Precision Irrigation with Cost-effective and Autonomic IoT Devices using Artificial Intelligence at the Edge

D3.1

Design of the OSIRRIS light-weight AI irrigation model

Responsible Editor: Innotec21

Contributors: Waziup e.V.

Document Reference: D3.1

Distribution: Public

Version: 1.0

Date: May, 2023

Table of Contents

1. Introduction

The “Osirris AI irrigation model” is a crucial component of the overall Osirris Irrigation System, as it allows accurate and reliable **forecasts of the soil moisture**. It will facilitate informed decision-making by the farmer. The AI model will be trained using the data collected during the data collection phase, and also external farming data. It will then be integrated into the Osirris Irrigation System and tested in real conditions.

The irrigation model will predict the soil moisture content in the near future, enabling proactive and efficient water management practices. In addition to soil moisture prediction, the model incorporates advanced capabilities to forecast the number of days remaining before the soil reaches a critically dry state for a specific crop. Furthermore, it provides insights into the required amount of water that needs to be added to the field to maintain optimal soil moisture levels.

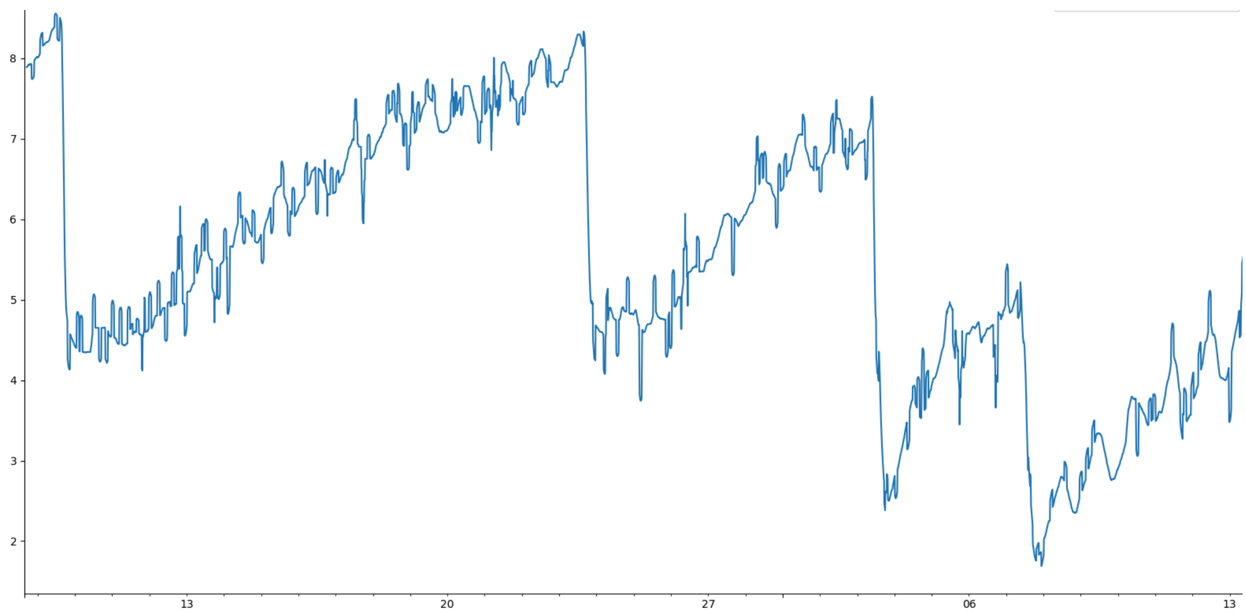

Figure 1: Sample of soil tension over time for apple tree in the region of Sbeïtla (Sandy soil)

To better understand the dynamics of soil moisture variation, the Figure above illustrates the measurements of soil water tension recorded by the soil tensiometer. The curve displayed in the diagram depicts the upward trend, indicative of soil drying, along with daily fluctuations. These variations reflect the natural processes of water evaporation and plant water uptake that contribute to the overall soil moisture dynamics.

The downward "slopes" observed in the curve represent the irrigation process, during which water is intentionally added to the field to replenish the soil moisture levels. These irrigation events play a vital role in sustaining adequate soil moisture for optimal plant growth and yield. By carefully analyzing the pattern of irrigation shots, the model can derive valuable insights into the frequency and quantity of water required for effective irrigation practices.

The model's ability to estimate the amount of water needed to maintain optimal soil moisture levels further enhances its practical utility. By considering factors such as weather conditions, crop evapotranspiration rates, and soil characteristics, the model provides valuable guidance on the precise quantity of water required for irrigation. This information aids in optimizing water resource management, promoting sustainable agricultural practices, and reducing water wastage.

For developing the AI model, our team has envisioned an **Agile approach**. We have planned to release multiple versions/iterations of the AI model, each aimed at enhancing and refining its predictive capabilities. Each version of the AI model will include additional data sources. Through careful analysis and validation of the models' outputs, we can identify areas for improvement and implement necessary adjustments to enhance accuracy, robustness, and generalization capabilities.

2. State of the Art

The development of advanced irrigation systems that can accurately predict soil tension and optimize water usage is an active area of research and innovation. Some of the key state-of-the-art approaches and resources in this context include:

Sensor Integration and Data Fusion

Integrating soil moisture and temperature sensors with weather data is a core component of predictive irrigation systems. Research has explored techniques for fusing these heterogeneous data sources to improve soil tension forecasting. For example, Karimi et al. [1] proposed a data fusion framework that combines soil moisture, temperature, and weather forecast data using Kalman filtering to predict soil tension up to 3 days in advance. Shelia et al. [2] developed a model that integrates soil sensor data with weather forecasts using machine learning to predict soil water potential and optimize irrigation scheduling.

Machine Learning for Predictive Modeling

Leveraging machine learning techniques is crucial for building accurate soil tension prediction models. Researchers have explored various algorithms and architectures, for example Bai et al. [3] used a deep neural network to predict soil moisture based on soil properties, weather data, and historical soil moisture observations. Liang et al. [4] compared the performance of different machine learning models, including random forests and support vector machines, for forecasting soil water potential.

Edge Computing and IoT

Integrating edge computing and Internet of Things (IoT) technologies enables real-time data processing and decision-making at the irrigation system level. This can improve responsiveness and reduce the need for cloud connectivity. For example Jia et al. [5] developed an IoT-based smart irrigation system that uses edge computing to process sensor data and optimize water application. Kamilaris et al. reviewed the use of IoT and edge computing in precision agriculture, including applications for predictive irrigation.

Optimization and Control Strategies

Research has also focused on developing optimization algorithms and control strategies to efficiently manage irrigation based on predicted soil tension. For example Haghverdi et al. proposed a model predictive control approach that uses soil moisture forecasts to optimize irrigation scheduling and reduce water usage. Dukes et al. reviewed the use of soil moisture sensors and control algorithms to automate and optimize irrigation systems.

By integrating these state-of-the-art techniques, it is possible to create advanced irrigation systems that can accurately predict soil tension, optimize water usage, and adapt to changing environmental conditions. The combination of sensor data, machine learning, edge computing, and optimization algorithms is a promising approach for developing the next generation of smart irrigation solutions.

[1] Karimi, Y., Prasher, S. O., Patel, R. M., & Kim, S. H. (2008). Application of support vector machine technology for the prediction of crop yield under variability in climate and management factors. Transactions of the ASABE, 51(1), 301-308.

[2] Shelia, V., Šimůnek, J., Boote, K., & Hoogenbooom, G. (2018). Coupling DSSAT and HYDRUS-1D for simulations of soil moisture and nitrate dynamics in the soil-plant-atmosphere system. Journal of Hydrology and Hydromechanics, 66(2), 232-245.

[3] Bai, Y., Feng, M., Hao, X., Wang, J., Qin, F., & Liang, T. (2018). Evaluation of soil moisture prediction from AMSR2 satellite data using distributed temperature sensing and in-situ observations. Remote Sensing, 10(1), 10.

[4] Liang, W. L., Kosugi, K. I., & Mizuyama, T. (2011). A three-dimensional model of the effect of stemflow on soil water dynamics around a tree on a hillslope. Journal of Hydrology, 400(1-2), 1-14.

[5] Jia, Q., Zhou, J., Wang, H., Cao, D., & Shen, C. (2020). An IoT-based greenhouse environment monitoring system for improving management strategies. Journal of Sensors, 2020.

Kamilaris, A., Kartakoullis, A., & Prenafeta-Boldú, F. X. (2017). A review on the practice of big data analysis in agriculture. Computers and Electronics in Agriculture, 143, 23-37.

Haghverdi, A., Leib, B. G., Washington-Allen, R. A., Ayers, P. D., & Buschermohle, M. J. (2016). Perspectives on delineating management zones for variable rate irrigation. Computers and Electronics in Agriculture, 123, 10-19.

Dukes, M. D., Zotarelli, L., Liu, G. D., & Simonne, E. H. (2012). Principles and practices of irrigation managemenIch war t for vegetables. Horticultural Sciences Department, UF/IFAS Extension.

Citations:

[1] https://www.waternewseurope.com/smart-irrigation-saves-water-with-state-of-the-art-technology/

[2] https://vro.agriculture.vic.gov.au/dpi/vro/vrosite.nsf/pages/lwm_state_art_irrigation_ed2

[3] https://www.oliveoiltimes.com/business/state-of-the-art-irrigation-management-leads-to-rising-yields-in-california/123109

[4] https://edepot.wur.nl/121330

[5] https://www.farmprogress.com/farming-equipment/the-irrigation-system-of-the-future

3. Model definition

3.1. First version

The first version of the model will include only the past soil tension, as measured by the sensors deployed in the field. As can be seen in the following Figure, it will perform a simple regression analysis in order to identify the “drying” trend. It will then calculate the number of days before the “over-dry” threshold will be reached. This threshold will be entered manually by the farmer.

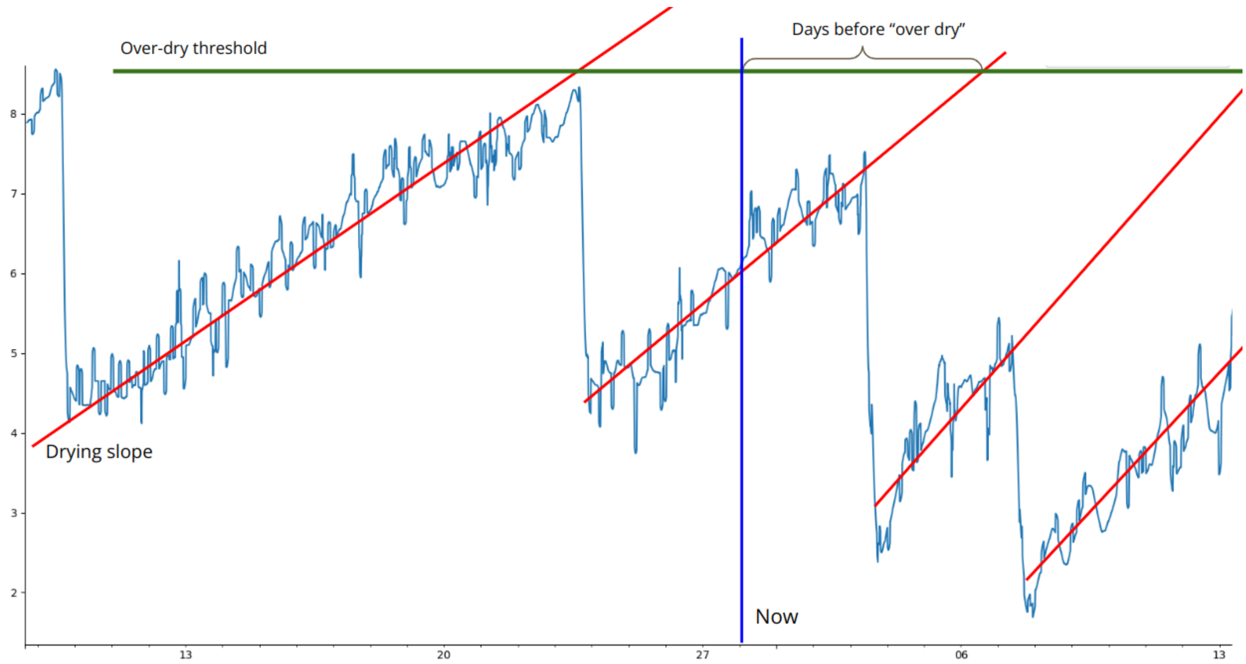

Figure 1: The first iteration of the system predicts the “days before over dry”

The model establishes an "over-dry" threshold specific to the crop under consideration. The green line in the diagram represents this threshold, calculated based on crop-specific parameters and optimal moisture requirements. The objective of the model is to accurately calculate the trend of soil drying over time and provide predictions regarding the number of days it will take for the soil to reach the critical "over-dry" threshold.

3.2. Second version

The second iteration of the prototype is aimed to refine the predictions. To achieve that, we **include a weather forecast** to improve the predictions of the model. Now we can adjust the prediction according to the weather forecast, which can, in the case of an accurate forecast, improve the predictions by a lot. Other potentially important feature to include in our dataset are:

- Crop coefficient Kc: A property of plants used in predicting evapotranspiration (ET). Evapotranspiration is the process by which water is transferred from the land to the atmosphere

through evaporation and plant transpiration. Kc is the most basic crop coefficient calculated as ETc / ETo
- ETc: The evapotranspiration rate observed in the crop being studied. This value is inputted by the farmer based on observations of the plant status.
- ETo: The evapotranspiration rate observed for a well calibrated reference crop under the same conditions. This value could come from a database or literature tables.

Those metrics might be confusing to a farmer, that is why we did not include them in our model yet. We want to evaluate how big of a difference it makes to omit and include those values. From literature these values seem to have a big influence on the performance of a regression model.

We also want to investigate the impact of including "lag features" to our training, former recorded drying of the soil could be beneficial for the model, afterwards we will evaluate if there are improvements according to different metrics.

3.3. Input data for model creation

The data to create the model contains: soil tension, soil temperature and the data from a weather station. For every field we are trying to estimate the soil water content, we install four sensor devices, to create an average and account for spatial differences of the soil condition in the field. The following data will be used:

- **Soil tension:** Soil tension, also known as soil water tension or soil matric potential, refers to the force or suction present in the soil due to water. Soil tension is measured in kilopascals (kPa) or bars and indicates the soil's moisture content. It measures the soil's ability to retain and hold water against gravity. If there are several sensors in the same field, the values must be averaged.
- **Soil temperature:** Soil temperature is a crucial factor that influences soil tension, which is the force or suction in the soil due to water. Changes in soil temperature can affect the soil's ability to retain and hold water against gravity, thereby impacting soil tension and moisture dynamics. As soil temperature increases, the water-holding capacity of the soil typically decreases. Warmer temperatures can accelerate evaporation and increase the rate of water movement from the soil to the atmosphere, leading to drier soil and higher positive soil tensions.
- **Air Temperature and Humidity:** Outdoor air temperature and humidity levels play a crucial role in determining the rate of evapotranspiration, which is the combined water loss from soil and plant surfaces. Warmer temperatures and lower humidity increase evapotranspiration, potentially resulting in drier soil.
- **Relative and Absolute Pressure:** Changes in air pressure can impact soil moisture by affecting the movement and distribution of atmospheric moisture. Low-pressure systems are associated with increased chances of precipitation, which can replenish soil moisture. Conversely, high-pressure systems often indicate drier weather conditions.
- **Wind Speed, Gust, and Direction:** Wind can accelerate the drying of soil by increasing evaporation rates. Higher wind speeds enhance moisture loss from the soil surface, potentially leading to decreased soil moisture content. Wind direction can also influence soil moisture patterns, particularly in relation to topography and vegetation.
- **Dew Point and Wind Factor:** Dew point is the temperature at which air becomes saturated, leading to the formation of dew or frost. It indirectly reflects the moisture content of the air. The wind factor, which combines wind speed and temperature, affects the rate of evaporation from the soil surface.

- **Precipitation:** Precipitation measurements, such as last hour, last 24 hours, weekly, monthly, and total precipitation, directly impact soil moisture levels. Rainfall adds moisture to the soil, increasing its moisture content. Monitoring precipitation patterns helps in predicting soil moisture changes and adjusting irrigation schedules accordingly.
- **Light Intensity (Lux) and UV Index (UVI):** Light intensity and UV index provide information about solar radiation levels. Solar radiation influences evaporation rates, plant transpiration, and overall water demand in the ecosystem. Higher light intensity and UV levels increase evapotranspiration, potentially affecting soil moisture.

The metrics mentioned above are just the raw input. We also create more features from them, by performing mean or rolling windows over all sensors available or e.g. engineering features from the current date. The techniques for data preprocessing are mentioned in the fifth bullet point.

The data will be retrieved from the installed sensors during the data collection phase (see report D5.1). Additional data can be collected from online resources, such as the International Soil Moisture Network[1].

3.4. The over-dry threshold

According to Shock[2], the soil water tension can be closely related to the stress experienced by plant tissues. It thus makes sense to set up a threshold over which irrigation should take place. As mentioned in a paper from the University of Arkansas[3], **a conservative over-dry threshold is 60 centibar average for silt loams and clays and 20-35 cb for sandy soils**. Those values will be used for the first version of the system.

In our system, we will include a table to calculate the over-dry threshold based on:

- Crop type
- Crop growth stage

We will use the thresholds given in D1.2.

3.5. Requirements

In the following there is an overview about functional requirements:

Id	Version	Requirements	Comments
1	V1	The Osirris Irrigation model should be able to **predict the soil tension** for a few days ahead.	
2	V1	The Osirris Irrigation model should be able to start making predictions **after a short data gathering time** of two to three days,	Necessary to create accurate predictions
3		The Osirris Irrigation model should be able to predict the time at which the soil will be **"over-dry"**.	See figure 1.

[1] https://ismn.earth/en/
[2] https://journals.ashs.org/hortsci/view/journals/hortsci/46/2/article-p178.xml
[3] https://www.uaex.uada.edu/publications/pdf/FSA58.pdf

4	V1	The Osirris Irrigation model should take into account the latest field information hourly.	Pipeline will retrain the model to account for the latest data.
5	V1	The Osirris Irrigation model should take **weather data** into account.	First iteration: historical, Second: forecast
6	V2	The Osirris Irrigation model should also take the **weather forecast** into account to create even more accurate predictions.	Will take closest weather station, according to GPS coordinates
7	V2	The Osirris Irrigation model should be integrated into an application e.g. a **WaziApp**, that has got an User Interface, to visualize the predictions.	
8	V2	The Osirris Irrigation model should allow the farmer to manually set the **"over-dry" threshold.**	Should be possible to just numerically change it and also automatically calculated, according to soil type, plant type and climatic conditions.
9	V3	The Osirris Irrigation model should be able to **calculate the "over-dry"** threshold for a given crop, soil and climatic conditions.	

Table 1: Functional requirements

In the following there is an overview about the non-functional requirements:

1	V1	The Osirris Irrigation model should be able to be trained on a Raspberry Pi 4 with 4GB of RAM.	RPI 3 is optional.
2	V1	The Osirris Irrigation model should be able to be trained below one hour, with all its refinement and data pre- and post processing steps.	This also includes all pre- and post processing steps.
3	V1	The Osirris Irrigation model should accurately demonstrate the core functionality of the final product. It should perform the essential tasks and operations that the final product is intended to carry out.	-

4	V1	The Osirris Irrigation model must maintain a high level of reliability, minimizing errors and system failures.	Needs to be a stable and reliable product. With measures to prevent unintended behavior.

Table 2: Non-functional requirements

4. Model development

4.1. Model pipeline

The architecture of the pipeline can be seen in the Figure below.

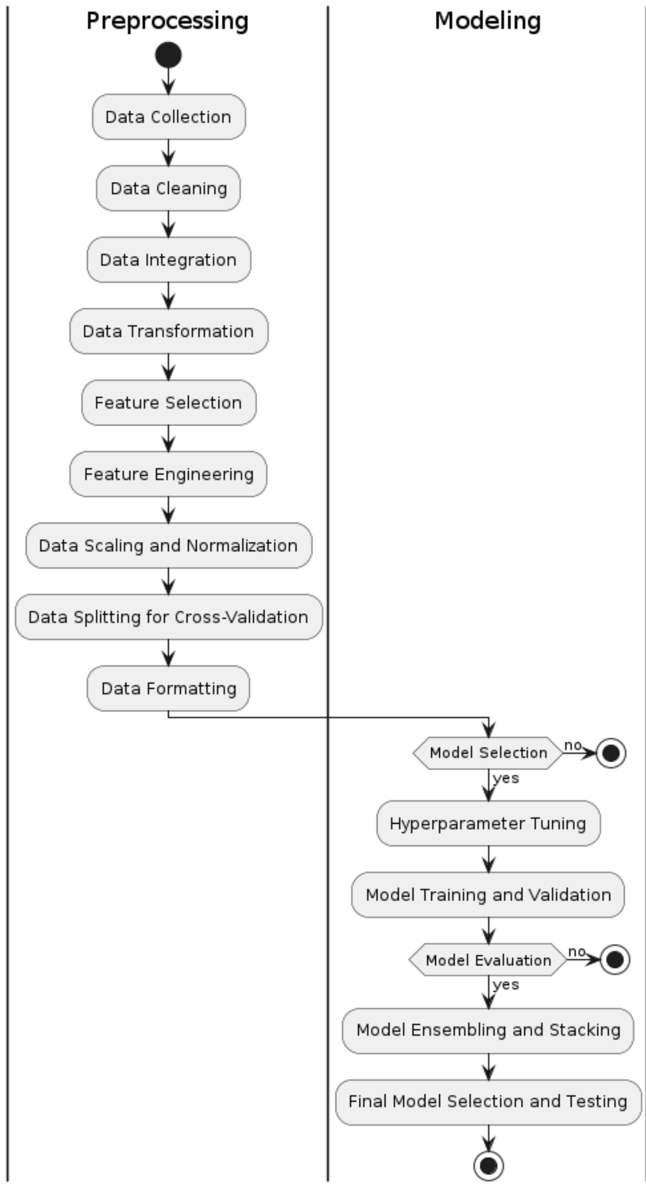

Figure 2: Activity Diagram of Preprocessing and Modeling

4.2. Data preparation/preprocessing

Data preparation, also known as data preprocessing, is a crucial step in the data analysis pipeline. It involves transforming raw data into a format suitable for analysis and modeling.

For us it is especially important to perform data preprocessing, since we need to automatically prepare the collected data in a meaningful way. Hence we have to automatically identify which specific approach can yield good results.

Here are the key steps that we used to perform data preparation:

1. **Data Collection**: Gather the required data from various sources such as weather APIs for historical and forecasts. We ensure that the data collected aligns with our analysis goals.
2. **Data Cleaning:** We cleaned the data to address issues like missing values, outliers, duplicates, and inconsistencies. This step ensures the data quality and integrity. Techniques for data cleaning include imputation for missing values, removing duplicates, and handling outliers.
3. **Data Integration:** Our analysis requires combining data from multiple sources, performing data integration. We identified the common fields or keys to merge our datasets correctly. We used techniques like joins and merges to combine the data.
4. **Data Transformation:** We also did transform the data to improve its quality and make it more suitable for analysis. This step involves normalization, scaling and handling skewed distributions. Some common transformations include log transformations, standardization.
5. **Feature Selection:** Identify the most relevant features that contribute significantly to the analysis task. Feature selection techniques help reduce dimensionality, improve model performance, and minimize computational requirements.
6. **Feature Engineering:** We created new features and derived additional meaningful information from the existing features. This step involves mathematical operations, aggregations, binning, or applying domain knowledge to create relevant features.
7. **Splitting the Data:** We divided the dataset into training, validation, and testing sets. The training set is used to train the model, the validation set helps tune model parameters, and the testing set evaluates the model's performance on unseen data.
8. **Data Scaling and Normalization:** We scale or normalize the data to ensure that all features are on a similar scale. We decided on min-max normalization, where all features are scaled into a range from 0 to 1.
9. **Data Splitting for Cross-Validation:** We use cross-validation for model evaluation, splitting data into multiple folds proved to be helpful. This technique helps estimate the model's performance on unseen data more robustly.
10. **Data Formatting:** Formatting the data according to the requirements of the analysis or modeling technique you plan to use. For example, certain algorithms may require data in a specific format such as matrices or sequences.
11. **Data Validation:** Perform final checks to ensure the data is ready for analysis. Verify that all the preprocessing steps have been correctly applied and the data is in the desired format.

By following these steps, we can prepare the data for analysis or modeling, ensuring that it is clean, reliable and suitable for the specific task to be performed.

4.3. Model creation

To train, compare, evaluate and further improve the suggested machine learning models we use the python package PyCaret as a fast developing environment. The PyCaret pipeline offers a comprehensive framework for training time series forecasting models while automating several essential steps. The pipeline encompasses a series of stages, each contributing to the overall process of model training and evaluation. The following is a detailed description of the steps involved in the PyCaret time series forecasting pipeline:

1. **Model Selection:** PyCaret provides an automated approach to select the best-performing forecasting models based on various evaluation metrics. The pipeline explores a range of candidate algorithms suitable for time series forecasting, such as ARIMA, SARIMAX, Prophet, XGBoost, LightGBM, etc. It conducts an internal evaluation to identify the most promising models for further optimization.
2. **Hyperparameter Tuning:** Once the initial set of models is identified, PyCaret employs an automated hyperparameter tuning process to optimize the selected models. It explores different combinations of hyperparameters using techniques like grid search, random search, or Bayesian optimization, aiming to find the configuration that yields the best performance.
3. **Model Training and Validation:** After hyperparameter tuning, the selected models are trained on the training dataset using the optimized settings. Cross-validation techniques, such as time series cross-validation or rolling window validation, are commonly employed to evaluate the models' performance on multiple folds of data. This helps assess the model's stability and ability to generalize to unseen future data.
4. **Model Evaluation:** PyCaret provides a comprehensive evaluation of trained models using various performance metrics specific to time series forecasting, such as mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and others. These metrics quantify the accuracy and precision of the models in predicting future time steps.
5. **Model Ensembling and Stacking:** To further improve the forecasting performance, PyCaret offers ensemble techniques such as stacking or blending. These techniques combine the predictions of multiple models, leveraging their individual strengths to generate more accurate and robust forecasts.
6. **Final Model Selection and Testing:** Based on the evaluation results, the pipeline helps identify the best-performing model or ensemble for deployment. The selected model is then tested on the unseen testing dataset to assess its performance in a real-world scenario.
7. **Model Deployment:** Once the final model is selected and validated, PyCaret enables easy deployment of the model in various production environments. This includes generating prediction outputs, setting up an API for real-time predictions, or exporting the model for integration into other applications or systems.

The PyCaret time series forecasting pipeline streamlines the entire process of building, evaluating, and deploying time series forecasting models. By automating crucial steps, it reduces the manual effort required and enables efficient and standardized model development, ultimately facilitating more accurate and reliable predictions.

5. Evaluation and Validation

The Osirris Irrigation Model will be evaluated by comparing its predictions against actual, measured soil moisture data. Two parameters will be evaluated:

1. Soil moisture trend prediction
2. Days before irrigation recommendation

The Soil moisture trend prediction will be evaluated using the following technique. The first step is to split the soil moisture dataset into training and testing subsets. This ensures that the model is evaluated on unseen future data. The splitting is done at 80% for training, 20% for testing. The Evaluation Metrics used to compare the prediction with the reality will be root mean squared error (RMSE).

The irrigation recommendation will be evaluated by comparing it with the actual irrigation performed by the farmer. Additionally, we will ask agronomic experts to evaluate the predictions and recommendations provided by the system. We will use their feedback to adjust the system.

The evaluation and validation pipeline in PyCaret for a time series forecasting model involves a series of steps to assess the model's performance and validate its predictive capabilities. Each step is designed to provide comprehensive insights into the model's accuracy, robustness, and ability to generalize to unseen future data. The following is a detailed description of the evaluation and validation steps in PyCaret for a time series forecasting model:

1. **Splitting the Data:** The first step is to split the time series dataset into training and testing subsets. This ensures that the model is evaluated on unseen future data. The splitting is done while preserving the temporal order of the data, typically by using a cutoff date or index.
2. **Cross-Validation Techniques:** PyCaret employs various cross-validation techniques to evaluate the model's performance on multiple folds of the training data. Time series cross-validation methods, such as rolling window validation or expanding window validation, are commonly used. These techniques ensure that the model is tested on data that follows the temporal order, simulating real-world forecasting scenarios.
3. **Training the Model:** The model is trained on the training dataset using the selected algorithm and optimized hyperparameters. The training process involves fitting the model to the historical data, capturing the underlying patterns and relationships to make accurate future predictions.
4. **Model Evaluation Metrics:** PyCaret provides a comprehensive set of evaluation metrics specifically tailored for time series forecasting. These metrics assess the accuracy and precision of the model's predictions against the true values. Common metrics used in time series forecasting include mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and others. These metrics quantify the model's performance in terms of prediction errors and help gauge its effectiveness.
5. **Backtesting and Forecast Evaluation:** After training the model, PyCaret facilitates backtesting and forecast evaluation to validate its performance. Backtesting involves comparing the model's predictions against the known historical values to assess its ability to accurately capture past patterns. Forecast evaluation is performed by comparing the model's predictions on the testing dataset with the actual future values. This evaluation provides insights into the model's predictive accuracy for unseen data points.
6. **Visualizations and Plots:** PyCaret offers a range of visualizations and plots to aid in the evaluation and validation process. Time series plots can show the model's predicted values overlaid with the actual values, allowing for visual comparison. Residual plots can help assess the

distribution and patterns of the prediction errors. These visualizations provide a deeper understanding of the model's performance and assist in identifying any systematic biases or anomalies.

7. **Model Comparison:** PyCaret allows for the comparison of multiple models and their respective evaluation metrics. This comparison helps identify the best-performing model based on the selected evaluation metrics and provides insights into the strengths and weaknesses of different models. The comparison facilitates informed decision-making in model selection.
8. **Sensitivity Analysis:** PyCaret enables sensitivity analysis by evaluating the model's performance under different scenarios or parameter settings. This analysis helps assess the model's stability and robustness and provides insights into how the model performs under varying conditions.
9. **Documentation and Reporting:** PyCaret offers tools to generate comprehensive reports and documentation summarizing the evaluation and validation results. These reports capture the evaluation metrics, visualizations, and model comparison information, providing a clear and concise overview of the model's performance.

The evaluation and validation pipeline in PyCaret for a time series forecasting model ensures a thorough assessment of the model's accuracy, reliability, and suitability for real-world forecasting tasks. By leveraging a range of evaluation metrics, visualization techniques, and cross-validation methods, PyCaret facilitates informed decision-making and enables the selection of the most effective time series forecasting model

6. Conclusion

In conclusion, the report presents the design and implementation of a precision irrigation system using cost-effective and autonomic IoT devices with artificial intelligence at the edge. The system aims to predict soil moisture content and provide insights for efficient water management in various fields, including agriculture, hydrology, and environmental monitoring.

The report highlights the iterative nature of the pipeline design, which allows for continuous improvement and adaptation to evolving requirements and advancements in the field. By releasing multiple versions of the pipeline, the team can incorporate new methodologies, algorithms, and data sources to enhance the accuracy, robustness, and generalization capabilities of the prediction models. User feedback, industry best practices, and advancements in machine learning techniques are leveraged to proactively overcome limitations and challenges, resulting in increasingly precise and reliable soil moisture predictions.

The irrigation model described in the report offers valuable insights into soil moisture dynamics, forecasting the number of days until the soil reaches a critically dry state and providing guidance on the required amount of water for optimal moisture levels. By leveraging environmental factors, such as weather predictions, the model corrects the drying rate for more accurate predictions. These capabilities empower farmers and agricultural stakeholders to make informed decisions regarding irrigation scheduling, leading to efficient water application, reduced crop stress, improved yield, and sustainable resource management.

The report also emphasizes the importance of understanding contributing factors to soil drying, including climate and weather patterns, rainfall patterns, soil type and structure, vegetation cover, land management practices, topography and slope, groundwater level, and human factors. By considering these factors, appropriate soil moisture conservation strategies, efficient irrigation practices, and sustainable land management approaches can be implemented to mitigate soil drying and maintain optimal soil moisture levels.

Data preparation and preprocessing are outlined as crucial steps in the analysis pipeline, involving data collection, cleaning, integration, transformation, feature selection and engineering, splitting, scaling, normalization, handling imbalanced data, and formatting. These steps ensure that the data used for analysis and modeling is clean, reliable, and suitable for the specific task.

The training phase involves using the PyCaret package as a fast developing environment for training time series forecasting models. The pipeline encompasses model selection, hyperparameter tuning, training, validation, evaluation, ensembling and stacking, final model selection and testing, and deployment. By automating these steps, PyCaret streamlines the process of model development and facilitates accurate and reliable predictions.

The evaluation and validation pipeline in PyCaret for time series forecasting models includes splitting the data, employing cross-validation techniques, training the model, evaluating performance using specific metrics, and conducting backtesting and forecast evaluation. These steps ensure comprehensive assessment of the model's accuracy, robustness, and generalization capabilities.

In summary, the precision irrigation system described in the report offers a cost-effective and autonomic solution for efficient water management using artificial intelligence at the edge. By continuously improving the prediction models, considering contributing factors to soil drying, and following a rigorous data preparation and training process, the system enables informed decision-making, sustainable resource management, and advancements in agricultural and environmental sciences.