observations, such as ‘high wind’ or ‘overcast’, explain the abnormalities or

Received: February 25, 2025. Revised: April 10, 2025.

	temperature	pressure	altitude	soilmiosture	note	status
2979	28.60	9953.83	-14.79	165	3	1
2214	29.30	9969.64	-13.46	326	1	0
3894	29.44	9928.36	-16.93	249	2	1
2879	29.23	9957.51	-14.48	236	2	1
1158	29.59	9985.00	-12.17	175	3	1
519	30.96	9971.78	-13.28	358	0	0
1246	28.32	9984.09	-12.24	175	3	1
1204	28.74	9985.28	-12.14	174	3	1
4362	31.74	9928.77	-16.90	342	0	1
4470	29.20	9928.92	-16.89	252	2	0

Figure.2 Data sample

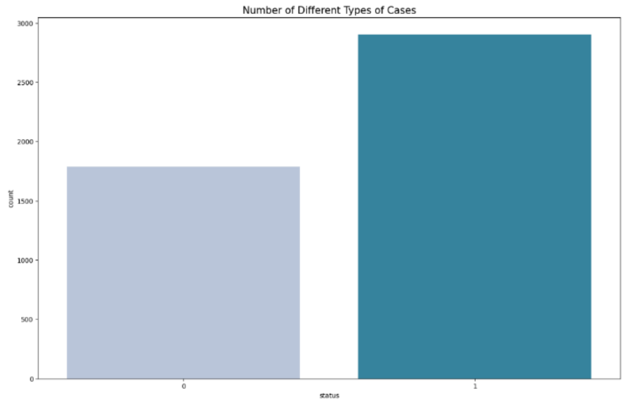

Figure.3 Distribution of case types

```
Summary Statistics:
       temperature      pressure     altitude  soilmiosture         note  \
count  4688.000000   4688.000000  4688.000000   4688.000000  4688.000000
mean     29.599089   9963.153215   -14.293590    243.692406     1.878413
std       5.842685   1383.602527     2.649662     76.176855     1.152977
min      27.970000  -2120.400000   -17.610000   -243.000000     0.000000
25%      28.630000   9935.255000   -16.350000    171.000000     1.000000
50%      29.180000   9969.535000   -13.470000    233.000000     2.000000
75%      29.990000   9975.700000   -12.950000    326.000000     3.000000
max     178.700000  99931.100000   116.700000    480.000000     3.000000

            status
count  4688.000000
mean      0.619027
std       0.485678
min       0.000000
25%       0.000000
50%       1.000000
75%       1.000000
max       1.000000

Missing Values:
temperature     0
pressure        0
altitude        0
soilmiosture    0
note            0
status          0
dtype: int64
```

Figure.4 Statistical summary

irregularities in the data. Finally, the ‘status’ of irrigation indicates whether irrigation was applied at the particular instance. The status is represented by binary values, where ‘0’ denotes ‘Not Irrigated’ and ‘1’ means ‘Irrigated’. Fig. 2 demonstrates the data sample.

Fig. 3 illustrates a bar plot that represents the distribution of binary status, which indicates whether the irrigation is applied or not, with labels ‘1’ and ‘0’ respectively. There are more cases of irrigated status compared to non-irrigated cases. This observation indicates that the majority of instances in the dataset required irrigation.

3.2 Data Preprocessing and Exploratory Data Analysis

The process of assessing, filtering, manipulating and encoding data, which converts it into machine-readable format, is called data preprocessing. Several preprocessing steps, like handling outliers and missing values, handling categorical variables and scaling of numerical features are involved. Normalization is employed to prevent bias towards large-scale features. Eq. (1) indicates the standardization that scales each feature to have standard deviation (σ) of one and zero mean (μ).

$$x' = \frac{x-\mu}{\sigma} \tag{1}$$

where the scaled feature value is represented by x'. The relationship between variables is analysed using the exploratory data analysis (EDA) by employing visualization and statistical metrics. The statistics summary, as shown in Fig. 4, indicates no missing values by offering a quantitative overview. The varying temperature, pressure and altitude provide different environmental and geographical conditions. Soil moisture also exhibits substantial variations.

Fig. 5 depicts the histogram of features, showing distribution in the dataset. A skewed distribution is provided by the variable ‘temperature’, having values ranging between 250C and 500C. This value indicates suitable agricultural conditions and higher temperature necessitates irrigation due to evaporation and transpiration. The factor, ‘pressure’ is distributed around the standard atmospheric pressure (1013 hPa), while altitude is highly skewed with most data near sea level. The distribution of soil moisture shows multiple peaks, representing different levels of water content in the soil from dry to saturated. The observed categorical contents are reflected by the distribution of ‘note’.

The heat map of the data, as illustrated in Fig. 6 reveals the relationship between the features in the dataset, represented by positive and negative values, where a positive value shows a positive relationship and a negative value specifies a negative relationship. Both linear and non-linear relationships are evaluated using the correlation heatmap. Features such as temperature and pressure validate a strong positive correlation, while soil moisture displays lower correlation with other features and is independently influenced by the irrigation status.

 DOI: 10.22266/ijies2025.0630.23

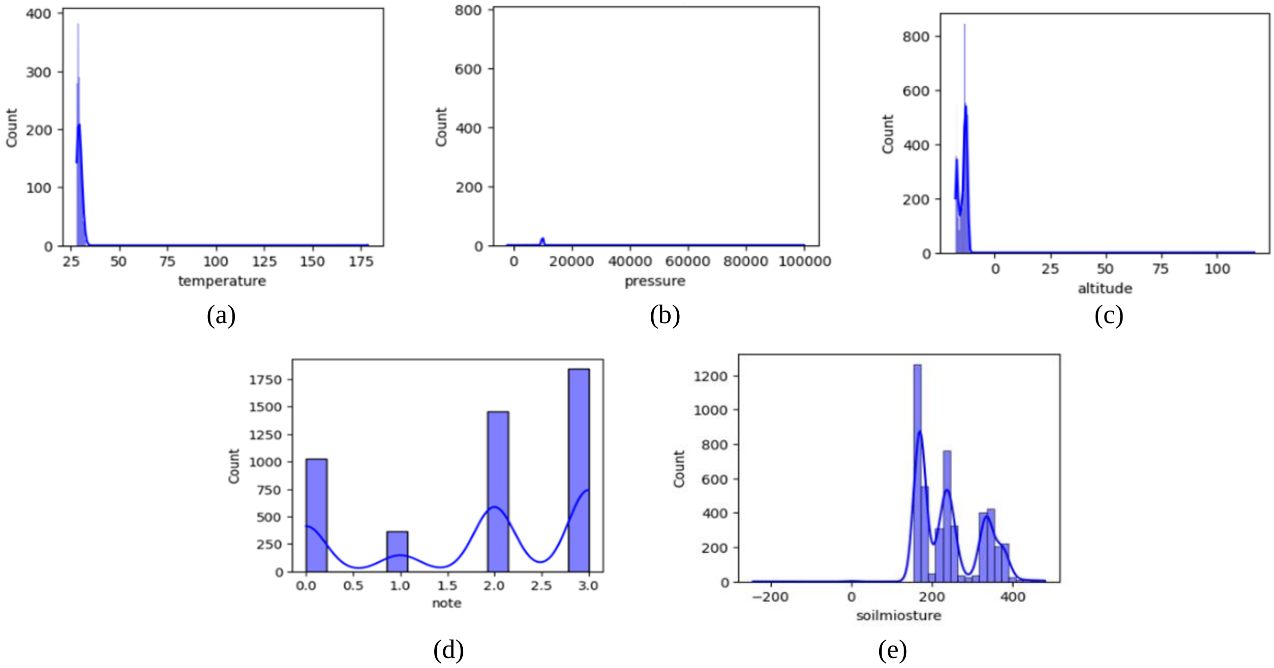

Figure.5 Distribution of features: (a) Distribution of temperature, (b) Distribution of pressure, (c) Distribution of altitude, (d) Distribution of note, and (e) Distribution of soilmiosture

Figure. 6 Correlation heatmap

Received: February 25, 2025. Revised: April 10, 2025.

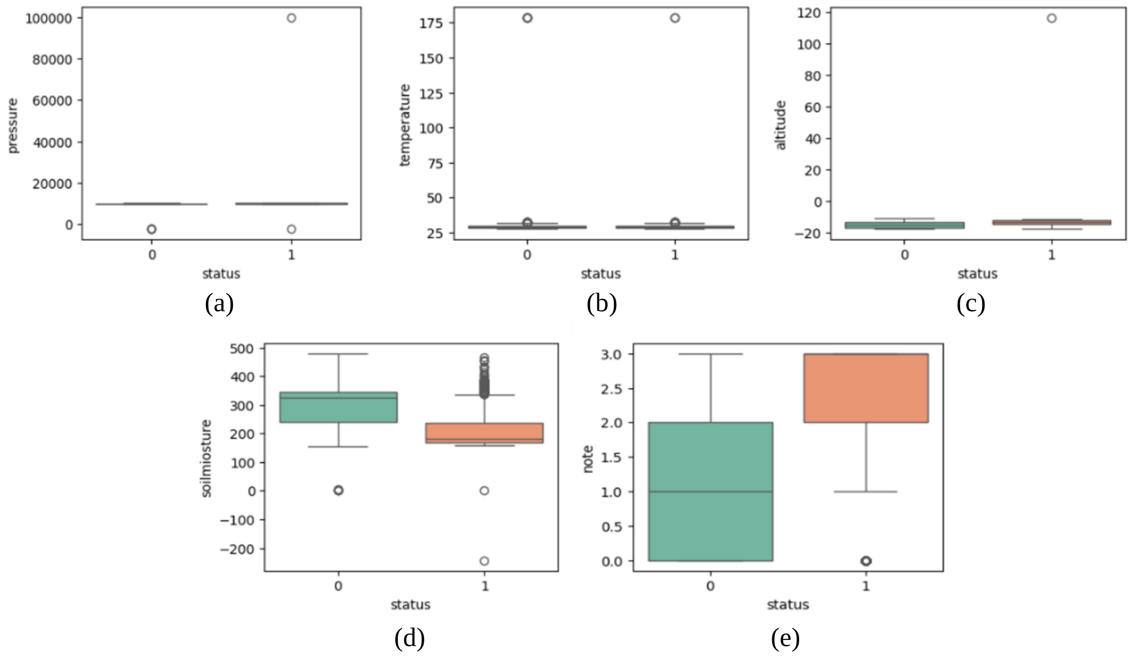

Figure.7 Correlation between target variable and the features: (a) temperature vs status, (b) pressure vs status, (c) altitude vs status, (d) soilmiosture vs status, and (e) note vs status

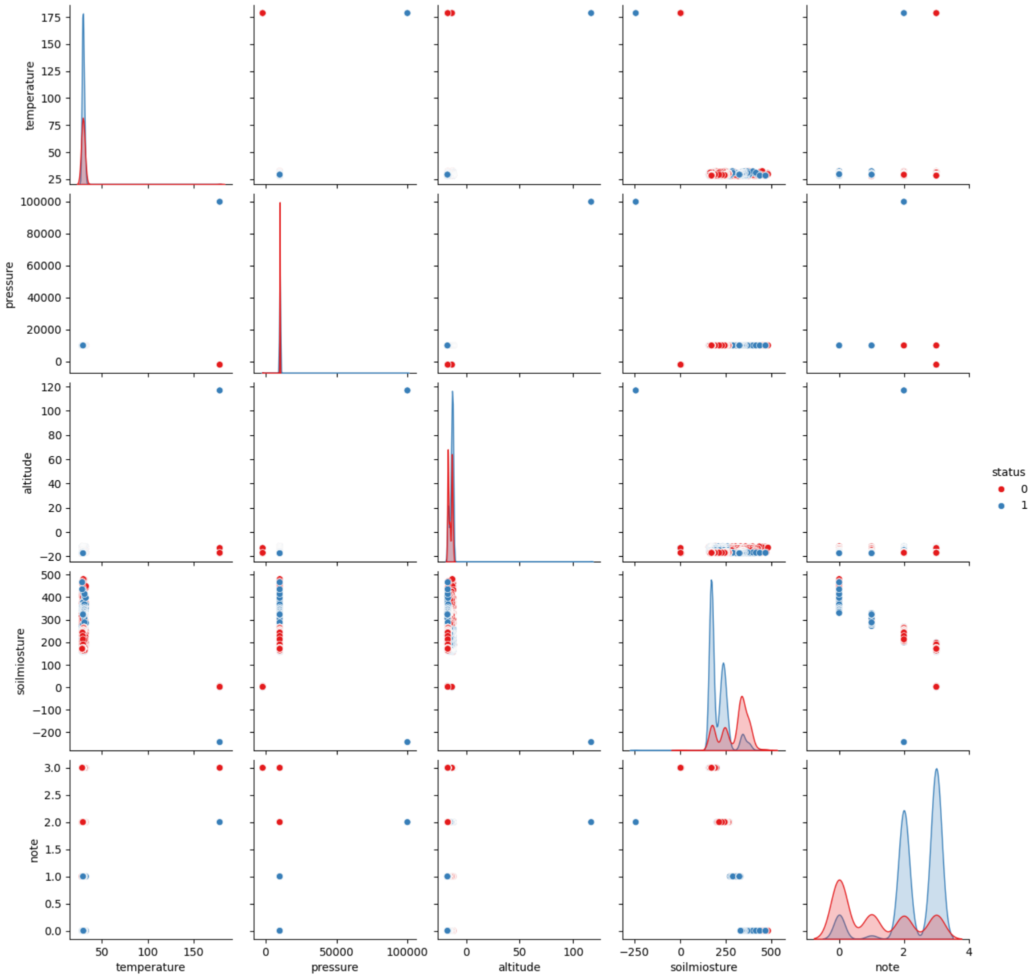

Figure.8 Scatter matrix visualization

 DOI: 10.22266/ijies2025.0630.23

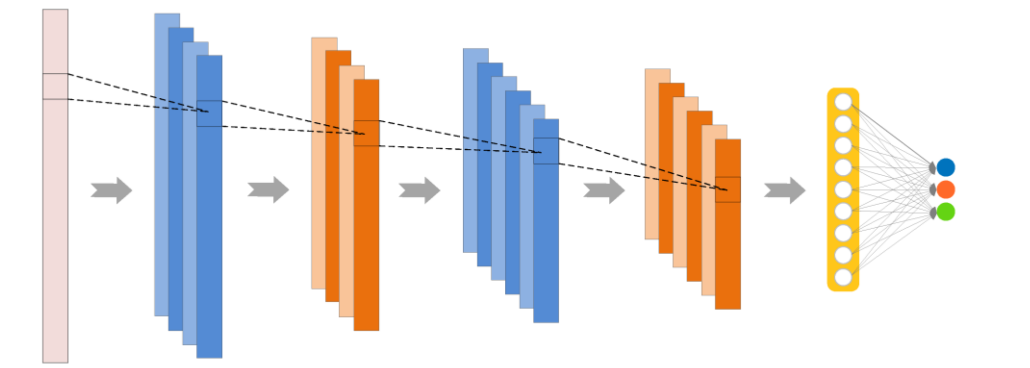

Figure.9 1D CNN architecture

The box plot, represented in Fig. 7, shows the relationship between key features and the irrigation status. The demand for irrigation is exhibited with increased temperature levels. The pressure factor shows minimal variation between irrigated and not irrigated cases. The altitude vs. status plot demonstrates a stable distribution across the statuses, mentioning the limited influence on irrigation decisions in the dataset. Soil moisture reveals a unique trend that it heavily relies on the irrigation status. Lower soil moisture necessitates irrigated cases, indicating the need for water when the soil is dry.

The data points by the status of irrigation are distinguished by understanding the feature relationships using a pair plot, as shown in Fig. 8. Temperature and pressure demonstrate a weak positive linear relationship, whereas soil moisture and altitude rely on particular values. The variability represented by the factor 'note' indicates the presence of contextual information that is crucial for irrigation decisions.

3.3 Model development

3.3.1. 1D convolutional neural network

One-dimensional sequential data is processed using a 1D CNN (Conv1D), which is a type of neural network [25]. Unlike traditional 2D CNNs, 1D CNNs process in one spatial dimension, making them suitable for applications involving time-series data, audio signal, text sequences and temporal or sequential structural data [26]. The necessity of manual feature engineering is reduced by automatically extracting features from the sequential data using Conv1D while preserving spatial dependencies in the input data. A three-dimensional tensor is used to structure the input of a 1D CNN layer. The number of independent samples processed simultaneously during training is represented by the batch size. Efficient use of computational resources is ensured by the multiple sequences in each batch. The time steps in each sequence is indicated by the sequence length. Finally, the number of features or channels related to each data point in the sequence is denoted by the input dimension. This structure helps Conv1D to handle multivariate sequential data. The Conv1D employs a convolutional operation that applies a kernel (filter) to the input sequence to extract local patterns. The basic 1D CNN architecture is shown in Fig. 9 with convolution layer, pooling layers and fully connected layer.

Eq. (2) defines the convolutional operation, which describes the computation of a weighted sum of input values after bias addition and a non-linear transformation.

$$y_t^j = \sigma\left(\sum_{i=0}^{k-1} w_{i,j} \cdot x_{t+i} + b_j\right) \tag{2}$$

where y_t^j represents the output of the j^{th} filter at position t, $w_{i,j}$ is the weights of the i^{th} element in the j^{th} filter, x_{t+i} is the input sequence values covered by the filter at position t, b_j is the bias term and k and σ are the kernel size and activation function, respectively.

Another key component of Conv1D, which functions as a sliding window across the input sequence, is the filter or kernel. The number of sequential inputs processed at a time is represented by the filter defined by its kernel size. The weighted sum of the input values is computed to capture localized patterns within the sequence. A feature map is generated for each filter that indicates the presence of specific patterns learned by the network.

Stride determines the movement of the filter across the input sequence. To facilitate precise feature extraction, the filter shifts over time, overlapping with the previous position. A higher stride value reduces the output size and computation

Received: February 25, 2025. Revised: April 10, 2025.

cost. The balance between computational efficiency and granularity of feature extraction is done with the stride parameter. The size of the output sequence is controlled by the padding technique. By adding zeros around the edges, padding is applied to the input sequence in Conv1D. It ensures that the output size is consistent with the input size. The main padding strategies are: padding=valid, where no padding is employed and the output sequence is smaller than the input sequence. This padding provides a computationally efficient process but may lose edge information. The next strategy is padding=same, where the same sequence length in the input is maintained in the output. This helps reduce dimensionality reduction across layers in complex architectures. Fig. 10 demonstrates 1D convolution.

The presence of features learned by the filters in the input data is summarized by a feature map represented by the output of Conv1D. The number of different patterns learned by the network is represented by the number of filters and each filter generates a unique feature map. Complex patterns are learned by introducing non-linearity into the network using an activation function. In CNNs, a down-sampling operation is employed by using 1D MaxPooling, especially for sequential or time-series data. Essential information is preserved while reducing the dimensionality of the feature maps shaped by convolutional layers. 1D MaxPooling emphasizes the most prominent features identified by the filters in specific areas of the input sequence within a sliding window by selecting the maximum value.

For an input feature map x of size L, the MaxPooling output y is computed by selecting the maximum value within the window, as given in Eq. (3).

$$y_t = max\,(x_t, x_{t+1}, \dots \dots x_{t+k-1}) \tag{3}$$

where t is the starting index of the pooling window and k represents the kernel size. The degree of window shifts at each step is determined by moving a sliding window across the sequence with a stride s. The overlapping of pooling operation happens if $s > 1$. Eq. (4) calculates the output length L_{out}.

$$L_{out} = \begin{cases} \frac{L-k}{s} + 1 & if\ padding = 'valid' \\ \frac{L+2p-k}{s} + 1 & if\ padding = 'same' \end{cases} \tag{4}$$

where p is the padding size.

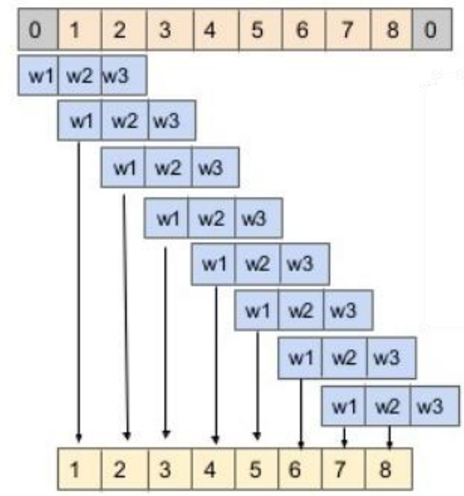

Figure.10 1D Convolution

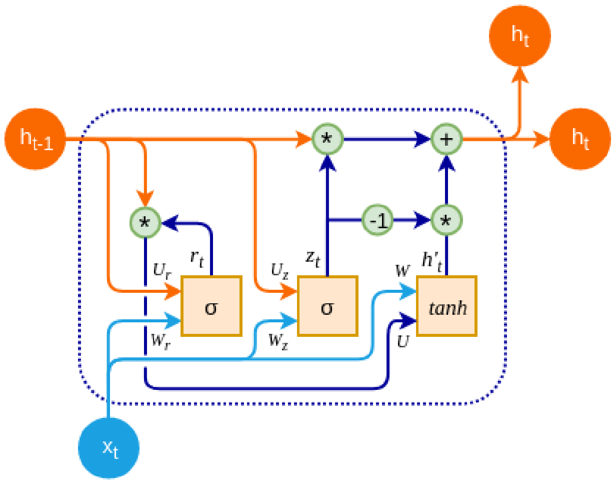

Figure.11. GRU architecture

3.3.2. Bi-GRU

GRU is a recurrent neural network (RNN) that process sequential data such as text, time series data and speech [27]. The GRU architecture comprises gates, as shown in Fig. 11. The sequential data, x^t is processed by the GRU cell that captures relevant information maintained by a hidden state from the past and updates this state based on the current input and previous hidden state, where t is the time step ranges from 1 to T.

As given in Eq. (5) the update gate regulates information of the past state to retain.

$$z^t = \sigma(W_z.x^t + U_z \cdot h^{t-1} + b_z) \tag{5}$$

where b is the bias term, h^{t-1} is the preceding time step's hidden state, U and W are the weight matrices and σ represents the sigmoid activation function. The reset gate regulates how much of the preceding hidden state should be forgotten, as shown in Eq. (6).

$$r^t = \sigma(W_r . x^t + U_r \cdot h^{t-1} + b_r) \quad (6)$$

The candidate hidden state represents the new candidate values for the hidden state, as expressed in Eq. (7).

$$\widetilde{h^t} = \tanh\left(W \cdot x^t + r^t \circ (U \cdot h^{t-1}) + b\right. \quad (7)$$

where $tanh$ is the hyperbolic tangent activation function.

The integration of the preceding hidden state and the candidate hidden state weighted by the update gate is indicated by the final hidden state. The mathematical expression of the final hidden state is given in Eq. (8).

$$h^t = (1 - z^t) \circ h^{t-1} + z^t \circ \widetilde{h^t} \quad (8)$$

Bi-GRU is a form of the GRU that analyses sequential data in both forward and backward directions to capture contextual information from both the past and the future [28]. This architecture is especially suitable for tasks like NLP, time-series forecasting and speech recognition. Fig. 12 shows the structure of Bi-GRU layers.

The Bi-GRU architecture comprises several components. In first input layer, input can be a sequence of words in a sentence, frames in a video or time steps in a timeseries. The next is the bidirectional layer, consisting of two GRU layers, which process the sequence in both directions. The forward layer processes the sequence from the first-time step x^t to the last time step x^T, whereas the backward layer processes in reverse direction, from the last time step x^T to the first-time step x^{t-1}, as expressed in Eq. (9) and Eq. (10).

$$h^t_{forward} = GRU_{forward}(x^t, h^{t-1}_{forward}) \quad (9)$$

$$h^t_{backward} = GRU_{backward}(x^t, h^{t+1}_{backward}) \quad (10)$$

The forward and backward layers produce two sets of hidden state, $h^t_{forward}$ for the forward direction and $h^t_{backward}$ for the backward direction. Each GRU cell that follows processes one-time step of the sequence in both backward and forward directions. The input x^t and the hidden state from the preceding time step are used to compute the current hidden state. The flow of information is controlled to retain the long-term dependencies and reduce the vanishing gradient problem by employing the reset gate and the update gate. To introduce non-linearity hidden states are passed through an activation function. The complex patterns in the sequential data are modelled using the activation function. Finally, after processing the input sequence, both forward and backward GRUs are concatenated as shown in Eq. (11).

$$h^t_{BiGRU} = Concat(h^t_{forward}, h^t_{backward}) \quad (11)$$

where h^t_{BiGRU} is the combined hidden state at time step t. This final hidden state is passed to further layers, like a dense layer for prediction tasks.

3.3.3. Proposed hybrid deep learning model

The proposed model integrates 1D CNN and Bi-GRU models to effectively predict irrigation. The overall predictive accuracy of the proposed model is improved by capturing temporal dependencies and extracting spatial features. The architecture begins with a 1D CNN layer that processes the input sequential data. An input layer is followed by a Conv1D layer with 32 filters and a kernel size of 3. From the input sequence local patterns are extracted by applying the ReLU activation function. A dropout layer with a 30% dropout rate is applied to reduce overfitting. Specific temporal features are extracted using the second Conv1D layer with a kernel size of 1 and 64 filters.The granularity of the features is maintained using a MaxPooling1D layer with a pool size of 1. 64 units of the Bi-GRU layer are employed for capturing temporal dependencies. This layer processes in both forward and backward directions to learn long-term dependencies and relationships in the input sequences. The final classification is attained using a fully connected dense layer. The extracted features by the 1D CNN and Bi-GRU layers are integrated by a hidden dense layer with the ReLU and 128 neurons. A dropout layer is followed by an output dense layer with a sigmoid activation function and a single neuron, which predicts the status of irrigation as ‘Irrigated’ or ‘Not-Irrigated’. The model summary and model architecture are given in Table 1 and Fig. 13, respectively. The algorithm that details the proposed irrigation prediction model is given below.

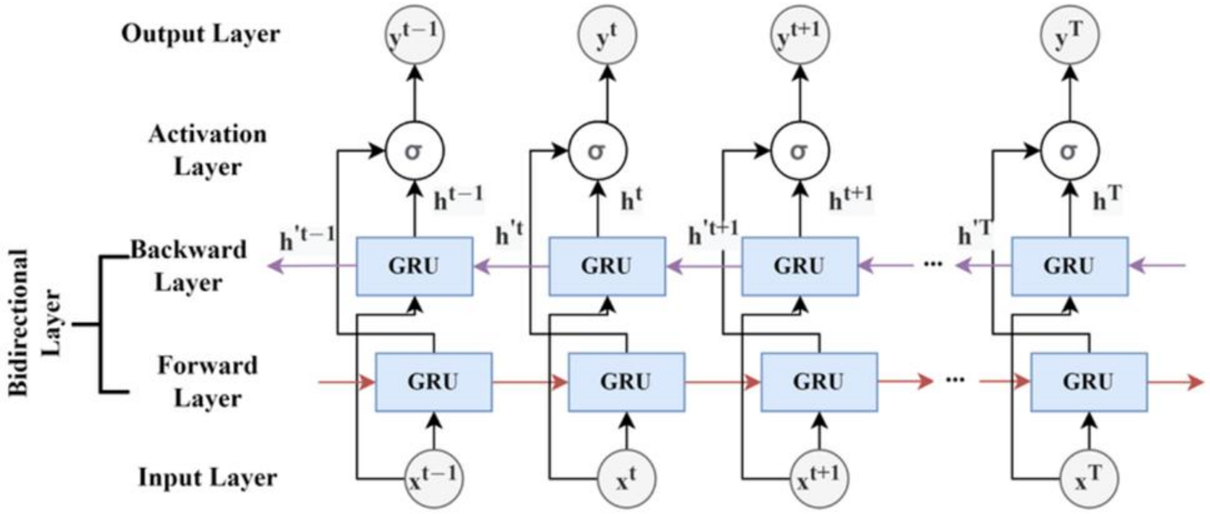

Figure.12 Architecture of Bi-GRU

Algorithm: *Irrigation prediction using 1D CNN and Bi-GRU*

Input: *Irrigation scheduling for smart agriculture dataset*
Output*: A binary prediction of irrigation status ('1'='Irrigated', '0'='Not-Irrigated')*

Begin
Step 1: Load data

- *The dataset comprising environmental, agricultural and geographical features that affect the irrigation process.*

Step 2: Data preprocessing and exploratory data analysis

- *Handle missing values*
- *Remove outliers*
- *Handle categorical variables*
- *Numerical feature scaling*
- *Normalization using,* $x' = \frac{x-\mu}{\sigma}$
- *Correlation analysis*
- *Feature Relationships with Target Variable*

Step 3: Model development

- ***Feature extraction using 1D CNN***
- ❖ *Add the Input layer as the first layer*
- ❖ *Apply the convolutional operation using,*

$$y_t^j = \sigma\left(\sum_{i=0}^{k-1} w_{i,j} \cdot x_{t+i} + b_j\right)$$

- ❖ *Apply MaxPooling using,*

$$y_t = max\ (x_t, x_{t+1}, \dots \dots x_{t+k-1})$$

- ❖ *Model= (Conv1D (32, kernel_size=3)*
- ❖ *MaxPooling1D =pool_size=3, Conv1D_1*
- ❖ *Conv1D layer with adjusted kernel_size, reduced pool size*
  Model = (Conv1D (64, kernel_size=1)
  MaxPooling1D =pool_size=1, Conv1D_2
- ***Temporal feature extraction using Bi-GRU***
- ❖ *Forward hidden state calculation using,*

$$h_{forward}^t = GRU_{forward}(x^t, h_{forward}^{t-1})$$

- ❖ *Backward hidden state calculation using,*

$$h_{backward}^t = GRU_{backward}(x^t, h_{backward}^{t+1}$$

- ❖ *Concatenation of forward and backward hidden state using,*

$$h_{BiGRU}^t = Concat(h_{forward}^t, h_{backward}^t)$$

- ***Final prediction using dense layer***
  - *Model= (Dense (128, activation='relu'))*
  - *Model = (Dense (1, activation='sigmoid'))*

Step 4: Model compilation

- *Building the 1D CNN + Bi-GRU model for irrigation prediction*
- *Model = CNN_Bi-GRU compile (optimizer=Adam (), loss='binary_crossentropy',*

Step 5: Model evaluation

- *metrics=['accuracy'])*
- *Adjusting hyperparameters*

Save the model

End

Table 1. Model summary

Layer (type)	Output Shape	Parameters
conv1d (Conv1D)	(None, 3, 32)	128
dropout (Dropout)	(None, 3, 32)	0
conv1d_1 (Conv1D)	(None, 3, 64)	2,112
max_pooling1d (MaxPooling1D)	(None, 3, 64)	0
dropout_1 (Dropout)	(None, 3, 64)	0
bidirectional (Bidirectional)	(None, 128)	49,920
dropout_2 (Dropout)	(None, 128)	0
dense (Dense)	(None, 128)	16512
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
Total params: 68,801 Trainable params: 68,801 Non-trainable params: 0		

Table 2. Hyperparameter specification

Hyperparameters	Values
Optimizer	Adam
Activation functions	ReLU, sigmoid
Number of epochs	200
Loss	Binary cross entropy
Dropout rate	0.3
Batch size	32

3.4 Hardware and software setup

A comprehensive arrangement comprising an NVIDIA GeForce GTX 1080Ti GPU, 32GB of RAM and an Intel Core i7 processor is used in this study for effective training. This novel hybrid model has been implemented on the Google Collaboratory platform with the Python language. Python integrated with the TensorFlow framework and Keras's user-friendly interface assures the execution of complex structures. A number of hyperparameters that are set before training are utilized in this research to optimize the performance of the hybrid model. These configuration parameters are essential that indicate the operation and functions of a DL model throughout the training. Table 2 shows the hyperparameters employed in this study. These hyperparameters collectively enhance the ability of the model for effective generalization and efficient data learning.

The model was optimized using the Adam optimizer, which is known for its efficient gradient-based optimization and adaptive learning rate. The network utilized a combination of ReLU and sigmoid activation functions, where ReLU was applied in hidden layers to introduce non-linearity and prevent vanishing gradient issues, while the sigmoid function was used in the output layer for binary classification of irrigation needs. The model was trained for 200 epochs, ensuring sufficient learning while minimizing the risk of overfitting. Binary cross-entropy was selected as the loss function due to the binary nature of the irrigation decision (water or no water). To prevent overfitting and improve generalization, a dropout rate of 0.3 was implemented, randomly deactivating 30% of neurons during training. The model processed data in mini-batches of 32, balancing computational efficiency and model stability during training.

4. Results and discussion

Accuracy and loss plots are visual tools that track the efficiency of the model over time during training and validation. The accuracy plot shows the proportion of correct predictions, helping identify if the model's predictive power improves with each

Received: February 25, 2025. Revised: April 10, 2025.

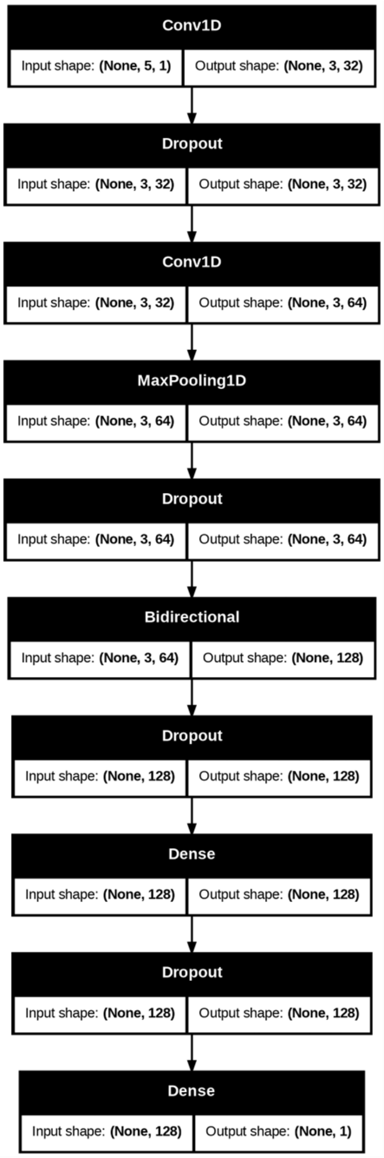

Figure.13 Proposed irrigation prediction model architecture

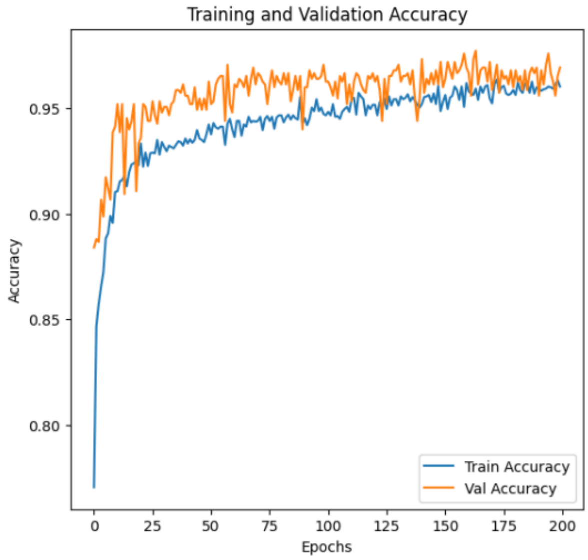

Figure.14 Accuracy plot of the irrigation prediction model

Figure.15 Loss plot of the irrigation prediction model

epoch. Meanwhile, the loss plot displays the error rate, indicating the model's effectiveness in fitting the data; lower loss over epochs improved learning. In this study, these plots are important for evaluating the effectiveness of the model in determining irrigation prediction. Fig. 14 and Fig.15 present the accuracy and loss plot of the framework.

The upward trends in training and validation accuracy signify effective learning and generalization. The validation accuracy increases steadily, starting at approximately 0.88 and eventually nearing 0.98, while the training accuracy follows a similar trend that continues to increase gradually and stabilizes close to 0.98 by the final epoch of 200. This close alignment between the validation and training accuracy denotes the efficiency of the model in learning and generalization without significant overfitting.

A consistent decrease as the training progresses is displayed by the validation and training loss over 200 epochs. The validation loss initializes at 0.36 and decreases eventually to a lower value of 0.08 in the final epochs. Similarly, the training loss steadily decreases from 0.6 to reach 0.95 over 200 epochs. These values indicate the model's ability in minimizing errors during training. The minimal gap between the curves implies that the model efficiently balances fitting the training data and generalization to unseen data.

Received: February 25, 2025. Revised: April 10, 2025.

Table 3. Classification report of the proposed model

	Precision	**Recall**	**F1-score**	**Support**
Irrigated	0.98	0.97	0.97	577
Not-Irrigated	0.95	0.97	0.96	361
Accuracy		0.9729		938
Macro avg	0.97	0.97	0.97	938
Weighted avg	0.97	0.97	0.97	938

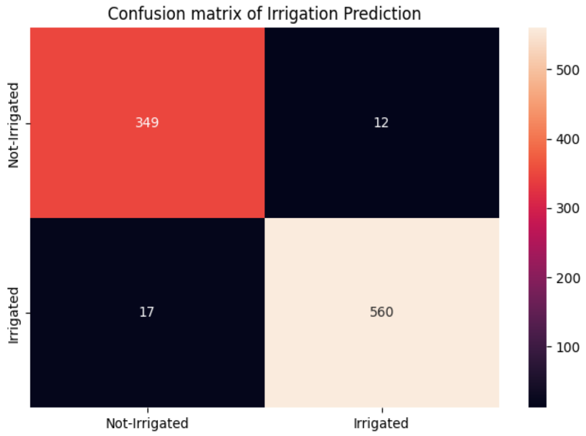

Figure.16 Confusion matrix of the irrigation prediction system

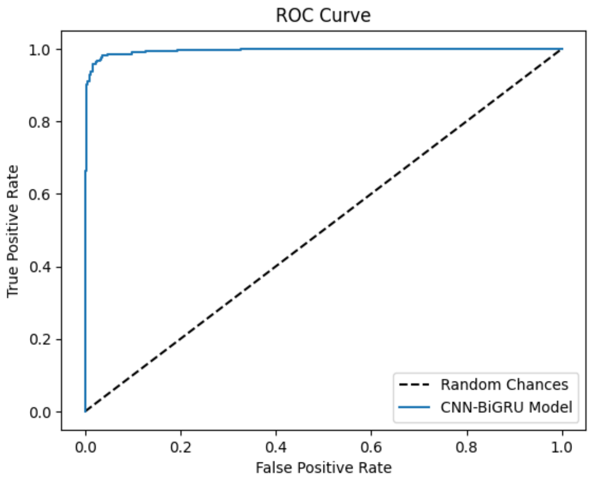

Figure.17. ROC curve

Various factors are defined to quantify essential performance parameters, as represented in following Equations. These metrics, based in the principles of False Positive (FP), True Negative (TN), False Negative (FN), and True Positive (TP), are crucial for evaluating the efficacy of the model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$F1-score = 2 \times \frac{precision \times Recall}{Precision+Recall} \quad (15)$$

The classification report, shown in Table 3, suggests that the proposed hybrid model accurately predicted classes, correctly identifying the majority of samples from both classes and maintains a balanced performance across the classes.

The model achieved an overall accuracy of 0.9729 in classifying between ‘Irrigated’ and ‘Not-Irrigated’ classes. For the ‘Irrigated’ class, the model attained a precision value of 0.98, indicating the correctness in predicting the classes. A recall value of 0.97 shows that the model predicts 97% of actual irrigated samples. The balance between the recall and precision is highlighted by an F1 score of 0.97. For the ‘Not-Irrigated’ class, the recall and precision values are 0.97 and 0.95, respectively, where the precision values are slightly lesser than the ‘Irrigated’ class. An F1-score of 0.96 shows the effectiveness of the model across the samples. The macro and weighted average is 0.97 for both the classes, indicating similar performance.

Fig. 16 demonstrates the confusion matrix of the suggested irrigation prediction model. This plot provides a visualization of the performance for the classification task. The confusion matrix helps in comparing the model’s prediction to true labels. For instance, in the proposed irrigation prediction model, the model correctly classified 349 samples as ‘Not-Irrigated’ and 560 samples as ‘Irrigated’ class. There are very few instances of misclassification with the majority of correct predictions, emphasizing the model’s robustness.

Fig. 17 demonstrates the Receiver operating characteristics (ROC) Curve that evaluates the ability of the proposed model to distinguish between the two classes. The 1D CNN-BiGRU model indicates robust discriminative performance. The diagonal line represents random chances. The high TP rate with a low FP rate reflects the model’s excellence in classification. These outcomes reveal the efficiency of the suggested model in achieving high accuracy, minimal error and excellent classification performance.

Table 4. Performance comparison with existing models

Author & Ref	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Ndunagu et al. [13]	Machine Learning	89	93	79	-
Singh et al. 14]	CNN	97.1	85.5	86.8	85.8
Rathore & Rajavat [23]	SVM	92	92	92	92
Proposed model		**97.29**	**97.87**	**97.06**	**97.46**

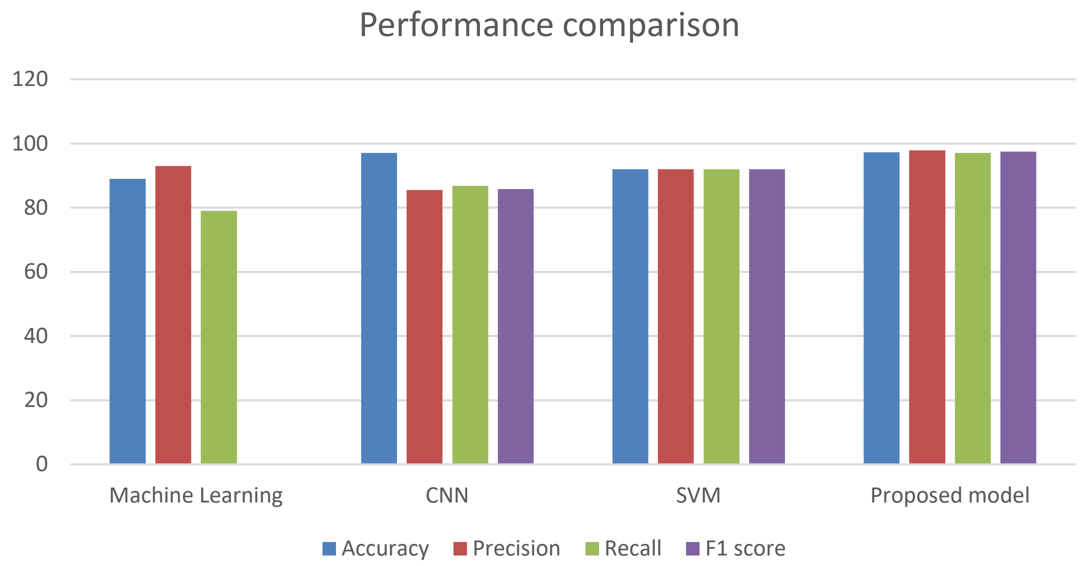

Figure.18 Performance comparison of the proposed model with existing methods

The performance of the proposed irrigation scheduling model using a 1-D CNN Bi-GRU has been evaluated and compared with existing models, as shown in Table 4 and Fig. 18. The proposed model achieved an overall accuracy of 97.29%, which is higher than the CNN (97.1%), and SVM (92%). Moreover, the proposed model attained a precision of 97.87% and a recall of 97.06%, resulting in a high F1 score of 97.46%. This consistent performance across different evaluation metrics indicates that the proposed model is not only highly accurate but also balanced in terms of its ability to identify both irrigated and non-irrigated conditions correctly.

Traditional machine learning models, while offering a respectable accuracy of 89%, showed limitations in recall (79%), suggesting a relatively lower ability to correctly identify irrigation needs based on real-time environmental inputs. This underperformance is likely due to their reliance on handcrafted features and limited capacity to capture complex temporal dependencies in the sensor data. Similarly, the SVM model, with an accuracy of 92%, shows limitations in handling large and complex datasets due to its computational complexity and difficulty in managing non-linear relationships. The CNN model, although achieving an accuracy of 97.1%, suffers from relatively low precision (85.5%) and recall (86.8%), indicating that it fails to consistently distinguish between irrigated and non-irrigated samples, leading to misclassification and lower F1 score (85.8%).

The superior performance of the proposed model can be attributed to the combined architecture of 1-D CNN and Bi-GRU. The CNN layer effectively extracts spatial features from the input data, capturing important patterns related to irrigation status, while the Bi-GRU layer enhances the model's capability to process sequential dependencies, improving temporal awareness. This hybrid architecture allows the model to learn both spatial and temporal correlations in the data, resulting in more accurate and reliable predictions. Furthermore, the Bi-GRU's bidirectional nature allows the model to retain past and future contextual information, enhancing its predictive capability for complex irrigation scenarios.

5. Conclusion

This study presents a hybrid deep learning model combining 1D Convolutional Neural Networks (1D-CNN) and Bidirectional Gated Recurrent Units (Bi-GRU) for intelligent irrigation prediction. The model effectively addresses critical agricultural challenges—such as over-irrigation and under-irrigation—by leveraging the spatial learning capabilities of 1D-CNN and the temporal modeling strength of Bi-GRU. Using the Irrigation Scheduling for Smart Agriculture dataset, which includes environmental variables such as altitude, atmospheric pressure, temperature, and soil moisture, the model achieved a notable accuracy of 97.29%. This high level of precision highlights the model's efficacy in learning spatial-temporal dependencies for irrigation forecasting.

The practical implications of this model are significant. Its deployment within smart irrigation systems can enhance water management practices by enabling timely, data-driven decisions. The binary classification of irrigation need can be integrated into automated irrigation controllers, making real-time adjustments based on environmental input data collected via IoT sensors or weather APIs. This not only optimizes water usage—critical in regions facing water scarcity—but also improves crop health and yield by ensuring consistent and accurate irrigation.

Integration into existing irrigation management systems can be achieved through APIs or lightweight edge computing modules, allowing farmers to receive irrigation alerts or enable fully automated watering schedules. The model's high accuracy ensures trust in decision outputs, while its relatively lightweight architecture allows deployment on portable agricultural devices. However, practical deployment does come with challenges such as sensor calibration, network connectivity in remote areas, and the need for regional model tuning based on local soil and crop conditions.

Looking ahead, future research could focus on enhancing real-time adaptability by integrating streaming data and exploring reinforcement learning approaches for personalized irrigation strategies. Additionally, large-scale field trials across diverse geographic locations could help validate the model's generalizability and refine its effectiveness in various agricultural ecosystems.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

Acknowledgments

We would like to express our sincere gratitude to all those who contributed to the completion of this research paper. We extend our heartfelt thanks to family, colleagues and fellow researchers for their encouragement and understanding during the demanding phases of this work.

References

[1] D. Pimentel, J. Houser, E. Preiss, O. White, H. Fang, L. Mesnick, and S. Alpert, "Water resources: agriculture, the environment and society", *BioScience*, Vol. 47, No. 2, pp. 97-106, 1997.

[2] V. Dhawan, "Water and agriculture in India", *in Background paper for the South Asia expert panel during the Global Forum for Food and Agriculture*, Vol. 28, pp. 80-85, 2017.

[3] C. Brouwer and M. Heibloem, "Irrigation water management: irrigation water needs", *Training manual*, No. 3, pp. 1-5, 1986.

[4] H. Krupakar and A. Jayakumar, "A review of intelligent practices for irrigation prediction", *arXiv preprint arXiv:1612.02893*, 2016.

[5] L. S. Pereira, T. Oweis, and A. Zairi, "Irrigation management under water scarcity", *Agricultural Water Management*, Vol. 57, No. 3, pp. 175-206, 2002.

[6] C. E. Hachimi, S. Belaqziz, S. Khabba, B. Sebbar, D. Dhiba, and A. Chehbouni, "Smart weather data management based on artificial intelligence and big data analytics for precision agriculture", *Agriculture*, Vol. 13, No. 1, p. 95, 2022.

[7] O. Adeyemi, I. Grove, S. Peets, and T. Norton, "Advanced monitoring and management systems for improving sustainability in precision irrigation", *Sustainability*, Vol. 9, No. 3, p. 353, 2017.

[8] R. O. Mendelsohn and S. N. Seo, "Changing farm types and irrigation as an adaptation to climate change in Latin American agriculture", *World Bank Policy Research Working Paper*, No. 4161, 2007.

[9] C. Sheline, F. Grant, S. Gelmini, and S. Pratt, "Designing a predictive optimal water and energy irrigation (POWEIr) controller for solar-powered drip irrigation systems in resource-constrained contexts", *Applied Energy*, Vol. 377, p. 124107, 2025.

[10] M. A. Youssef, R. T. Peters, M. El-Shirbeny, A. M. Abd-ElGawad, Y. M. Rashad, M. Hafez, and Y. Arafa, "Enhancing irrigation water management based on ETo prediction using machine learning to mitigate climate change", *Cogent Food & Agriculture*, Vol. 10, No. 1, p. 2348697, 2024.

[11] H. Yan, F. Xie, D. Long, Y. Long, P. Yu, and H. Chen, "An accurate irrigation volume prediction method based on an optimized LSTM model", *PeerJ Computer Science*, Vol. 10, p. e2112, 2024.

[12] G. K. Kumar, M. L. Bangare, P. M. Bangare, C. R. Kumar, R. Raj, J. L. Arias-Gonzáles, and M. S. Mia, "Internet of things sensors and support vector machine integrated intelligent irrigation system for agriculture industry", *Discover Sustainability*, Vol. 5, No. 1, p. 6, 2024.

[13] J. N. Ndunagu, K. E. Ukhurebor, M. Akaaza, and R. B. Onyancha, "Development of a wireless sensor network and IoT‐based smart irrigation system," *Appl. Environ. Soil Sci.*, Vol. 2022, No. 1, pp. 7678570, 2022.

[14] N. Singh, A. Kumar, L. K. Dhruw, and B. U. Choudhury, "Optimization of irrigation timing for sprinkler irrigation system using convolutional neural network-based mobile application for sustainable agriculture", *Smart Agricultural Technology*, Vol. 5, p. 100305, 2023.

[15] P. Suresh, R. H. Aswathy, S. Arumugam, A. A. Albraikan, F. N. Al-Wesabi, A. M. Hilal, and M. Alamgeer, "IoT with evolutionary algorithm based deep learning for smart irrigation system", *Computers, Materials & Continua*, Vol. 71, No. 1, pp. 1713-1728, 2022.

[16] K. Alibabaei, P. D. Gaspar, and T. M. Lima, "Crop yield estimation using deep learning based on climate big data and irrigation scheduling", *Energies*, Vol. 14, No. 11, p. 3004, 2021.

[17] K. I. M. Abuzanouneh, F. N. Al-Wesabi, A. A. Albraikan, M. Al Duhayyim, M. Al-Shabi, A. M. Hilal, and K. Muthulakshmi, "Design of machine learning based smart irrigation system for precision agriculture", *Computers, Materials & Continua*, Vol. 72, No. 1, pp. 109-124, 2022.

[18] S. Yonbawi, S. Alahmari, B. R. S. S. Raju, C. H. G. Rao, M. K. Ishak, H. K. Alkahtani, and S. M. Mostafa, "Modeling of sensor enabled irrigation management for intelligent agriculture using hybrid deep belief network", *Computer Systems Science & Engineering*, Vol. 46, No. 2, pp. 1-13, 2023.

[19] X. Xu, R. L. Patibandla, A. Arora, M. Al-Razgan, E. M. Awwad, and V. O. Nyangaresi, "An adaptive hybrid (1D-2D) convolution-based ShuffleNetV2 mechanism for irrigation levels prediction in agricultural fields with smart IoTs", *IEEE Access*, Vol. 12, pp. 1-13, 2024.

[20] R. Benameur, A. Dahane, B. Kechar, and A. E. H. Benyamina, "An innovative smart and sustainable low-cost irrigation system for anomaly detection using deep learning", *Sensors*, Vol. 24, No. 4, p. 1162, 2024.

[21] A. Kaur, D. P. Bhatt, and L. Raja, "Developing a hybrid irrigation system for smart agriculture using IoT sensors and machine learning in Sri Ganganagar, Rajasthan", *Journal of Sensors*, Vol. 2024, p. 6676907, 2024.

[22] S. Dhal, J. Alvarado, U. Braga-Neto, and B. Wherley, "Machine learning-based smart irrigation controller for runoff minimization in turfgrass irrigation", *Smart Agricultural Technology*, Vol. 9, p. 100569, 2024.

[23] N. Rathore and A. Rajavat, "Smart farming based on IoT-edge computing: Applying machine learning models for disease and irrigation water requirement prediction in potato crop using containerized microservices", *Precision Agriculture for Sustainability*, pp. 399-424, 2024.

[24] [Online]. Available: https://www.kaggle.com/code/triptmann/irrigation-scheduling-for-smart-agriculture

[25] H. Pan, X. He, S. Tang, and F. Meng, "An improved bearing fault diagnosis method using one-dimensional CNN and LSTM", *Journal of Mechanical Engineering/Strojniški Vestnik*, Vol. 64, p. 1, 2018.

[26] E. U. H. Qazi, A. Almorjan, and T. Zia, "A one-dimensional convolutional neural network (1D-CNN) based deep learning system for network intrusion detection", *Applied Sciences*, Vol. 12, No. 16, p. 7986, 2022.

[27] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks", In: *Proc. of 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597-1600, 2017.

[28] S. Wang, C. Shao, J. Zhang, Y. Zheng, and M. Meng, "Traffic flow prediction using bi-directional gated recurrent unit method", *Urban Informatics*, Vol. 1, No. 1, p. 16, 2022.