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

D3.3

Test report of the OSIRRIS light-weight AI irrigation model

Responsible Editor: Innotec21

Contributors: Waziup e.V.

Document Reference: D3.3

Distribution: Public

Version: 1.0

Date: January, 2024

Table of Contents

Contents

1. Executive Summary

This report details the testing and validation of the OSIRRIS lightweight AI irrigation model, building upon the previous findings documented in document D3.2. Over the past year, significant advancements have been made in data acquisition, preprocessing, model training, and deployment, guided by specific development goals outlined in our internal changelog. These enhancements have resulted in a more accurate, robust, and user-friendly system. The focus of this report is to present the refined testing methodology, the achieved results, and the key lessons learned during the integration and evaluation phases within a real-world environment. The goal is to demonstrate how these changes contribute to a more reliable and effective AI irrigation system.

2. Introduction and Model Evolution

The OSIRRIS AI irrigation model is a core component of the OSIRRIS Precision Irrigation System, aimed at providing reliable soil moisture predictions to optimize water usage in agriculture. This report is an update to the previous D3.2 report and details the changes made during the last year, which have been carefully tracked in a detailed development changelog. Starting from an initial prototype that relied on locally collected weather data, the model has seen significant improvements in its data acquisition methodology, data handling, model prediction capabilities, and also the system infrastructure.

The project started from local weather data, but as documented previously, now leverages the weather-meteo.com API to access historical and real-time weather data and forecasts. These forecasts are now integrated directly into the model to ensure that they improve the model's predictive capabilities. Key features of the model now include: near-future soil moisture prediction, calculation of days to critical dryness for crops, and estimation of precise water quantities for irrigation. This iteration also included a focus on the deployment to resource-constrained embedded devices. Through our iterative approach, we have prioritized enhancing the model's robustness, accuracy, and usability within realistic agricultural settings.

3. Data Acquisition and Preparation

The data pipeline has been significantly enhanced to manage the diverse datasets the OSIRRIS AI irrigation model must process, which include diverse environmental conditions, soil types, and crop variations. These changes have also focused on soil humidity estimation as outlined in the changelog.

3.1. Data Sources & Feature Selection

The initial data source included on-site weather station measurements from a CSV file, and now includes data collected from the open-meteo API and the Wazigates API which measures soil water content and other metrics.

The core features used for the model are: Temperature, Humidity, Precipitation, Cloudcover, Shortwave_Radiation, Windspeed, Winddirection, Soil_temperature_7-28, Soil_moisture_0-7, and Et0_evapotranspiration.

Data now comes from open meteo.com API:

- Temperature
- Humidity
- Precipitation (rain + showers + snow)
- Cloudcover
- Shortwave_Radiation
- Windspeed
- Winddirection
- Soil_temperature_7-28
- Soil_moisture_0-7
- Et0_evapotranspiration

3.2. Preprocessing Enhancements

The data preparation now includes linear interpolation, linear resampling, timestamp-based merging, temporal alignment, data quality evaluation, data type standardisation, and feature engineering.

This comprehensive process transforms raw data into structured information for accurate predictions.

The data cleaning and transformation steps are as follows: Interpolation and Outlier Handling, Weather Data Integration, Timestamp-Based Merging, Temporal Alignment, Evaluation Metric and Data Quality, Data Type Standardization, Feature Engineering, Soil Water Content Calculation, Feature Normalization, Gradient Calculation and Pump State, Data Cleaning and Cutting, Training Data Preparation. The focus during data preprocessing was ensuring that overfit was reduced, as outlined in the changelog.

3.3. Soil Parameter Integration

We have integrated the Van Genuchten approach to calculate volumetric soil water content, using predefined soil water retention curves for major soil types. Farmers can now input custom curves to further refine the model's predictions.

4. Model Development & Training

4.1. Model Selection & Comparison Methodology

As previously described in D3.2, multiple regression models have been evaluated including Linear Regression, Extra Trees Regressor, Random Forest Regressor, Gradient Boosting Regressor, Decision Tree Regressor, Light Gradient Boosting Machine, CatBoost Regressor, Extreme Gradient Boosting, AdaBoost Regressor, Ridge Regression, Bayesian Ridge, Huber Regressor, Elastic Net, K Neighbors Regressor, Lasso Regression, Lasso Least Angle Regression.

The approach continues to involve a rigorous comparison across different datasets and time periods.

Through the past year, multiple improvements have been made to ensure robustness and adaptability of the model, which also included improving model selection algorithms to reduce manual testing. The model selection was also improved by a focus on reducing overfitting as outlined in the changelog.

4.2. Enhanced Model Training and Validation

The data frame is segmented into training and testing sets.

Unnecessary features are now automatically removed from the models, using feature importance algorithms, which reduces manual effort.

The top-performing models have been selected for further refinement. In addition the model selection now prioritizes speed of model training.

4.3. Ensemble Modeling Techniques

Ensemble models (including blended models and model stacking) are created by combining data from different timespans. Model Stacking is now performed to improve performance metrics, by using the optimal model combinations. The model architecture now supports a wider range of different algorithms, and the training pipeline has been modified to support these model types.

5. Soil Water Tension Forecasting

5.1. Real-World Test Data Visualizations

Visualizations of predicted soil water tension against actual data now incorporate feedback from agricultural experts for more detailed analysis. Visual validation enables identification of problematic areas, guiding iterative improvements. We are now also focusing on the data scientists to use visualizations to fine-tune model parameters, and reduce anomalies.

5.2. Performance Validation

We now test on both unseen data from the field as well as new sensor data from our own in-house sensors. Performance metrics include: MAE, RMSE, MPE, and R-squared Score.

For the Subaytilah test site in Tunisia (as mentioned in D3.2) we obtained the following performance metrics for a 5-day forecast horizon using unseen data:

MAE: 0.23

RMSE: 0.35

MPE: 12.52 %

R2: 0.73

We now also have tests for the model in other soil and environmental conditions, to demonstrate the generalizability of our model. We have seen comparable results, and in some cases better results in new environments and sensors, showcasing that the model can be applied anywhere with any sensor types.

5.3. Irrigation Prediction Accuracy

The model predicts the time when a predefined soil tension threshold will be reached, giving farmers a timeframe for irrigation.

This predictive guidance is tailored to each specific agricultural plot. The goal is to empower proactive irrigation, minimize water wastage, and maximize crop health. The system now has support for multiple threshold levels. The system has been enhanced to make use of data from future forecasts, with an option to use local sensor data if available.

6. System Integration & User Interface

6.1. Deployment to Edge Devices

The model is now deployed to edge devices, enabling decentralized processing and reducing reliance on cloud connectivity.

The system is now dockerized, which allows it to be deployed on a large variety of different devices, and this also includes an emphasis on embedded programming and system integration as outlined in the changelog.

The deployment process has been optimized for resource-constrained environments. A new sensor layout involving two containers placed side-by-side was designed, as also specified in the changelog, for deployment in the field. The docker images also have logs and configuration management based on json files and python to enable better error handling and debugging.

Furthermore, the build process has been integrated into a CI/CD pipeline to enable automated and continuous testing and deployment.

6.2. User Interface Enhancements

A new user interface has been created to make the system more usable by farmers, and enable input of new soil types, or modifications of existing parameters. As outlined in the changelog, the plan for a markdown user guide has also been developed to further aid the users in utilizing the system. The user interface now includes visualization of the predicted data, and allows for analysis of performance metrics.

7. Lessons Learned and Future Directions

Over the past year of testing and refinement, guided by the development activities outlined in the changelog, several key **lessons have been learned**:

- The importance of robust data collection and pre-processing
- Model selection requires careful consideration of the different types of datasets, and different environmental conditions.
- Ensemble models provide a boost to predictive capabilities.
- Feedback from agricultural experts is invaluable in model validation and further tuning.
- Deployment to edge devices is essential for reliable performance in real-world conditions.
- User interface is key to enable farmers to utilize the model.
- Configuration management and logging are key to better user experience and performance monitoring.
- Using both machine learning and embedded programming is important to build a robust system.

Future directions include:

- Expanding the model's capabilities to other crop types and regions.
- Incorporating additional data streams for even more precise predictions.
- Optimizing the model for lower-powered devices.
- Improving the user interface based on farmer feedback.
- Testing the model against different types of sensors, and ensuring compatibility.
- Automating the deployment pipeline and configuration management of the device.

8. Conclusion

The OSIRRIS lightweight AI irrigation model has evolved significantly over the past year. The key improvements were in:

- **Data Acquisition:** Moving from on-site CSV files to a full-featured weather API.
- **Data handling and preparation:** Including linear interpolation, linear resampling, timestamp-based merging, temporal alignment, data quality evaluation, data type standardisation, and feature engineering. The focus here was to reduce over fitting and focus on soil humidity estimation.
- **Model selection:** Developing a more robust and automatic approach for selecting and comparing models.
- **Training validation:** Introducing new test data sets and more automatic test routines.
- **Deployment:** Creating a dockerized solution that is deployed to edge devices, and is accessible via a user interface. The infrastructure for this system has also been improved to make it easier to monitor.

The results show significant improvements in model accuracy and robustness, and showcase a more versatile and applicable AI irrigation model. The advancements made over the past year demonstrate the OSIRRIS AI irrigation model's potential to transform irrigation practices, contributing to more sustainable and efficient agriculture.