SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I3P117

Volume 12 Issue 3, 170-184, March 2025
© 2025 Seventh Sense Research Group®

Original Article

Advanced Irrigation Prediction Using a 1-D CNN and Multi-Head Attention Hybrid Framework

Anju Markose[1], K. Baalaji[2]

[1]*Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India.*
[2]*Department of CSE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India.*

[1]*Corresponding Author: anjumarkose@gmail.com*

Received: 30 November 2024 Revised: 11 March 2025 Accepted: 25 March 2025 Published: 31 March 2025

Abstract - *The increasing global population and the corresponding demand for food production necessitate the efficient use of agricultural resources, particularly water. The proposed study investigates the effectiveness of a deep learning model for classifying irrigated and non-irrigated land. The relevance of this study lies in its potential contributions to sustainable agriculture and resource management, particularly in optimizing irrigation practices and improving crop yields. Utilizing a 1-D CNN-Multi-Head Attention model, the proposed study employed a robust methodology that includes data preprocessing, feature extraction and rigorous training processes to enhance classification accuracy. The model was trained and validated on a well-curated dataset containing labeled data representing both irrigated and non-irrigated regions. Performance evaluation proved the model's efficacy with an accuracy of 97%, precision of 96.9% for irrigated areas and 99% for non-irrigated areas, highlighting its ability to accurately distinguish between the two classes. The results indicate a strong potential for this model in real-world applications, providing actionable insights for agricultural stakeholders. This method contributes to the ongoing efforts to leverage technology for efficient agricultural practices, ultimately aiming for more sustainable land use management.*

Keywords - *Irrigation prediction, Multi-head attention, Agricultural resources, Convolutional neural network, Deep learning, Kernel density estimate.*

1. Introduction

Irrigation is one of the most vital processes in the agricultural domain. Water needs differ depending on the crop and plant; therefore, irrigation is more than just dousing the plants with water [1]. Crop yields suffer from either over or under-irrigation when crops are not given the necessary amount of water.

In addition, incorrect irrigation techniques lead to a number of issues, which are listed below: the spread of chemicals, diseases, and weeds, unintended vegetative growth, and water overflows that cause soil erosion [2].

Figure 1 illustrates the various factors influencing crop-water demand, including climatic conditions, soil properties, and crop type, which collectively determine the amount of water required for optimal crop growth. Additionally, the world's water shortage is becoming a big problem, which has raised the possibility of a serious food crisis [3]. As a result, there must be a productive way to use the resources at hand wisely. Agriculture is one of the industries that uses more water. Controlling irrigation levels helps to conserve water for other uses-inadequate management results in 25% of water in agriculture [4].

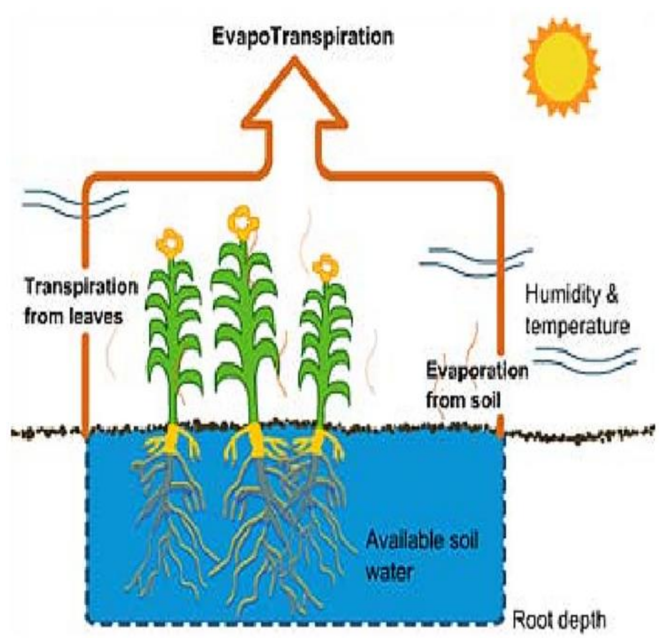

Fig. 1 Factors affecting crop-water demand

The need for food increases with the rising population, but agricultural production is further complicated by the risks

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

posed by climate change [5]. Accurately identifying irrigated and non-irrigated regions is critical for efficient water management and environmentally friendly farming methods, as irrigation significantly increases crop yields [6]. Water resource management methods are needed, particularly for the prudent utilization of water in growing crops [7]. Conventional methods faced challenges such as labour-intensive nature, time commitment, and susceptibility to human mistakes. Water availability is impacted by climate change and shifting weather patterns. So, innovative methods must be used to automate and increase the efficiency of irrigation monitoring [8]. The proposed approach combines deep learning methods for better feature extraction and contextual understanding of data by achieving better classification performance using a 1-D CNN multi-head attention model. The suggested method helps to develop precision agriculture and water resource management, thereby supporting sustainable farming practices. The study's primary contributions are listed below:

- The study addresses the limitations of traditional methods, reducing manual labor and human error through automation.
- Using a 1-D CNN-Multi-Head Attention model improves the precision of irrigation monitoring, particularly in environments impacted by climate change and varying weather patterns.
- By employing advanced DL techniques, the model enhances feature extraction and contextual understanding, leading to superior classification performance.

The remainder of the paper is organized as follows: A thorough literature review highlighting the necessity of the current research is presented in Section 2. The deep learning model architecture and technique for the suggested model are described in detail in Section 3. The results and discussion are presented in Section 4, which also highlights the potential of the proposed model. Section 5 analyses the key contributions to wrap up the research.

2. Related Works

Chen et al. [9] suggested an ensemble learning-based smart irrigation model. They used the controlling records and an IoT system to collect the data. After preprocessing the data, the authors train four models, namely linear Support Vector Regression, support vector classification, Adaboost, and Logistic regression. Additionally, XGBoost was used as the blender, and a stacked generalization was used to combine the advantages of four different methods. The single-shed and multi-shed models had mean absolute errors of 10.91 and 15.45, respectively; the mean bias error rates were -1.81 and -6.36, and the root mean square error values were 15.36 and 17.32. Through technology-assisted irrigation, Megalingam et al. [10] suggested an IoT-based monitoring and wirelessly controlled irrigation system to assist farmers in producing more productive crop harvests. A microcontroller unit serves as the master controller for this wireless rover system. The rover is controlled wirelessly with joysticks. The rover system's sensor unit comprises a temperature and moisture sensor, which detects the temperature and moisture, respectively. A smart irrigation system was introduced by Esmail et al. [11] that makes use of data analysis, ML, and the IoT to figure out the best method to schedule and apply water for irrigation. This creative method makes use of a well-selected dataset called "Crop Irrigation Scheduling," sourced from Kaggle.

The decision tree algorithm achieved the highest accuracy at 89.68%, followed by the logistic regression algorithm at 85.14% and the support vector machine at 84.15%, demonstrating their effectiveness in predictive performance. A DL-based IoT-enabled irrigation method for precision agriculture was proposed by Kashyap et al. [12]. The system to forecast soil moisture content was based on recurrent neural networks. Regulating the irrigation scheduler's operation concentrates on the essential needs of agriculture, such as the quantity of water conserved and the duration of irrigation.

An IoT system for smart watering of greenhouses was proposed by Risheh et al. [13], employing four soil sensors and artificial neural networks. Four sensors located in various soil strata are used in the approach to forecast future moisture levels. An IoT and ML-enabled smart irrigation approach was proposed by Abuzanouneh et al. [14] for precision agriculture. With the use of this technique, field parameters were sensed, and irrigation decisions were made accordingly. The Artificial Algal Algae (AAA) model was used in conjunction with the Least Squares-SVM model for classification, and the method's accuracy was 97.5. An IoT infrastructure based on ThingSpeak and NodeMCU was developed by Saini et al. [15] to enable farmers to monitor temperature and moisture parameters and regulate irrigation from a PC or smartphone at any time. An edge-based irrigation system was developed by Premkumar et al. [16] to help farmers determine the optimal timing for watering plants through a decision-making framework.

The farming system can automatically and effectively adjust to changes in climatic circumstances with an enabled edge computing framework. Regression algorithms were used to predict soil moisture, and then k-means clustering was included to estimate the changes in soil moisture prediction. Zouizza et al. [17] suggested a CNN-LSTM model-based smart irrigation system. By managing the irrigation scheduling function, it seeks to meet the essential needs of agricultural irrigation, including water supply and irrigation scheduling. Tests were conducted under various environmental conditions, including temperature and humidity, using data collected from sensors.

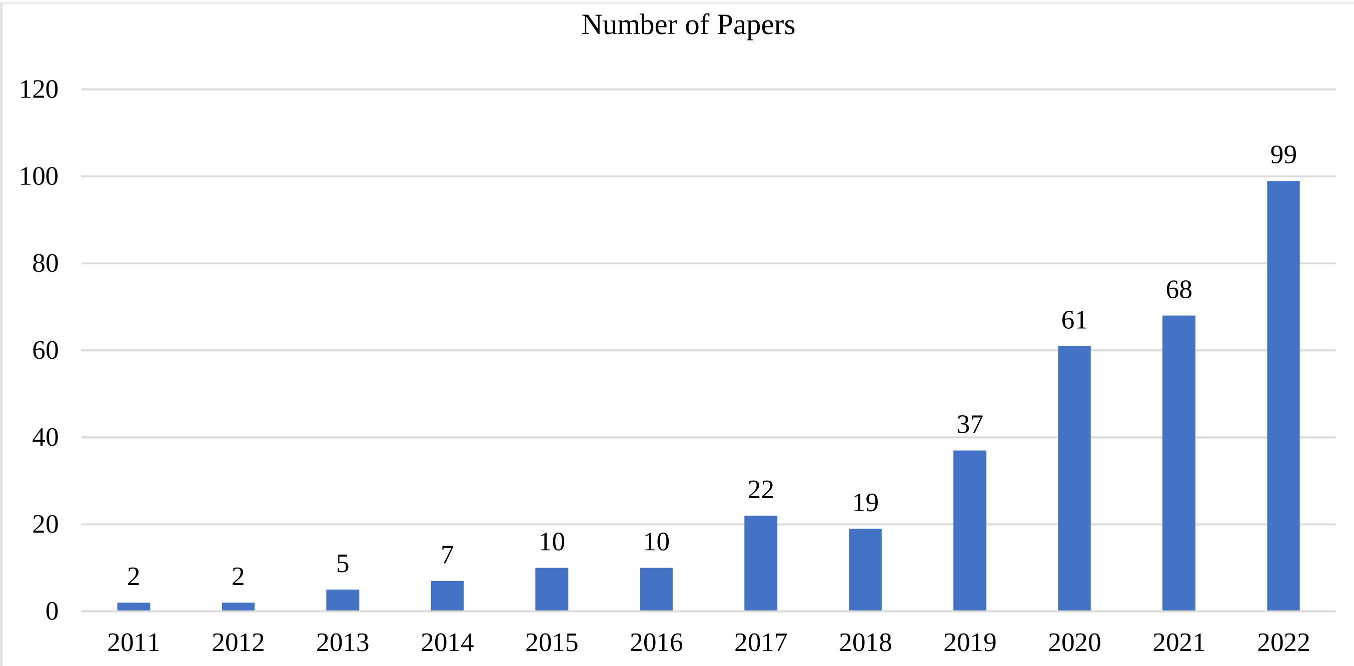

Fig. 2 Number of papers published per year on irrigation

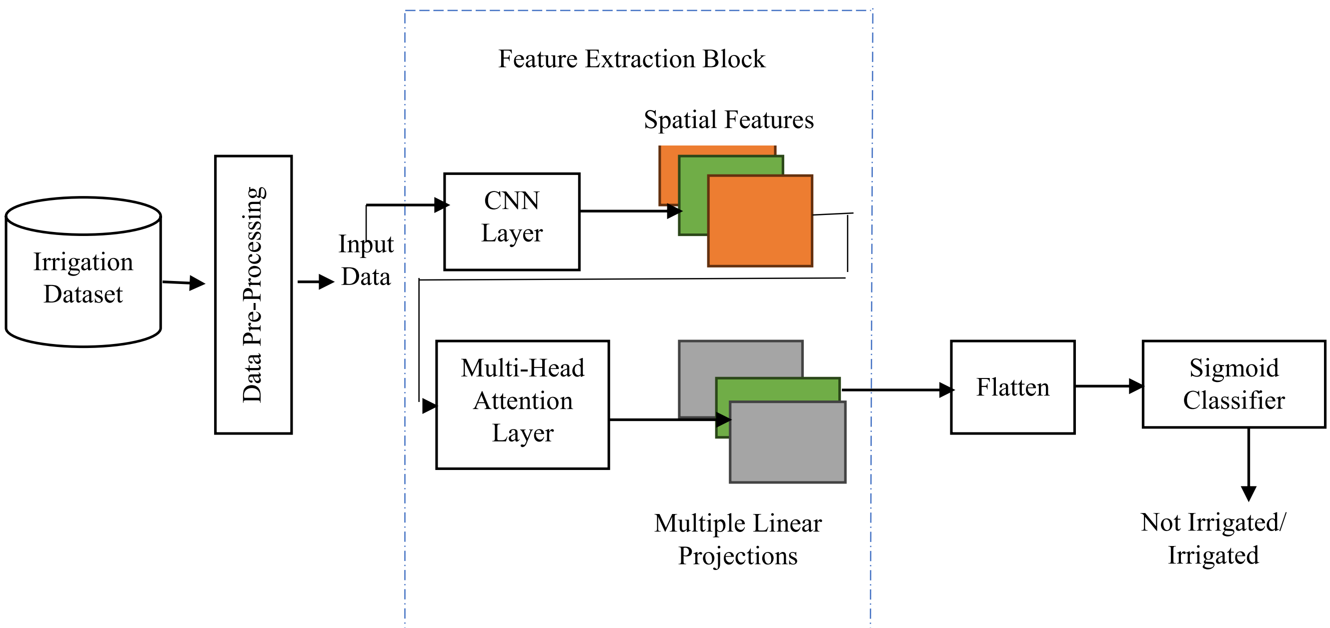

Fig. 3 Conceptual diagram of the proposed method

Al-Ghobari et al. [18] created ANN and multiple linear regression models using data gathered from published studies on anticipated WDEL. The yearly trend in the number of published articles on irrigation is depicted in Figure 2, which emphasizes the recent rise in this field of study. Conventional irrigation methods have issues with water loss and scheduling, making it exciting to meet the increasing requirement for food production while simultaneously protecting important water sources. Despite the advancements in remote sensing and image classification techniques, current studies often suffer from limitations, such as inadequate datasets, the inability to generalize across different geographical regions, and using less sophisticated models.

Many existing methodologies do not adequately leverage deep learning's capabilities, resulting in suboptimal performance in accurately distinguishing between irrigated and non-irrigated land.

3. Materials and Methods

The proposed model is a hybrid architecture that combines 1-D CNN with Multi-Head Attention mechanisms, as shown in Figure 3. In this study, the dataset used for irrigation prediction was derived from irrigation scheduling data in the form of a CSV file. The input features were normalized using the Standard Scaler method to ensure uniformity. Before feeding the data into the model, an exploratory analysis was conducted to understand the structure of the dataset better.The 1-D CNN served as the initial feature extraction layer, passing the input data through multiple convolutional layers. These layers applied filters to get local patterns in the data. The output of these convolutional layers was then flattened to prepare it for the next stage of the model. Following the CNN layers, the model incorporated a mechanism of Multi-Head Attention. By reshaping the flattened output into a sequence format, this approach enabled the attention layers to focus on various portions of the input data concurrently. Ultimately, fully connected layers processed the data from the attention layers to generate a final binary prediction indicating whether irrigation was necessary.

3.1. Dataset

In this study, the dataset was obtained from irrigation scheduling data in the form of a CSV file, available at [19]. The dataset consists of several input features and a target variable, as shown in Table 1, labeled as 'status', which indicates whether irrigation was required (on/off) or not.

Table 1. Class labels in the dataset

Feature	Description
Temperature	Represents the environmental temperature at the time of data collection. Higher temperatures typically increase the water demand, while lower temperatures might reduce it. Usually ranges between -10°C to 50°C, depending on the environmental conditions.
Pressure	Atmospheric pressure is another important environmental parameter that can influence weather conditions and irrigation needs. It might be used in conjunction with other factors like temperature and humidity to assess the overall climate condition and its impact on soil moisture and plant health. Standard atmospheric pressure at sea level is about 1013 hPa (hectopascals), and variations can range from 900 hPa to 1050 hPa, depending on altitude and weather conditions.
Altitude	Refers to the height of the location where the data was collected, above sea level. Altitude can affect temperature, pressure, and oxygen levels, all of which can influence plant growth and irrigation needs. Higher altitudes generally have cooler temperatures and lower atmospheric pressures. It can vary widely depending on the geographical location, from sea level (0 meters) to several thousand meters (e.g., mountainous regions).
Moisture in Soil	It is a critical indicator of the water content in the soil. It directly affects plant health and the need for irrigation. Lower moisture levels show dry soil, while higher levels suggest that the soil is sufficiently wet. Typically expressed as a percentage, 0% indicates completely dry soil, and 100% indicates fully saturated soil.
Notes	This column likely contains additional information or annotations related to the specific data instance. It could include observations made during data collection, such as weather conditions, irrigation status, or other contextual details that might influence the interpretation of the data. This feature can be useful for understanding anomalies or special conditions not captured by the numerical data. It varies widely depending on the context and could include notes like "High wind", "Overcast", or specific conditions observed during data collection.
Status	Not Irrigated = 0, Irrigated = 1.

	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

Fig. 4 Sample dataset

Figure 4 shows a sample dataset, providing a snapshot of the data used in the study, including features related to crop-water demand and irrigation status. The input features were gathered from the target variable, and the data was split into two sets: eighty percentage was used for training the model, while twenty percentage was held out for testing. Figure 5 presents the data visualization, offering a graphical representation of the dataset to illustrate how much of the data is under the irrigated and not irrigated category.

3.2. Data Preprocessing and Exploratory Data Analysis

Preprocessing steps are applied to ensure that the input features are in an optimal format for the model. To ensure uniformity, the input features were normalized using the Standard Scaler method.

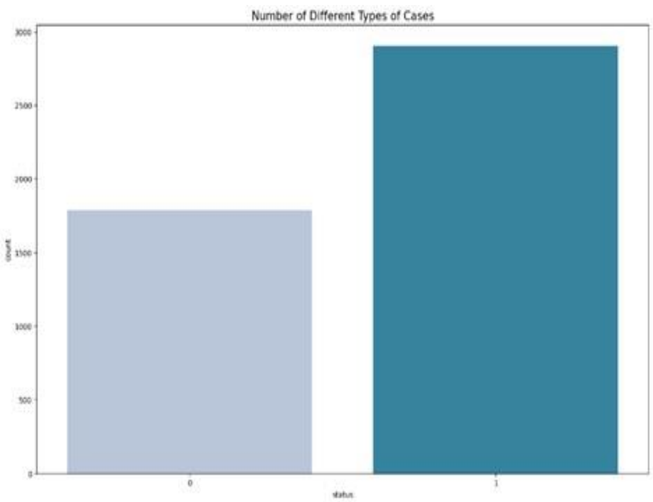

Fig. 5 Data visualization

By changing them so that their standard deviation is one and their mean is zero, this technique standardized the features essential for improving model performance.The normalized data was reshaped into the required format for CNN input, and a channel dimension was added to the data. Before feeding the data into the model, an exploratory analysis was conducted to better understand the dataset's structure.

This involved visualizing the distribution of the target variable and examining correlations between input features. Figure 6 showcases the statistical analysis of the data to provide details about the distribution and characteristics of the dataset. Understanding the properties of each feature, including how data is spread and whether it is skewed or not, depends on a study of the dataset feature distribution. A set of histograms is generated in this process, whereas the frequency distributions of values for that particular feature differ.

	temperature	pressure	altitude	soilmiosture	note	status
count	4688.000000	4688.000000	4688.000000	4688.000000	4688.000000	4688.000000
mean	29.599089	9963.153215	-14.293590	243.692406	1.878413	0.619027
std	5.842685	1383.602527	2.649662	76.176855	1.152977	0.485678
min	27.970000	-2120.400000	-17.610000	-243.000000	0.000000	0.000000
25%	28.630000	9935.255000	-16.350000	171.000000	1.000000	0.000000
50%	29.180000	9969.535000	-13.470000	233.000000	2.000000	1.000000
75%	29.990000	9975.700000	-12.950000	326.000000	3.000000	1.000000
max	178.700000	99931.100000	116.700000	480.000000	3.000000	1.000000

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

Fig. 6 Illustration of data statistical analysis

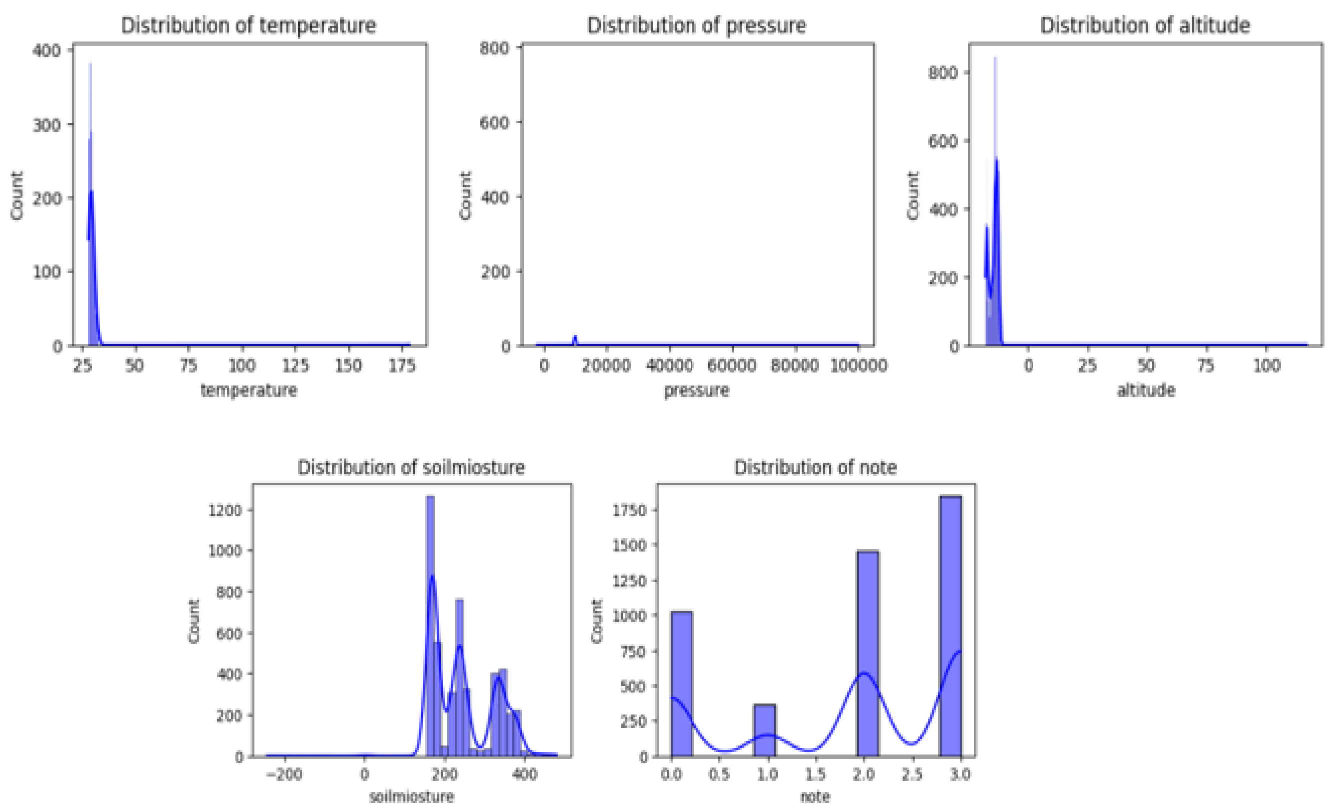

Fig.7 Data distribution

A kernel density curve is added on the top of the histograms, thereby improving the visualization. The titles of each subplot clearly indicate the feature being visualized, helping the user quickly identify which feature's distribution is being examined. Figure 7 illustrates the data distribution, depicting how various data points are spread across different categories or values. The intensity and direction of correlations between various features in the dataset are represented clearly in Figure 8. Correlation coefficients are present in every cell; they range from -1 to 1, with a perfect positive correlation represented by one and a negative correlation denoted by -1. The temperature feature shows very little correlation with other variables, as most of its values are close to zero, except for a weak positive relationship with altitude. Pressure has a moderate positive correlation with altitude. In contrast, soil moisture exhibits a notable negative correlation with status, meaning higher soil moisture values are associated with lower note and status values. This analysis aids in identifying key feature interactions, which can be important for model development and understanding the underlying data patterns. Feature relationships with the target variable help to understand how individual features correlate with the output or class label, as shown in Figure 9. Analyzing these relationships can identify which features provide a strong correlation for predicting the target variable and which might have a weaker or no impact. Understanding these relationships is crucial for feature selection, model training, and improving predictive accuracy.

Fig. 8 Heat map visualization of data

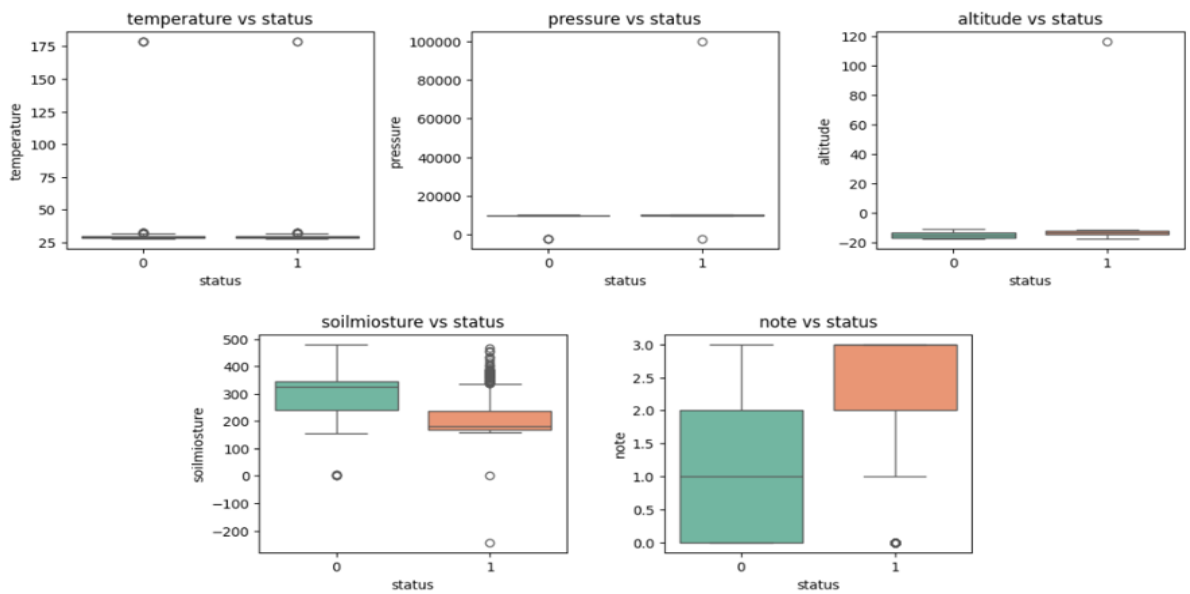

Fig. 9 Feature representation with target variable

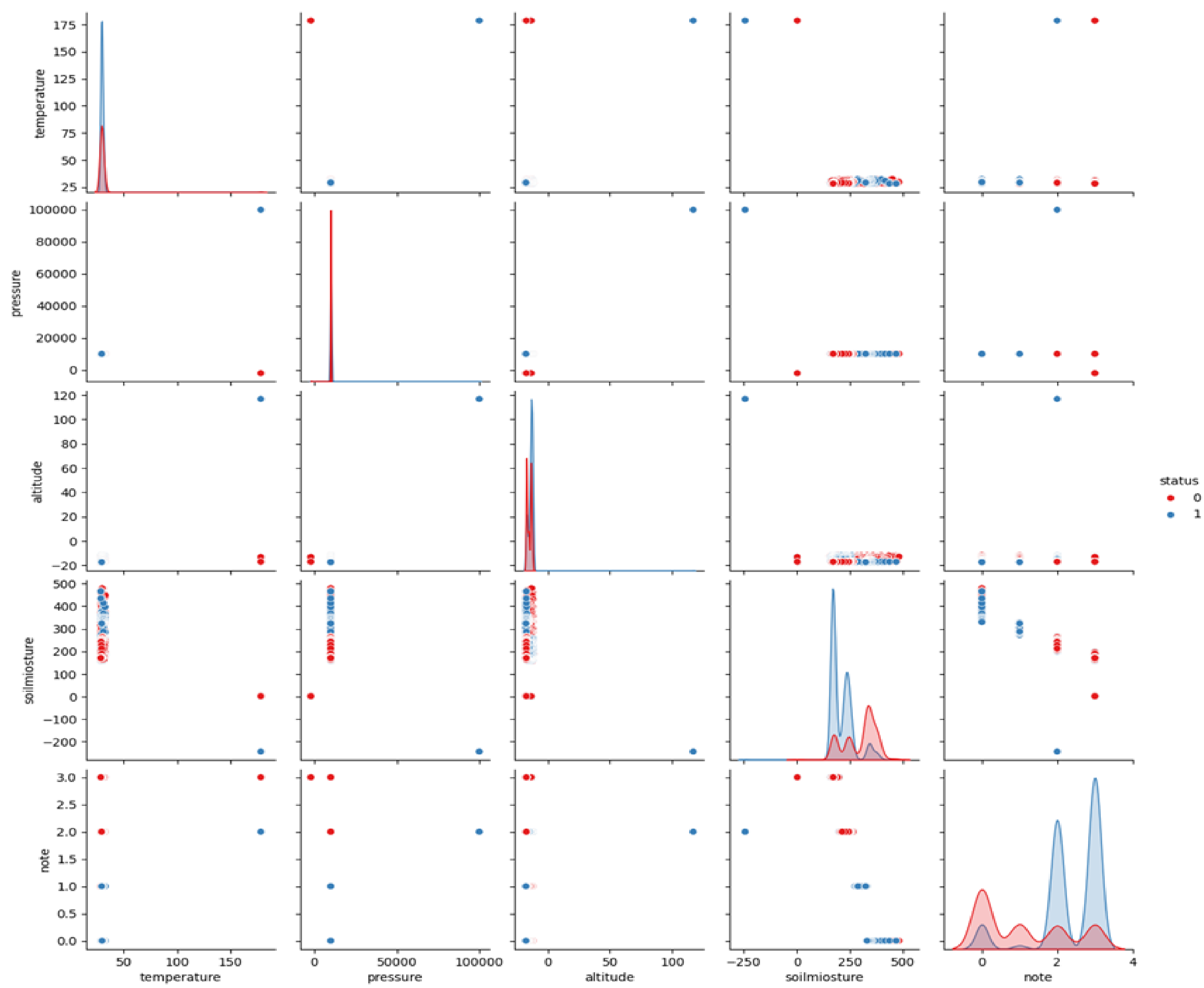

Fig. 10 Pair plot of the feature set

The pair plot in Figure 10 displays pairwise relationships between features in the dataset, with the "status" feature serving as the hue to differentiate between the two. Each scatterplot and kernel density plot in the matrix reveals how each feature interacts with one another while also providing insight into the distribution of individual features along the diagonal. For instance, the distribution of "soil moisture" shows distinct density peaks for each class, whereas other feature combinations like "pressure" and "temperature" or "altitude" show minimal variations between the two classes, suggesting that these features might not provide significant differentiation between the "status" groups.

3.3. Model Development

The proposed model is a hybrid architecture that combines 1-D CNN with Multi-Head Attention mechanisms. The 1-D CNN served as the initial feature extraction layer, where the input data was passed through multiple convolutional layers. These layers applied filters to get local patterns in the data. The output of these convolutional layers was then flattened to prepare it for the next stage of the model.

Following the 1-D CNN layers, the model incorporated a mechanism of Multi-Head Attention. By reshaping the flattened output into a sequence format, this approach enabled the attention layers to concurrently focus on various portions of the input data. Ultimately, fully connected layers processed the data from the attention layers to generate a final binary prediction indicating whether irrigation was required.

3.3.1. 1D CNN Model

A 1D Convolutional Neural Network is effective for sequence data, such as time series or sequentially ordered data. Unlike 2D CNNs, which operate on two-dimensional data, 1D CNNs apply convolutional operations along a single dimension [20]. Below is a detailed description of a typical 1D CNN architecture, including the key components for each layer. Figure 11 illustrates the architecture of the 1D CNN.

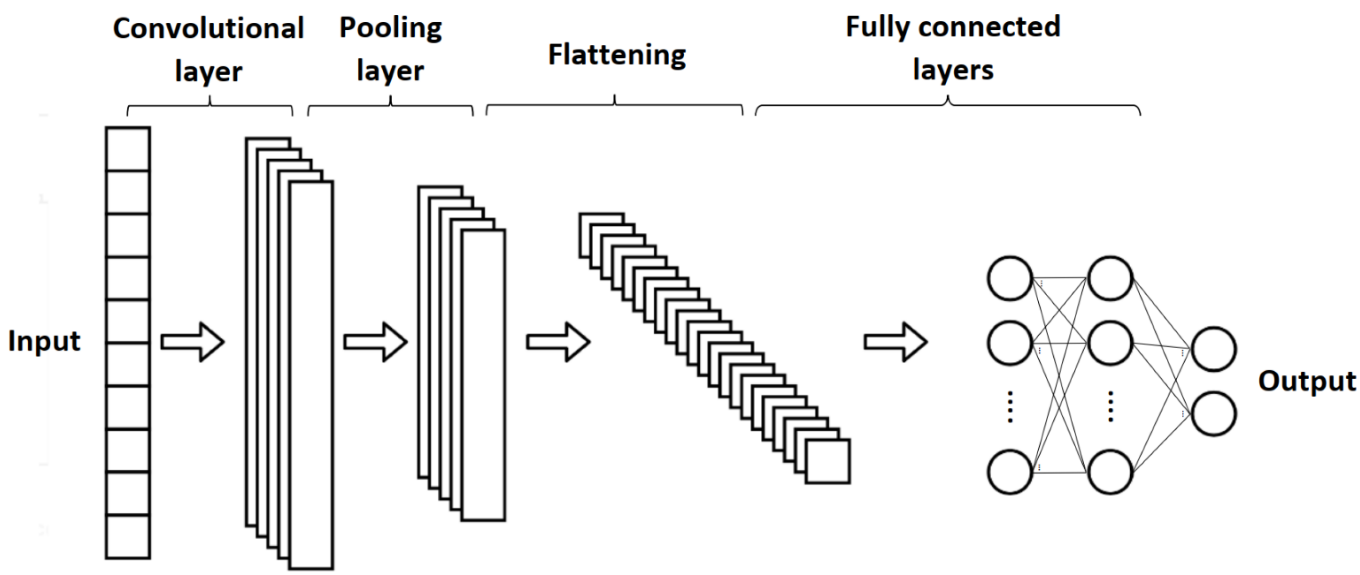

Fig. 11 1-D CNN architecture

The data is entered into the network through the input layer. For a 1D CNN, the input is typically a sequence of data points arranged in a one-dimensional format. This could be time series data, sensor readings, or any other sequential data. The shape of the input tensor is generally batch size, sequence length, and channels, where batch size denotes the number of samples processed simultaneously, sequence length denotes how many time steps there were in the input sequence, and channels indicate how many features there were at each time step. The convolutional layer is a CNN key component that trains the network on how to identify local characteristics in the input data. For a 1D convolution, the output Y is calculated as per Equation (1):

$$Y[n] = \sum_{k-0}^{k-1} W[k].X[n+k] + b \quad (1)$$

where X is the input data, W denotes the weights of the convolutional filter, b is the bias term, K is the filter size, and n indicates the position in the output feature map. Through activation functions, as expressed in Equation (2), the network gains non-linearity.

$$A[n] = \max(0, Y[n]) \quad (2)$$

By lowering the spatial size of the feature map, pooling layers downsample the data, and the pooling operation is expressed in Equation (3). This reduces the likelihood of overfitting, computation, and the number of parameters. Max Pooling is the commonly employed pooling technique in 1D CNNs.

$$P[n] = \max(A[i: i + S] \quad (3)$$

Where $A[i: i + S]$ is the section of the input feature map covered by the pooling window of size S. Feature maps must be transformed into a one-dimensional vector after the convolutional and pooling layers in order to be passed to the fully connected layers, the process is known as flattening. Dense layers are conventional neural network layers where high-level reasoning occurs in the network. All neurons in one layer are connected to all other neurons in the layer before it, and its operation is denoted in Equation (4).

$$Z = WX + b \quad (4)$$

Where the input vector from the flattening layer is denoted by X, the output is denoted by Z, W measures the weight. The dropout layer helps to avoid overfitting during training by randomly setting a portion of the input to zero, which is expressed in Equation (5).

$$O = \frac{X}{1-P} \quad (5)$$

Where X is the input, and this scaling ensures that the expected output remains the same during training and testing. The output layer delivers the final predictions. Depending on the task (classification or regression), this layer uses different activation functions. Softmax Function represented in Equation (6).

$$\mathrm{P(y = k|x)} = \frac{e^{Z_k}}{\sum_j e^{Z_J}} \quad (6)$$

where Z_k is the output of the dense layer for class k. The output logits are transformed into probabilities using the softmax method.

3.3.2. Multi-Head Attention Model

Transformer models depend on this mechanism as they simultaneously focus on different parts of the input sequence. By means of multiple sets of attention weights, or 'heads', this mechanism enables the model to identify diverse relationships and patterns in the data [21]. Figure 12 presents the general multi-head attention model design.

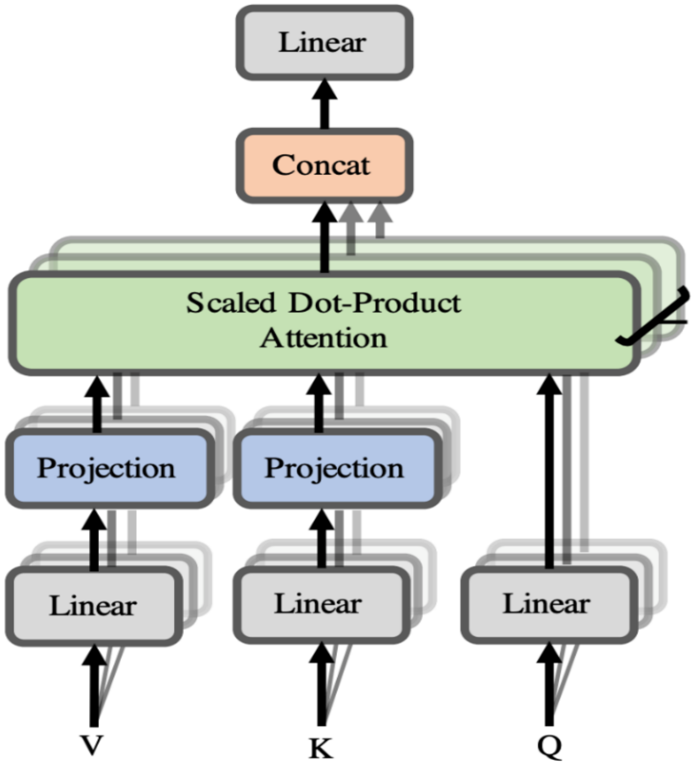

Fig. 12 Multi-head attention general block diagram

The mechanism divides the input data into several 'heads', each of which processes the data separately. The final output is produced by concurrently concatenating and linearly transforming the output from these heads. This method enables the model to learn different input data aspects, enhancing the representational power. The input sequence is represented as a matrix X, where every row represents a token in the sequence. The input shape is represented in Equation (7).

$$X \in R^{n \times d} \tag{7}$$

Where the dimensionality of the input number is denoted by d, and the input sequence is represented by n. For each head h, three different representations, namely the Key (K), Query (Q), and Value (V) matrices, are computed by applying a linear transformation to the input X as expressed from Equation (8) to Equation (10).

$$Q_h = XW_Q^h \tag{8}$$

$$K_h = XW_K^h \tag{9}$$

$$V_h = XW_V^h \tag{10}$$

W_Q^h, W_K^h, W_V^h are learnable weight matrices for head h. The overall shapes are $Q_h \in R^{n \times d_k}$, $K_h \in R^{n \times d_k}$, $V_h \in R^{n \times d_k}$. Attention scores is calculated for each head h, and applying the softmax function as denoted in Equation (11).

$$Attention(Q_h, K_h, V_h) = \text{Softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h \tag{11}$$

Where the dot product $Q_h K_h^T$ computes the attention score between query and key, the scaling factor $\sqrt{d_k}$ is used to stabilize gradients during training, the softmax function normalizes the scores, turning them into probabilities. After calculating the attention outputs for each head, concatenate the outputs and apply a final linear transformation as denoted by Equation (12).

$$Multihead(Q, K, V) = Concat(head_1, head_h)W_o \tag{12}$$

Where each head $head_h$ is computed as per Equation (13).

$$head_h = Attention(Q_h, K_h, V_h) \tag{13}$$

W_o is the learnable weight matrix.

3.3.3. Proposed 1D CNN-Multiple Attention Head Hybrid Model

The proposed hybrid model begins with a 1 D CNN block with an input layer and a series of convolutional layers. The first layer had 64 filters with a kernel size of 3, activated by the ReLU, and a kernel size of 3. It was followed by a max-pooling operation to reduce the dimensionality. This was repeated in the second convolutional layer, which contained 128 filters. The CNN block helps to extract useful features from the input data, and its output was flattened for further processing. After the flattened layer in the CNN block, a multi-head attention mechanism was applied. The flattened output from the CNN was reshaped into a sequence format, making it suitable for attention mechanisms.The Multi-Head Attention layer, with four heads, allows the capture of relationships between various features by attending to different parts of the input simultaneously. This added another abstraction layer to the model by highlighting key dependencies between the features that enhance predictive accuracy. Table 2 gives an overview of the hyperparameters utilized in the study.

Table 2. Hyperparameters

Hyper Parameter	Value
Optimizer	Adam
Learning rate	0.0001
Epoch	100
Loss	Binary cross entropy
Activation Function	ReLU, Sigmoid
Batch size	32
Total parameters: 313,601 Trainable parameters: 313,601 Non-trainable params: 0	

Figure 13 illustrates the model architecture used in the study, showcasing the various layers and components contributing to the irrigation prediction process. The Algorithm for the model is given in Algorithm 1.

3.4. Hardware and Software Setup

The proposed hybrid 1D CNN-MHA model was implemented using Google Collaboratory as the workstation platform. This platform was chosen for its flexibility, accessibility, and support for parallel processing, which is crucial for training DL models on large datasets. Colab's integration with Google Drive also allows for seamless data storage and retrieval during model training and evaluation.

The libraries, combined with Python's robust community support, provided the necessary tools to implement, train, and evaluate the hybrid model efficiently. For the components, the Keras library, integrated with TensorFlow as the backend, was employed. TensorFlow's flexibility, combined with its GPU support, ensured that the model could be trained efficiently, while Keras provided an intuitive framework for defining the suggested architecture. This combination of Python, Google Colab, Keras, and TensorFlow offered an efficient and scalable setup for developing and deploying the hybrid model.

Algorithm 1: Irrigation Prediction Using 1D CNN-Multi-Head Attention Hybrid Model

---

Input: Irrigation scheduling data in CSV format
Output: Prediction of Irrigation (Irrigated or Not-Irrigated)

Begin:
Load and preprocessing data

1. Dataset collection: C= $\{(A_i, b_i)$, where A_i →irrigation scheduling data and $b_i \in \{0,1\}$ (0: Not Irrigated, 1: Irrigated).
2. Preprocess:
   - Standardize: Apply Standard Scaler () to normalize the data
     $$A_i' \rightarrow \frac{A_i - \mu}{\sigma}$$
   - Reshape: Reshape the input data
     $$X_{train} \rightarrow (X_{train}.\text{shape}[0], X_{train}.\text{shape}[1], 1)$$
     $$X_{test} \rightarrow (X_{test}.\text{shape}[0], X_{test}.\text{shape}[1], 1)$$

Define Base Model:

1. Input: $\text{none} \times 5 \times 1$
2. Initial Convolutional Layer: Conv2D (64, (3,3); activation=’relu’)
3. Maxpooling2D (pool size= (2; stride=2)
4. Convolutional Layer: Conv2D (128, (3,3); activation=’relu’)
5. Maxpooling2D (pool size= (2, stride=2)
6. Flatten ()
7. Reshape ()
8. Apply Multi-Head Attention: (num_heads = 4, key_dim = 128)
9. Flatten ()
10. Fully Connected Layers:
    a. Dense (128, activation=’relu')
    b. Dropout (0.5)
    c. Dense (64, activation=' relu')
    d. Dense (1, activation=' sigmoid')

Compilation and Training:

1. Compile each model M:
   loss=Binary_crossentropy
   learning rate=0.0001
   optimizer=Adam ()
2. Evaluate model performance on test data
3. Predict irrigation status for new data

Save the Model
End

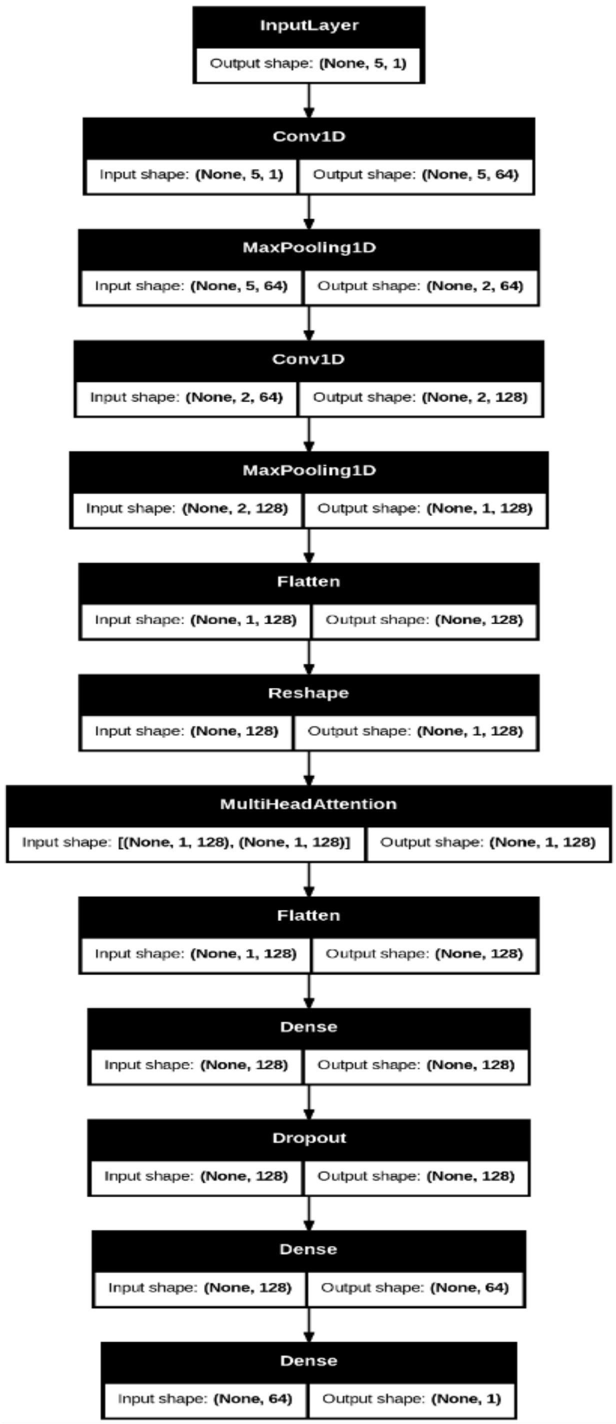

Fig. 13 Model architecture

4. Results and Discussion

Understanding the effectiveness and learning trends of the suggested model depends on the accuracy and loss plots. On both the training and validation datasets, the accuracy plot graphically illustrates the model's capacity to consistently predict data labels throughout training iterations. An accuracy plot, as illustrated in Figure 14, depicts the model's accuracy during training and validation over epochs. Throughout the epochs, the model continues to improve, with accuracy gradually increasing and validation loss decreasing. By epoch 10, accuracy rises to 91.10%, while the validation loss drops to 0.2178, with a validation accuracy of 92.13%. Between epochs 15 and 20, the model stabilizes with an accuracy of around 92-93%, but fluctuations in validation loss and accuracy indicate some overfitting. By epoch 30, the model reaches a high accuracy of 94.55%, with a validation accuracy of 95.47%.

After epoch 40, the accuracy peaks at 95.57%, though validation performance fluctuates, reaching a high of 96.00% in epoch 47. However, overfitting is evident in the later stages of training as the validation loss increases while accuracy remains stable around 95-96%. The final few epochs show slight decreases in validation performance, suggesting the model may have reached its optimal point around epoch 50 despite continuing improvements in training accuracy. A loss plot, as shown in Figure 15, represents the loss value of the model during training and validation across epochs. It helps assess the model's learning process, where a decreasing loss indicates better performance and convergence towards optimal weights.

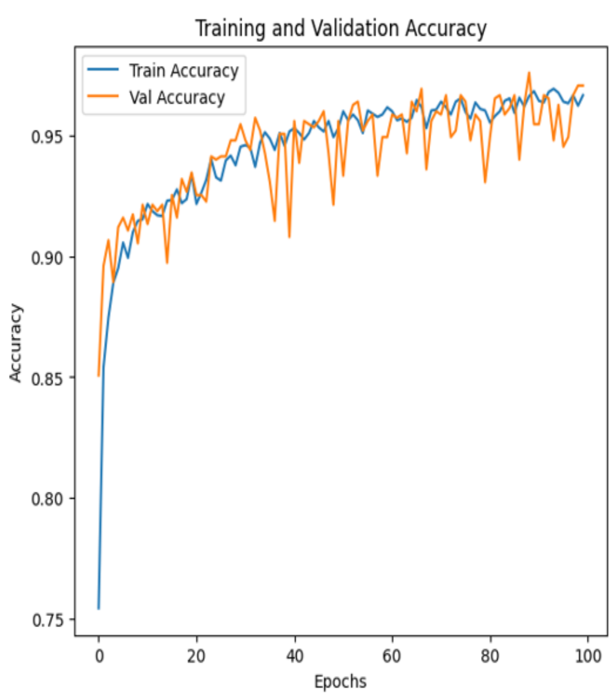

Fig. 14 Accuracy plot of the proposed system

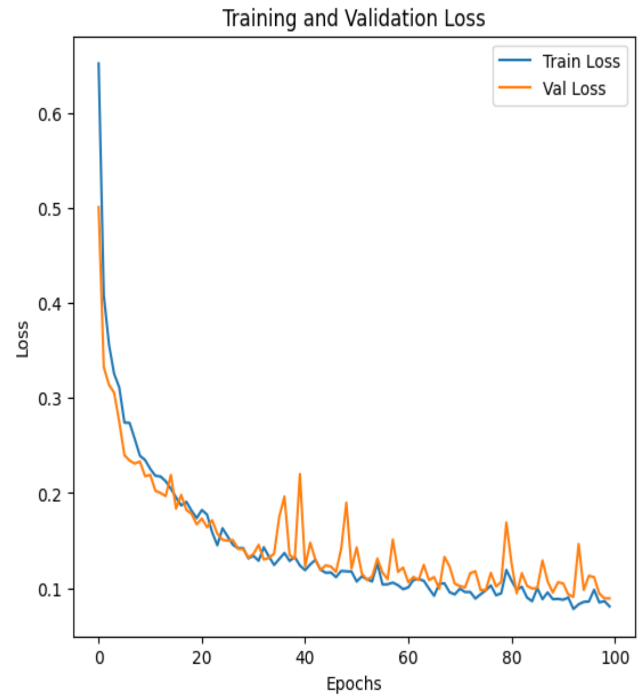

Fig. 15 Loss plot of the proposed system

Initially, the accuracy starts at 66.52% with a loss of 0.6809, improving steadily with each epoch. By epoch 10, the accuracy reaches 91.10% with a reduced loss of 0.2396 and the validation accuracy peaks at 92.13% with a loss of 0.2178. This trend continues with occasional fluctuations, and by epoch 20, the accuracy reaches 93.83% with a loss of 0.1648, and the validation accuracy reaches 93.47% with a loss of 0.1671. Further epochs show a stable trend with improvements, reaching 95.68% accuracy and 96.00% validation accuracy in the later stages. There are occasional increases in validation loss, suggesting slight overfitting at certain points, but the model consistently shows high performance.

A confusion matrix helps assess how well a model distinguishes between different classes, allowing for a detailed analysis of its performance, as shown in Figure 16. The model accurately classifies 572 instances as irrigated while misclassifies 26 instances as not irrigated. Similarly, it correctly identifies 335 instances as not irrigated but incorrectly predicts 5 instances as irrigated. Performance indicators obtained from the confusion matrix provide a comprehensive assessment of the proposed model's effectiveness. To comprehensively assess the effectiveness of the proposed approach, the four principal metrics employed are F1-score, accuracy, precision, and recall. These metrics, grounded in the principles of False Positive (FP), False Negative (FN), True Negative (TN), and True Positive (TP), are crucial for evaluating the model's efficacy. The mathematical formulations for these performance parameters are shown in Equations (14)-(17).

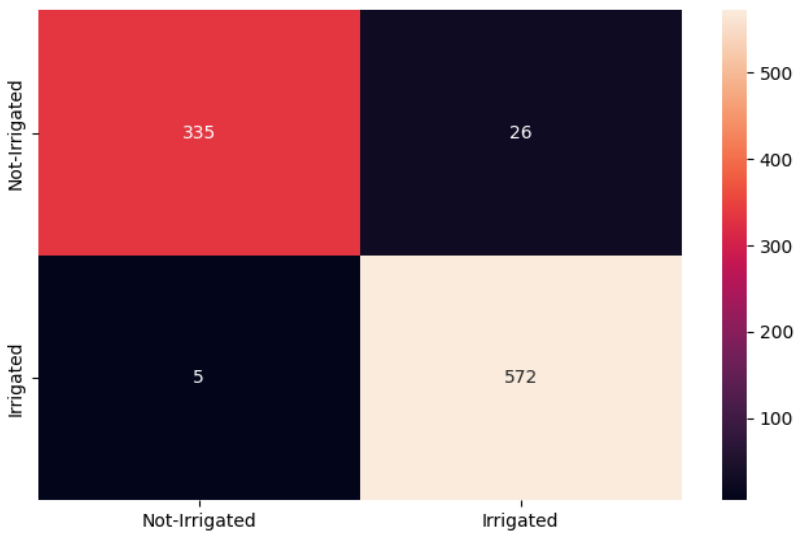

Fig. 16 Confusion matrix of the proposed system

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

$$F1-score = 2 \times \frac{precision \times Recall}{Precision+Recall} \quad (17)$$

The classification report in Table 3 demonstrates strong model performance with a high overall accuracy of 97%. For the "Irrigated" class, the model achieved a precision of 0.96 and a recall of 0.99. The "Not-Irrigated" class showed even higher precision at 0.99, though its recall was slightly lower at 0.93, suggesting the model missed some not-irrigated instances. Both the macro and weighted averages for precision, recall and F1-Score were around 0.96 and 0.97, reflecting strong overall performance, with the weighted average considering the class distribution.

Table 3. Classification report

Class	Precision	F1-Score	Recall
Irrigated	0.96	0.97	0.99
Not-Irrigated	0.99	0.96	0.93
Weighted-average	0.97	0.97	0.97
Macro-average	0.97	0.96	0.96
Accuracy: 0.97			

The Receiver Operating Characteristic (ROC) demonstrates the trade-off between the TP and FP rates across various threshold conditions. The Area Under the Curve (AUC) signifies the overall performance of the model, where an AUC of 1.0 specifies a perfect model with an FP rate of '0' and a TP rate of '1'. The ROC curve, as shown in Figure 17, depicts the efficiency of the CNN-MultiHead Attention Model in distinguishing between classes across various threshold settings. Table 4 and Figure 18 compare the performance of the suggested method with those of existing approaches. The performance comparison of various models for irrigation prediction reveals that the proposed hybrid model, integrating 1D CNN with Multi-Head Attention, significantly outperforms other techniques. While traditional models like Decision Tree (89.68%) and Logistic Regression (89.45%) show moderate accuracy, more advanced methods like Multiple Linear Regression (92.11%) and LSTM (93.34%) improve prediction performance. However, the proposed hybrid model achieves the highest accuracy at 97%, surpassing all others. This exceptional performance results from the 1D CNN's proficiency in capturing spatial information from sequential data and the Multi-Head Attention's ability to emphasize pertinent temporal and contextual patterns, which standard approaches inadequately handle. The hybrid approach ensures more accurate feature extraction and classification, making it highly suitable for irrigation monitoring in the context of complex and variable environmental conditions.

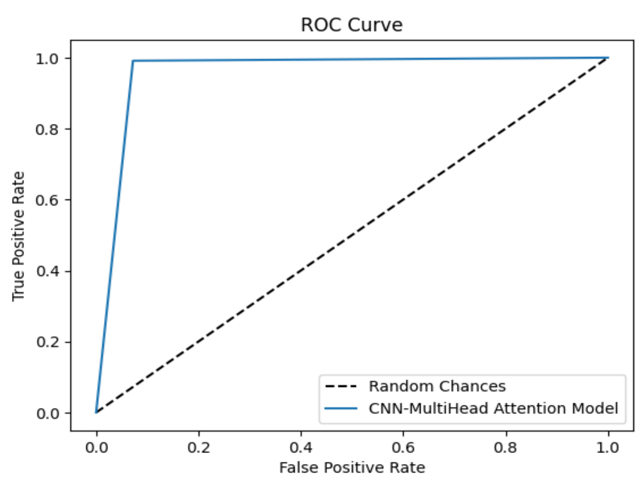

Fig. 17 ROC of the proposed system

Table 4. Performance comparison with existing methods

Author	Model	Accuracy (%)
Behzadipour et al [24]	Logistic regression	89.45
Esmail et al [11]	Decision tree algorithm	89.68
El Bilali [22]	Multiple Linear Regression	92.11
Togneri et al [23]	LSTM	93.34
Proposed hybrid model of 1D CNN with multi-head attention model		**97**

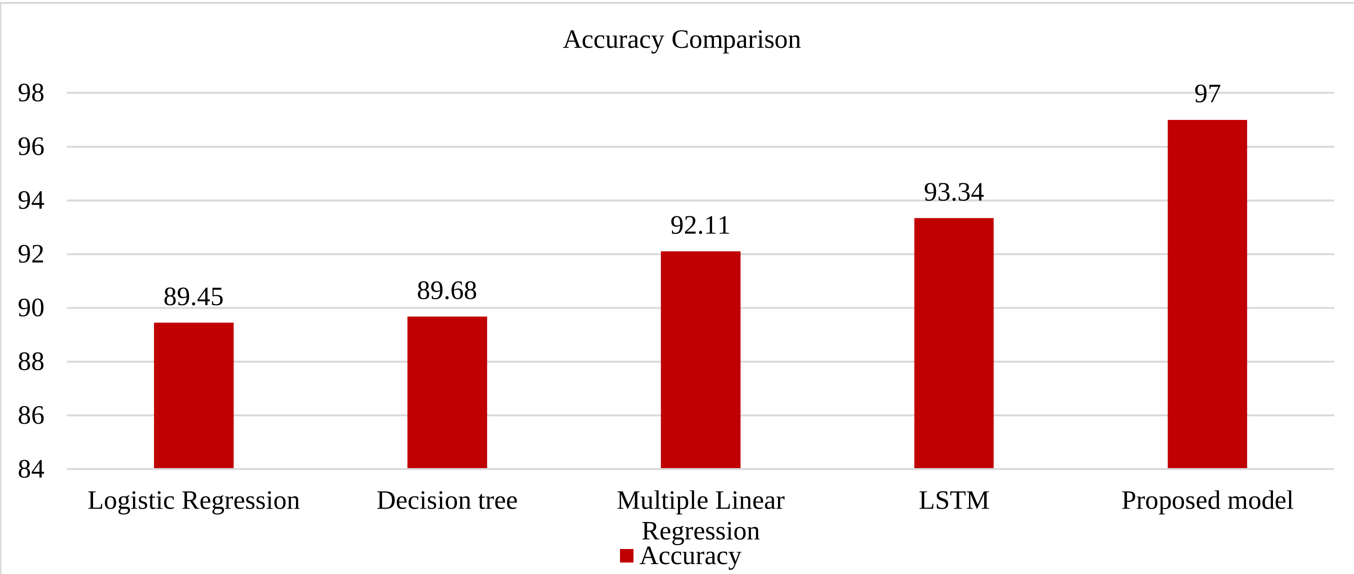

Fig. 18 Performance comparison with existing methods

5. Conclusion

The growing global population and increasing demand for food necessitate innovative approaches to agricultural management, particularly in the realm of irrigation practices. Accurate classification of irrigated and non-irrigated land plays a critical role in understanding land use patterns and optimizing irrigation strategies. This study successfully demonstrates the effectiveness of a 1D CNN-MultiHead Attention model in classifying irrigated and non-irrigated land, addressing the pressing need for efficient agricultural resource management in the context of a growing global population. The high-performance metrics, 97% accuracy, alongside precision values of 96.9% for irrigated areas and 99% for non-irrigated areas, validate the model's capability to accurately differentiate between the two land classes. The research offers significant contributions to sustainable agriculture by employing advanced DL techniques, emphasizing the importance of optimizing irrigation practices and enhancing crop yields. The findings underscore the model's practical implications for agricultural stakeholders, providing them with valuable insights for informed decision-making. Future work could explore the integration of additional data sources and refinement of the model to further improve classification performance and expand its applicability to various agricultural settings. This research lays a foundation for leveraging innovative technologies in promoting sustainable land use and resource management practices in agriculture.

Acknowledgements

The author conveys deep gratitude to the supervisor for offering direction and steadfast support during this study.

References

[1] Muhammad Farooq et al., “Physiological and Agronomic Approaches for Improving Water-Use Efficiency in Crop Plants,” *Agricultural Water Management*, vol. 219, pp. 95-108, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Bryan O'Hara, *No-Till Intensive Vegetable Culture: Pesticide-Free Methods for Restoring Soil and Growing Nutrient-Rich, High-Yielding Crops*, Chelsea Green Publishing, 2020. [Google Scholar] [Publisher Link]

[3] Ariel Dinar, Amanda Tieu, and Helen Huynh, “Water Scarcity Impacts on Global Food Production,” *Global Food Security*, vol. 23, pp. 212-226, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Saima Nasreena, and Muhammad Aqeel Ashraf, “Inadequate Supply of Water in Agriculture Sector of Pakistan Due to Depleting Water Reservoirs and Redundant Irrigation System,” *Water Conservation & Management*, vol. 5, no. 1, pp. 13-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Luxon Nhamo et al., "Improving the Accuracy of Remotely Sensed Irrigated Areas Using Post-Classification Enhancement through UAV Capability," *Remote Sensing*, vol. 10, no. 5, pp. 1-12, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Peter H. Gleick, "A Look at Twenty-First Century Water Resources Development," *Water International*, vol. 25, no. 1, pp. 127-138, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ravneet Kaur Sidhu et al., "Automation in Drip Irrigation for Enhancing Water Use Efficiency in Cereal Systems of South Asia: Status and Prospects," *Advances in Agronomy*, vol. 167, pp. 247-300, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Sumit Ray, and Sitabhra Majumder, "Water Management in Agriculture: Innovations for Efficient Irrigation," *Modern Agronomy*, pp. 169-185, 2024. [Google Scholar]

[9] Yan-An Chen et al., "An Ensemble Learning Model for Agricultural Irrigation Prediction," *International Conference on Information Networking*, Jeju Island, Korea (South), pp. 311-316, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Rajesh Kannan Megalingam et al., "Irrigation Monitoring and Prediction System Using Machine Learning," *International Conference for Emerging Technology*, Belgaum, India, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ahmed A. Esmail et al., "Smart Irrigation System Using IoT and Machine Learning Methods," *5th Novel Intelligent and Leading Emerging Sciences Conference*, Giza, Egypt, pp. 362-367, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Pankaj Kumar Kashyap et al., "Towards Precision Agriculture: IoT-Enabled Intelligent Irrigation Systems Using Deep Learning Neural Network," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17479-17491, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ali Risheh, Amirmohammad Jalili, and Ehsan Nazerfard, "Smart Irrigation IoT Solution Using Transfer Learning for Neural Networks," *10th International Conference on Computer and Knowledge Engineering*, pp. 342-349, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Khalil Ibrahim Mohammad Abuzanouneh, "Design of Machine Learning Based Smart Irrigation System for Precision Agriculture," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 109-124, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Vijay et al., "An IoT Instrumented Smart Agricultural Monitoring and Irrigation System," *International Conference on Artificial Intelligence and Signal Processing*, Amaravati, India, pp. 1-4, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Premkumar, and A.N. Sigappi, "IoT-Enabled Edge Computing Model for Smart Irrigation System," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 632-650, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] El Mehdi Raouhi et al., "AIDSII: An AI-Based Digital System for Intelligent Irrigation," *Software Impacts*, vol. 17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Hussein M. Al-Ghobari et al., "Prediction of Wind Drift and Evaporation Losses from Sprinkler Irrigation Using Neural Network and Multiple Regression Techniques," *Agricultural Water Management*, vol.195, pp. 211-221, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Triptmann, Irrigation Scheduling for Smart Agriculture, Kaggle, 2023. [Online]. Available: https://www.kaggle.com/code/triptmann/irrigation-scheduling-for-smart-agriculture

[20] Liying Liu, and Yain-Whar Si, "1D Convolutional Neural Networks for Chart Pattern Classification in Financial Time Series," *The Journal of Supercomputing*, vol. 78, no. 12, pp. 14191-14214, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Yuming Lin et al., "Multi-Head Self-Attention Transformation Networks for Aspect-Based Sentiment Analysis," *IEEE Access*, vol. 9, pp. 8762-8770, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ali El Bilali, and Abdeslam Taleb, "Prediction of Irrigation Water Quality Parameters Using Machine Learning Models in a Semi-Arid Environment," *Journal of the Saudi Society of Agricultural Sciences*, vol. 19, no. 7, pp. 439-451, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rodrigo Togneri et al., "Soil Moisture Forecast for Smart Irrigation: The Primetime for Machine Learning," *Expert Systems with Applications*, vol. 207, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Faeze Behzadipour et al., "A Smart IoT-Based Irrigation System Design Using AI and Prediction Model," *Neural Computing and Applications*, vol. 35, no. 35, pp. 24843-24857, 2023. [CrossRef] [Google Scholar] [Publisher Link]