Intelligent Irrigation System for Low-cost Autonomous Water Control in Small-scale Agriculture

Deliverable D2.3a

First report on multi-level & multi-stream agricultural data framework for integration into decision support system

Responsible Editor:	AUA
Contributors:	Dimitrios Giannopoulos
	Georgios Chatzipavlidis

Document Reference:	INTEL-IRRIS D2.3a
Distribution:	Public
Version:	1.1
Date:	January 2023

Contributors table

DOCUMENT SECTION	AUTHOR(S)	REVIEWER(S)
SECTION 1	T. Bartzanas, G. Chatzipavlidis, D. Giannopoulos	C. Pham
SECTION 2	T. Bartzanas, G. Chatzipavlidis, D. Giannopoulos	C. Pham

Document revision history

Version	Date	Changes
V1.1	Jan 30th, 2023	Public release
V1.0	Jan 15th, 2023	First draft version for internal approval
V0.1	Dec 15th, 2022	First release for review

Executive Summary

D2.3a describes the usage of the INTEL-IRRIS sensing platform to build training dataset. It illustrates how the data from the INTEL-IRRIS sensor are collected, processed, and analyzed. It also gives the methodology that will be used for building datasets.

Table of Contents

1. INTRODUCTION

The estimation of soil moisture has gained significant interest amongst researchers recently. The different applications where moisture detection is crucial are hydro-meteorological, agricultural research and climate change. The detection of soil moisture is crucial as it replenishes the water table, controls the penetration of the quantity of water into the soil and is a significant contributor to channel flow and surface runoff. To detect the soil moisture, contact sensors and remote sensing sensors are widely used. Contact sensors require the manual placement of the sensor into the soil (Uthayakumar et al., 2022). Soil moisture affects evapotranspiration and vegetation water availability, which are at the core of the climate-carbon cycle and play an important role in hydrological risks such as floods, drought, erosion, and landslides (Batchu et al., 2022). Agricultural drought has a catalytic effect that contributes to social and political conflicts in developing countries. Therefore, soil moisture modeling and monitoring are of increasing interest. Monitoring the spatial and temporal variations of soil moisture is a prerequisite both for mitigating and adapting to climate changes for the sustainability of cropping systems as well as for developing precision agriculture and food security. Moreover, as the agricultural sector plays a strategic role in the economy, arid countries are more prone to be sensitive to climate events. In arid regions, droughts are recurring climatic events, often threatening agricultural systems and food security (Adab et al.,2020).

2. Data collection, processing & analysis

2.1 Data collection

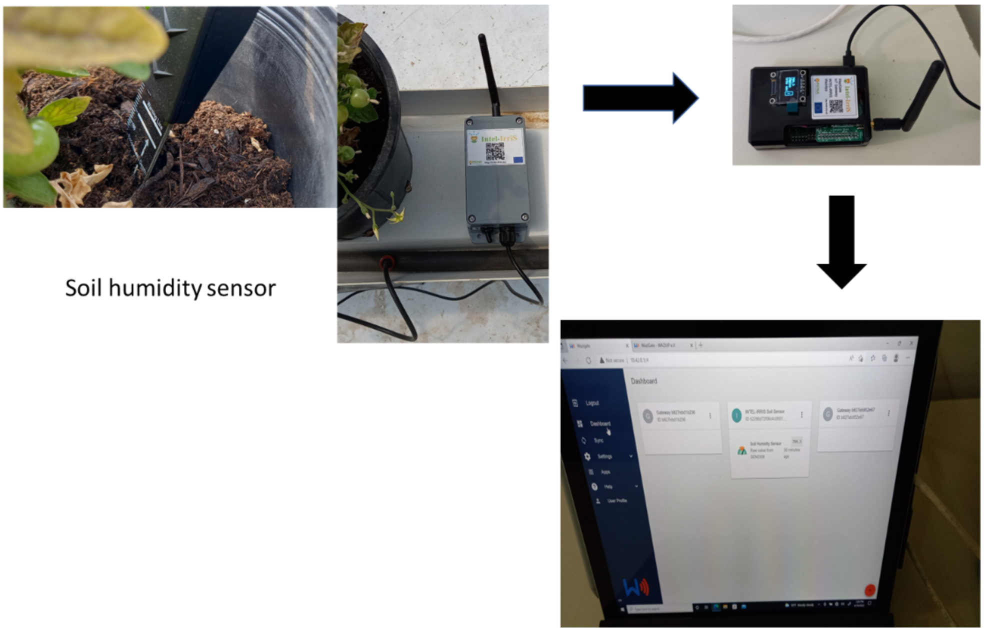


Figure 1: Data collection with INTEL-IRRIS sensing platform

Figure 1 describes how the data were collected from the field using the INTEL-IRRIS sensor and stored at the Wazi-Up platform. Moreover, the platform gives the opportunity to users for quick data visualization as seen in Figure 2.

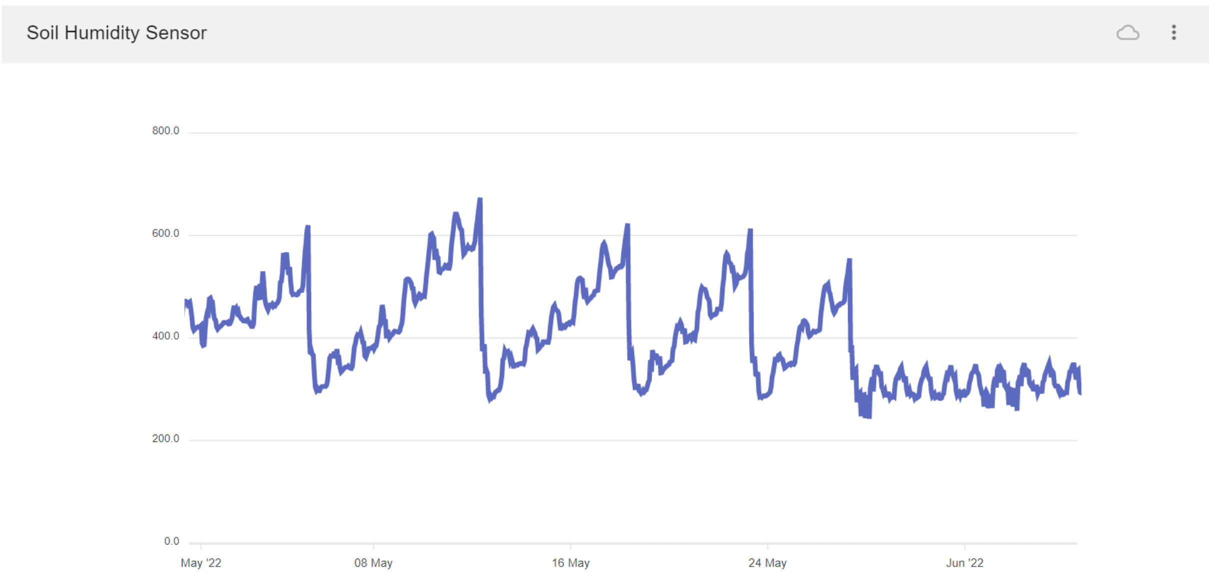


Figure 2: Data visualization

2.2 Data processing

The experimental part of INTEL-IRIS is based on measurements of a series of sensors of industrial and experimental under development. All sensors output analog measurements that are collected as numerical values. These values are defined in the same time series and stored in a single data table. The data in this table will be exploited using SQL. Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL statements to store, update, remove, search, and retrieve information from the database. You can also use SQL to maintain and optimize database performance.

2.2.1 Importance of SQL

SQL is a popular query language that is frequently used in all types of applications. Data analysts and developers learn and use SQL because it integrates well with different programming languages. For example, they can embed SQL queries with the Java programming language to build high-performing data processing applications with major SQL database systems such as Oracle or MS SQL Server. SQL is also easy to learn as it uses common English keywords in its statements. SQL implementation involves a server machine that processes the database queries and returns the results. The SQL process goes through several software components, including the following:

- Parser: The parser starts by tokenizing, or replacing, some of the words in the SQL statement with special symbols. It then checks the statement for the following:
- Correctness: The parser verifies that the SQL statement conforms to SQL semantics, or rules, that ensure the correctness of the query statement. For example, the parser checks if the SQL command ends with a semi-colon. If the semi-colon is missing, the parser returns an error.
- Authorization: The parser also validates that the user running the query has the necessary authorization to manipulate the respective data. For example, only admin users might have the right to delete data.
- Relational engine: The relational engine, or query processor, creates a plan for retrieving, writing, or updating the corresponding data in the most effective manner. For example, it checks for similar queries, reuses previous data manipulation methods, or creates a new one. It writes the plan in an intermediate-level representation of the SQL statement called byte code. Relational databases use byte code to efficiently perform database searches and modifications.
- Storage engine: The storage engine, or database engine, is the software component that processes the byte code and runs the intended SQL statement. It reads and stores the data in the database files on physical disk storage. Upon completion, the storage engine returns the result to the requesting application.

SQL commands are specific keywords or SQL statements that developers use to manipulate the data stored in a relational database. You can categorize SQL commands as follows:

- Data definition language: Data definition language (DDL) refers to SQL commands that design the database structure. Database engineers use DDL to create and modify database objects based on business requirements. For example, the database engineer uses the CREATE command to create database objects such as tables, views, and indexes.

- Data query language: Data query language (DQL) consists of instructions for retrieving data stored in relational databases. Software applications use the SELECT command to filter and return specific results from a SQL table.
- Data manipulation language: Data manipulation language (DML) statements write new information or modify existing records in a relational database. For example, an application uses the INSERT command to store a new record in the database.
- Data control language: Database administrators use data control language (DCL) to manage or authorize database access for other users. For example, they can use the GRANT command to permit certain applications to manipulate one or more tables.
- Transaction control language: The relational engine uses transaction control language (TCL) to automatically make database changes. For example, the database uses the ROLLBACK command to undo an erroneous transaction.

2.2.2 Creating a dataset

For INTEL-IRRIS data, we will program SQL commands modified, near our task. We will use parameters from weather climate, irrigation data from the system and crop standards. All those will be fixed with the different soil types properties. With appropriate tests and changes, we will then proceed to the creation, to the training of the "intel iris" platform. It will be a rubber system, in which parameters will be incorporated and various types of soils from all countries worldwide can be analyzed. A special part of our effort will be, for a start, the control of the correctness and proper functioning of the sensors. Since the platform records measurements from industrial sensors (calibrated ready) and the measurements of the INTEL-IRRIS sensors, the platform will also function as a control system. It will compare the simultaneous measurements of the two categories of sensors, calculate the difference and the error and record in separate extractable data the quality of the measurement after the comparison. If the error is within the limit, then "accept measurement" will be printed. If not, then it will be printed.

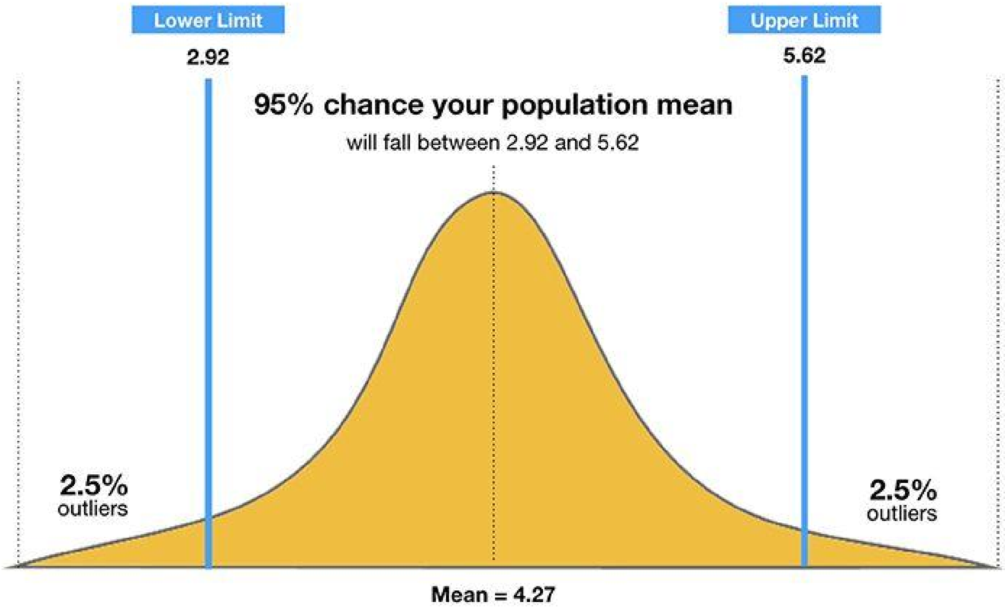
2.3 Data analysis

Statistical method analysis “Confidence Interval Estimates for Smaller Samples (95%)” for small samples methodology. With smaller samples ($n< 30$) the Central Limit Theorem does not apply, and another distribution called the t distribution must be used. The t distribution is like the standard normal distribution but takes a slightly different shape depending on the sample size. In a sense, one could think of the t distribution as a family of distributions for smaller samples. Instead of "Z" values, there are "t" values for confidence intervals which are larger for smaller samples, producing larger margins of error, because small samples are less precise. t values are listed by degrees of freedom (df). Just as with large samples, the t distribution assumes that the outcome of interest is approximately normally distributed. A table of t values is shown in the frame below.

Table - Z-Scores for Commonly Used Confidence Intervals

Desired Confidence Interval	Z Score
90%	1.645
95%	1.96
99%	2.576

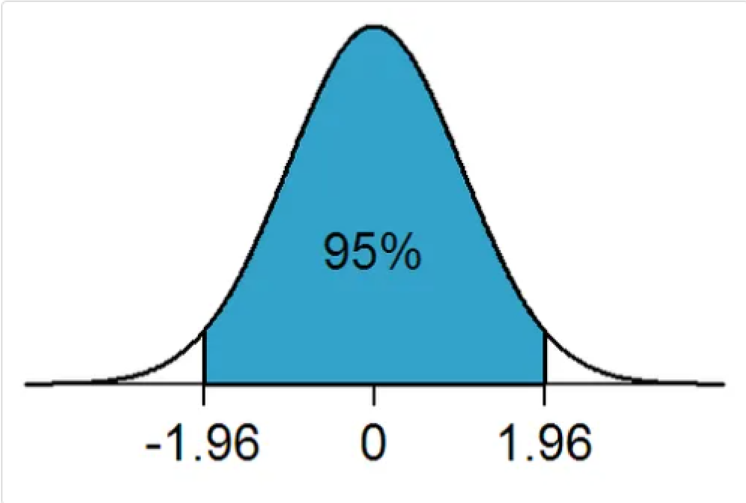
The confidence interval (CI) is a range of values that's likely to include a population value with a certain degree of confidence. It is often expressed as a % whereby a population mean lies between an upper and lower interval.



The 95% confidence interval is a range of values that you can be 95% confident contains the true mean of the population. Due to natural sampling variability, the sample mean (center of the CI) will vary from sample to sample.

The confidence is in the method, not in a particular CI. If we repeated the sampling method many times, approximately 95% of the intervals constructed would capture the true population mean.

Therefore, as the sample size increases, the range of interval values will narrow, meaning that you know that means with much more accuracy compared with a smaller sample. We can visualize this using a normal distribution (see the below graph).



For example, the probability of the population mean value being between -1.96 and +1.96 standard deviations (z-scores) from the sample mean is 95%.

Accordingly, there is a 5% chance that the population mean lies outside of the upper and lower confidence interval (as illustrated by the 2.5% of outliers on either side of the 1.96 z-scores).

With the application of the statistical analysis mentioned, we manage to validate the correct operation and correctness of the measurements continuously, which is the main task in "intel-iris". The construction and use of more economical measuring sensors for further irrigation decision-making.

References

1. Uthayakumar A.; Mohan M.J.; Khoo E.H; Jimeno J; Siyal M.Y.; Karim M.F. Machine Learning Models for Enhanced Estimation of Soil Moisture Using Wideband Radar Sensor. Sensors 2022, 22(15), 5810; https://doi.org/10.3390/s22155810
2. Batchu V.; Nearing G.; Gulshana V. A Machine Learning Data Fusion Model for Soil Moisture Retrieval https://arxiv.org/ftp/arxiv/papers/2206/2206.09649.pdf
3. Adab H.; Morbidelli R.; Saltalippi C.; Moradian M.; Ghalhari G.A.F. Machine Learning to Estimate Surface Soil Moisture from Remote Sensing Data

Acronyms List

Acronym	Explanation

Project Co-ordinator Contact

Pr. Congduc Pham

University of Pau

Avenue de l'Université

64000 PAU

FRANCE

Email: Congduc.Pham@univ-pau.fr

Acknowledgement

This document has been produced in the context of the PRIMA INTEL-IRRIS project. The INTEL-IRRIS project consortium would like to acknowledge that the research leading to these results has received funding from the European Union through the PRIMA program.