**UNIVERSIDADE FEDERAL DE PELOTAS**
**Centro de Desenvolvimento Tecnológico**
**Programa de Pós-Graduação em Computação**

Tese

# An API-based Framework for Clustering Meteorological Time Series for Agricultural Applications

**Marcos Antonio de Oliveira Junior**

Pelotas, 2023

**Marcos Antonio de Oliveira Junior**

# An API-based Framework for Clustering Meteorological Time Series for Agricultural Applications

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Advisor: Prof. Dr. Gerson Geraldo Homrich Cavalheiro

Coadvisores: Prof. Dr. Ricardo Matsumura Araujo
Prof. Dr. Clyde William Fraisse

Pelotas, 2023

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação da Publicação

O48a Oliveira Junior, Marcos Antonio de

An API-based Framework for Clustering Meteorological Time Series for Agricultural Applications [recurso eletrônico] / Marcos Antonio de Oliveira Junior ; Gerson Geraldo Homrich Cavalheiro, orientador ; Ricardo Matsumura Araujo, Clyde William Fraisse, coorientadores. — Pelotas, 2023.
140 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2023.

1. Clusterização. 2. Séries temporais. 3. Estações meteorológicas. 4. Aprendizado de máquina. I. Cavalheiro, Gerson Geraldo Homrich, orient. II. Araujo, Ricardo Matsumura, coorient. III. Fraisse, Clyde William, coorient. IV. Título.

CDD 005

Elaborada por Simone Godinho Maisonave CRB: 10/1733

**Marcos Antonio de Oliveira Junior**

**An API-based Framework for Clustering Meteorological Time Series for Agricultural Applications**

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

**Data da Defesa:** 12 de dezembro de 2023

**Banca Examinadora:**
Prof. Dr. Glauber Acunha Goncalves
Doutor em Ciências Geodésicas Universidade Federal do Paraná.

Prof. Dr. Paulo Roberto Ferreira Jr.
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Thiago Berticelli Ló
Doutor em Engenharia Agrícola pela Universidade Estadual do Oeste do Paraná.

Aos meus pais, Denise e Marcos, meus maiores exemplos de dedicação, resiliência e perseverança.

# AGRADECIMENTOS

Agradeço, primeiramente, à minha família, por todo o suporte durante todo o ciclo do curso de doutorado. À minha esposa, Monalisa, agradeço pelo amor, cuidado, paciência e compreensão ao longo desses quatro anos. Obrigado por sempre ter uma palavra de incentivo em meio ao cansaço e de tranquilidade nos momentos difíceis. Aos meus pais, Denise e Marcos, deixo também meu agradecimento por sempre acreditarem em mim, muito antes de eu mesmo acreditar. Eu vi vocês percorrerem esse caminho, com muitas dificuldades, mas sem se deixar abater, com uma dedicação incansável, que me inspira diariamente a buscar meus objetivos. Vocês me ensinaram a estudar e a ter responsabilidade com meus deveres. Se hoje concluo mais essa etapa acadêmica, vocês são os principais responsáveis por me colocarem nesse caminho.

Aos professores orientadores, muito obrigado pela atenção dedicada aos meus questionamentos, pelas trocas valiosas e orientação durante essa caminhada. Vocês possuem trajetórias acadêmicas muito inspiradoras e me sinto privilegiado por ter aprendido um pouco com cada um. Vocês me ensinaram a ser um pesquisador melhor, um profissional melhor e, principalmente, uma pessoa melhor.

- Ao Prof. Gerson, agradeço pela forma respeitosa e cordial que sempre me tratou, desde o primeiro contato, lá no início do doutorado, quando eu não tinha nem ideia de pesquisa, até os mais recentes, nas semanas de interações intensas que antecederam a entrega desta tese. Obrigado por acreditar no meu potencial e acolher minhas ideias, não medindo esforços para me auxiliar a buscar meus objetivos, especialmente no meu sonho de realizar um doutorado sanduíche. Foi uma experiência muito enriquecedora ter você como orientador, me dando liberdade para as tomadas de decisão, e ao mesmo tempo, me puxando de volta para o rumo quando necessário.

- Ao Prof. Ricardo, agradeço também pela sua coorientação, que foi fundamental para o desenvolvimento deste trabalho. Desde a disciplina de Aprendizado de Máquina no segundo semestre, às contribuições no projeto do doutorado sanduíche, até as conversas mais recentes, sempre aprendi muito com seus feedbacks e insights, e esse conhecimento serviu como a base técnica de machine learning para o desenvolvimento deste trabalho.

- Ao Prof. Clyde, expresso também uma imensa gratidão por me receber em seu grupo de pesquisa para o período de doutorado sanduíche, pela acolhida calorosa e pela coorientação durante e após esse intercâmbio. Para além dos valiosos aprendizados sobre agrometeorologia, as experiências culturais proporcionadas pelo Agro-Climate foram marcantes, não somente enquanto pesquisador mas também enquanto ser humano. O doutorado sanduíche foi um momento de muito crescimento e sua coorientação fundamental para isso.

À Universidade Federal de Pelotas e aos professores do PPG Computação, agradeço imensamente pelo suporte ao longo do doutorado, em todas as etapas acadêmicas e administrativas percorridas. Agradeço também às agências brasileiras de fomento à pesquisa, em especial à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo recurso financeiro disponibilizado para realização do doutorado sanduíche no exterior. Da mesma forma, agradeço à University of Florida pela abertura e receptividade para uma cooperação acadêmica, que tornou possível a internacionalização.

Aos colegas de doutorado, com os quais pude compartilhar muitos momentos de aulas, trabalhos e trocas de experiência, também deixo meu agradecimento. Em especial, agradeço aos colegas Anderson e Fernando, com os quais tive a oportunidade de trabalhar de forma mais próxima, assim como ao Gregory, por todo o aprendizado que tive trabalhando com vocês. Da mesma forma, aos colegas de AgroClimate, Maurício, Vinicius e Leandro, pela hospitalidade com que me receberam, por toda a ajuda na ambientação em um novo país e pelo convívio e risadas compartilhadas no laboratório durante os meses de trabalho em conjunto. Sou grato por minha trajetória acadêmica ter cruzado com a de todos vocês, pesquisadores muito competentes, abertos a ensinar e aprender, e que se tornaram grandes amigos.

Aos meus colegas de trabalho, servidores do IFFar, em especial aos da Diretoria de Tecnologia da Informação, meus mais sinceros agradecimentos pela parceria durante o doutorado. Para além do aprendizado constante no dia a dia de trabalho e das risadas nos momentos de descontração, não foram poucas as vezes em que alguém da equipe assumiu demandas extras devido à minha ausência, principalmente durante o meu afastamento para o doutorado sanduíche. Sem a compreensão e a colaboração de todos vocês tenho certeza que essa etapa de internacionalização, que foi um capítulo especial do doutorado, não teria se concretizado.

Por fim, agradeço de coração a todos que de uma forma ou outra contribuiram para essa caminhada. Aos meus amigos de fé, dessas e de outras idas, agradeço pelo carinho, compreensão e paciência nos momentos de ausências nos churrascos, jogos de futebol, e outras confraternizações. Esse ciclo foi muito trabalhoso e foi muito especial sempre sentir o apoio dos amigos nas diversas etapas do curso. Muito obrigado pelas caronas e viagens compartilhadas, pelas estadias quando estive fora de casa, pelas refeições preparadas para que eu pudesse dedicar mais tempo ao meu trabalho e por tantas outras demonstrações de afeto. O doutorado foi um caminho desafiador, mas percorrê-lo se tornou viável graças ao apoio de vocês.

*"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."*
— Eliezer Yudkowsky

# RESUMO

OLIVEIRA JUNIOR, Marcos Antonio de. **An API-based Framework for Clustering Meteorological Time Series for Agricultural Applications**. Orientador: Gerson Geraldo Homrich Cavalheiro. 2023. 140 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2023.

A variabilidade climática possui grande importância na definição dos resultados agrícolas, influenciando o crescimento das culturas, o rendimento e as estratégias de gestão de recursos. A capacidade de identificar padrões significativos em conjuntos de dados meteorológicos complexos é de grande valor para otimizar as práticas agrícolas e garantir a segurança alimentar mundial. O zoneamento climático, por exemplo, é um conhecimento importante que contribui para a precisão dos sistemas de recomendação agrícola. Neste contexto, este trabalho propõe um framework baseado em API para clusterizar dados meteorológicos em formato de série temporal para aplicações agrícolas. O principal objectivo do framework é viabilizar a identificação padrões climáticos em variáveis recolhidas por estações meteorológicas, a fim de subsidiar a tomada de decisões agrícolas com um zoneamento climático, de forma confiável e automatizada. Após uma extensa e descritiva revisão sistemática da literatura existente, técnicas estatísticas, algoritmos de clusterização e métricas de similaridade foram reunidos, extendidos e implementados no formato de API. O conjunto de métodos aplicados foi ordenado em uma sequência lógica e eficiente, de forma a guiar as tarefas de extração de dados, pré-processamento, engenharia de features, clusterização e validação. A aplicabilidade do framework foi validada por meio de dois estudos de caso utilizando dados meteorológicos, da FAWN/FL e do SIMAGRO-RS, seguidos de discussão dos resultados. Os resultados obtidos indicaram a viabilidade da utilização do framework, suas contribuições e limitações, destacando o seu potencial para melhoria da entrada de sistemas de decisão agrícola. Por fim, são destacados insights obtidos a partir da clusterização e são propostas formas de utilização da API, interligada com sistemas agrícolas.

Palavras-chave: Clusterização. Séries Temporais. Estações Meteorológicas. Aprendizado de Máquina.

# ABSTRACT

OLIVEIRA JUNIOR, Marcos Antonio de. **An API-based Framework for Clustering Meteorological Time Series for Agricultural Applications**. Advisor: Gerson Geraldo Homrich Cavalheiro. 2023. 140 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2023.

Climate variability plays a key role in shaping agricultural outcomes, influencing crop growth, yield and resource management strategies. The ability to discern meaningful patterns in complex meteorological data sets is invaluable for optimizing agricultural practices and ensuring global food security. Climate zoning, for example, is important knowledge that contributes to the accuracy of agricultural recommendation systems. In this context, this work proposes an API-based framework to group meteorological data in time series format for agricultural applications. The main objective of the framework is to enable the identification of climate patterns in variables collected by meteorological stations, in order to provide climate zoning for agricultural decision-making, in a reliable and automated way. After an extensive and descriptive systematic review of existing literature, statistical techniques, clustering algorithms and similarity metrics were brought together, extended and implemented in API format. The set of applied methods was ordered in a logical and efficient sequence, guiding the tasks of data extraction, pre-processing, feature engineering, clustering and validation. The applicability of the framework was validated through two case studies using meteorological data, from FAWN/FL and SIMAGRO-RS, followed by a discussion of the results. The results obtained indicated the feasibility of using the framework, its contributions and limitations, highlighting its potential for improving input into agricultural decision systems. Finally, insights obtained from clustering are highlighted and ways of using the API are proposed in an interconnected way with agricultural systems.

Keywords: Clustering. Time Series. Meteorological Stations. Machine Learning.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| API | Application Programming Interface |
| BAS | Blueberry Advisory System |
| DAS | Disease Alert System |
| DOY | Day of the Year |
| DTW | Dynamic Time Warping |
| EM | Expectation-Maximization |
| ESS | Sum of Squares Error |
| FAWN | Florida Automated Weather Network |
| FCM | Fuzzy C-Means |
| GMM | Gaussian Mixture Model |
| HAC | Hierarchical Agglomerative Clustering |
| IoT | Internet of Things |
| IQR | Interquartile Range |
| ML | Machine Learning |
| MTS | Meteorological Time Series |
| PCA | Principal Component Analysis |
| QA | Quality Assessment |
| RHyp | Research Hypoteshis |
| RMSE | Root Mean Squared Error |
| RQ | Research Question |
| SIMAGRO-RS | Sistema de Monitoramento e Alertas Agroclimáticos do RS |
| SLR | Systematic Literature Review |
| SOM | Self-Organizing Maps |
| SMS | Short Message Service |
| StAS | Strawberry Advisory System |
| WCSS | Within-Cluster Sum of Squares |

# CONTENTS

# 1 INTRODUCTION

Agriculture has a direct impact on global food security and economic stability, with a multitude of climate-dependent activities, including planting, harvesting, irrigation, and disease control, that are closely intertwined with seasonal and climatic variations. The interaction of climatic conditions and crop-related activities has also been changing with constant and accentuated climate changes around the world. In addition to this, the evolution of technology in recent years, especially assisted by computer systems, led to the development of several applications for analyzing meteorological data, especially focused on agricultural activities, through a dynamic approach (ZHAO; LIU; HUANG, 2023). Thus, monitoring weather variables has emerged as a powerful tool for discovering patterns and understanding the impact of meteorological events on agricultural activities.

In recent years, remarkable developments in technology have been experienced that have profoundly transformed the hardware and software in the agriculture domain (KARUNATHILAKE et al., 2023). When it comes to hardware, the proliferation of Internet of Things (IoT) sensors has ushered in an era of unprecedented data collection capabilities. These sensors, which range from tiny, low-power devices to sophisticated environmental monitoring equipment, has not only made it possible to capture large amounts of data, but has also driven innovations in sensor miniaturization, energy efficiency, and wireless connectivity, further expanding the potential applications of sensors devices in agriculture (PRAKASH et al., 2023). On the software side, Machine Learning (ML) techniques have also undergone a remarkable revolution, allowing the extraction of valuable information from the huge data sets generated by sensors. ML algorithms have improved their performance in tasks such as prediction, anomaly detection and classification. From an agricultural perspective, these software advances have enabled stakeholders to harness the wealth of data, making it possible to automate processes, optimize resource allocation, and improve the overall efficiency and effectiveness of agricultural decision support systems (JAVAID et al., 2023). The synergy between sensors and machine learning techniques is reshaping the way data is analyzed, paving the way for a more interconnected, data-driven, and efficient future.

With regard to meteorological data, sensor devices are now widespread in agricultural regions, especially in the form of weather stations, capable of collecting a wide range of variables such as temperature, rainfall, relative humidity, solar radiation, wind speed and direction. These variables are generally expressed in time series format, a specific and challenging data structure (MUDELSEE, 2014). Time series data, collected through a variety of meteorological instruments, provides an extensive and invaluable resource for understanding the dynamic nature of these environmental variables. Thus, a technique that has wide applicability in this context is the cluster analysis, capable of extracting relevant information from a set of data and providing outputs such as dynamic climate zoning of agricultural areas, aware of and monitoring climate change.

The intersection of time series data clustering and climate monitoring in agriculture offers a multidisciplinary approach that has the potential to improve crop yields (REYES et al., 2023), resource management (AKAY, 2021), and resilience to climate change (CLIFTON; LUNDQUIST, 2012). On a regional scale, for example, the occurrence of extreme weather events was recently observed in the state of Rio Grande do Sul, especially intense rains in the center of the state at the beginning of September 2023, which were not common in recent years and which certainly will impact agricultural production in the next harvest. In view of these changes, historical and traditional climate patterns, defined in past decades, are no longer sufficient to assist producers in making decisions for the coming years (LEAL FILHO et al., 2023). Climate zoning, especially for agriculture, now presents dynamic properties, requiring constant monitoring and tools that enable analysis and decisions, almost in real time. Furthermore, this knowledge, in addition to helping in future agricultural production cycles, can assist in political decisions, the allocation of financial resources, and the promotion of sustainable agricultural practices (TORRESAN et al., 2016).

Therefore, in an overview, this research explores the challenges and possibilities of applying clustering techniques to Meteorological Time Series (MTS) used in agricultural applications. After a literature review on state-of-the-art knowledge on this topic, a framework is proposed, supported by a software package in the form of an API, in order to guide the task of agricultural climate zoning based on MTS, going through several steps, starting from data extraction, data preprocessing, feature extraction, data clustering and evaluation clustering. With this, the frontier of knowledge is advanced, producing a technological artifact capable of assisting stakeholders in the agricultural context by improving decision support systems, as well as compiling well-known and context-aware practices for future researchers on this topic.

The framework was validated through two case studies, with meteorological datasets from different regions (one from Rio Grande do Sul, the southernmost state of Brazil, and another from Florida, southeast of the USA) and with different agricultural purposes. The case studies carried out not only validated the framework, but also contributed in several aspects to its development and implementation. As a result, climatic zonings were obtained for both regions, based on the analysis of climatic variables and specific periods of agricultural production, in order to base agricultural management decisions.

## 1.1 Research Context

MTS differ from other types of time series in several ways due to the unique nature of meteorological data and the specific challenges they present. MTS generally have high temporal resolution, with measurements recorded at frequent intervals, such as minutes or hours. This high frequency is necessary to capture rapid atmospheric changes and variations, such as temperature fluctuations, changes in wind speed, or short-lived precipitation events. Also, MTS frequently exhibit clear cyclic patterns, which sets them apart from other types of data that may lack such pronounced periodicity.

Missing data and data quality are also factors of concern in this type of data. MTS are susceptible to data loss and outliers due to instrument malfunctions, sensor failures, or data transmission problems (CERLINI; SILVESTRI; SARACENI, 2020). Furthermore, while other types of time series may not require this specialized domain knowledge, analysis of MTS data often requires domain-specific knowledge and experience for interpretation and analysis (DAS; GHOSH, 2018). Understanding meteorological variations as well as agricultural activities is crucial for accurate modeling and forecasting.

Time series data in meteorology represents the sequential measurements of various meteorological parameters such as temperature, precipitation, wind speed and atmospheric pressure, recorded at regular intervals over time. Analysis of these datasets provides valuable information for climate zoning in agriculture, including weather forecasting, climate modeling and assessment of long-term climate trends (ANDERSON; BAYER; EDWARDS, 2020). Time series analysis techniques such as statistical methods are often employed to identify patterns and trends in meteorological data and using ML techniques it is possible to complement and deepen the analyses.

The most common statistical MTS analyzes encompass a variety of methods employed to extract meaningful information from data. Some of the most widely used analyzes include: *Descriptive Statistics* (KOMALASARI; PAWITAN; FAQIH, 2017), such as mean, median, variance, and standard deviation, providing a basic summary of data, central tendencies, variability, and distribution of variables; *Trend Analysis* (MUDELSEE, 2019), evaluates long-term changes in meteorological parameters; *Seasonal Decomposition* (QIAN et al., 2019), separating time series data into components like trend, seasonal, and residual variations; *Principal Component Analysis (PCA)* (TADIĆ; BONACCI; BRLEKOVIĆ, 2019), used to reduce the dimensionality of meteorological data by identifying dominant patterns; *Anomaly Detection* (BLÁZQUEZ-GARCÍA et al., 2021), to identify unusual or extreme climatic events, such as heatwaves, cold spells, or intense precipitation; and, *Data Visualization* (RUDENKO et al., 2022), including time series plots, heatmaps, and contour plots, for interpreting MTS data and conveying information to a broad audience. Generally, these techniques are applied in a complementary way, in order to increase the range of possible inferences. In addition to statistics, a very interesting alternative to analyzing MTS is the use of machine learning techniques, such as clustering (GOVENDER; SIVAKUMAR, 2020), to complement statistical analyses.

## 1.2 Motivation, Research Questions and Hypothesis

With the increasing availability of high-resolution weather data, the motivation for this research arises from the pressing need to develop advanced analytical techniques that can reveal hidden patterns and structures within MTS. Such insights can greatly enhance climate zoning for agricultural contexts and improve decision-making. This work is built around the problem of clustering climate time series for agricultural applications, which is intensely challenging due to its complexity, which involves steps such as preprocessing, the clustering itself and the evaluation of results, even more so from a specific perspective, such as agricultural. Existing solutions are characterized by mosaics and combinations of techniques, without a central guiding line, which allows, for example, reliable comparisons between different solutions.

Beyond the immediate practical applications, this thesis is deeply rooted in the global challenges posed by climate change. As our planet experiences in global scale unprecedented shifts in weather patterns and rising temperatures, and in local scale an increased frequency of extreme events, such as rainfall, understanding the intricacies of meteorological data becomes vital for agriculture, to ensure food production. By uncovering hidden patterns through clustering techniques, we can gain deeper insights into the drivers of climate change and its potential consequences. These clustering outputs extract a new layer of information, which can be used as input in agricultural decision support systems, to improve crop performance and sustainable use of natural resources.

Based on this motivation, problem, and context introduced previously, the central Research Question (cRQ) of this thesis focuses on:

**cRQ - How to effectively cluster meteorological time series data in order to help agricultural decision support systems?**

Furthermore, to clarify the discussion, this research question can be divided into minor specific questions (mRQ), as follows:

- ***mRQ1:*** *What are the main solutions for clustering meteorological time series in the agricultural context?*
- ***mRQ2:*** *How are these algorithms evaluated and compared?*
- ***mRQ3:*** *What factors motivate the clustering of time series in agriculture applications?*

After conducting a literature review system presented in the following chapter, it was observed that existing solutions follow very specific methodologies, with many manual procedures and analyses, requiring a lot of domain and statistical knowledge, combining different techniques according to convenience, without procedure standardization. This leads us to raise a central Research Hypothesis (RHyp):

**RHyp:** *The existence of a framework, supported by a software implementation to automate processes, can assist stakeholders and make the use of meteorological time series clustering accessible to a greater number of agricultural applications.*

## 1.3 Goals and Thesis Contributions

The primary goal of this research is to advance the state of the art in meteorological time series clustering through the development of a framework capable of guiding researchers and stakeholders in this task. This research aims to provide computer scientists, agrometeorologists, and other climate stakeholders with powerful tools to better understand, categorize, and interpret weather data patterns. This research not only contributes to the scientific community's knowledge base but also offers actionable information that can guide policymakers and stakeholders in developing effective strategies for climate adaptation.

Thus, the **General Objective** of this thesis is to introduce an API-based framework for clustering time series in agricultural applications, taking into account, mainly, the particularities of this type of data, the behavior of variables, clustering algorithms and agricultural practices.

To achieve this goal, the workflow is divided into some **Specific Objectives**:

- Investigate and map existing solutions for clustering meteorological time series, as well performance metrics to evaluate them;
- Establish a working methodology, in a framework format, with well-defined steps, that allows replicability and comparison of different solutions for clustering methodological time series;

- Demonstrate the applicability and usefulness of this framework with data from real agricultural applications;
- Explore potential applications of meteorological time series clustering in agricultural support decision systems.

## 1.4 Results Summary

After the design and implementation phases, the framework was tested and validated through two different case studies. The first study used meteorological data from SIMAGRO-RS, from Rio Grande do Sul, in southern Brazil, and the second used meteorological data from FAWN, from Florida, United States of America. First, key variables such as Temperature, Relative Humidity, Precipitation and Solar Radiation were extracted from meteorological stations spread across different locations in each state, during different time windows. Then, the data was subjected to pre-processing, including quality checks and filtering adapted to different agricultural seasons. The third stage involved resource engineering, where data was aggregated to obtain daily averages and standardized values. Finally, clustering techniques were applied and the results were evaluated using similarity metrics.

As the main result of this process, climate zonings were generated for each state, grouping meteorological stations into clusters. This strategic clustering aimed to provide additional valuable meteorological information to support agricultural decision systems. When examining the results and cross-referencing them with traditional climate reference documents, it was observed that the proposed clustering results generally align with the spatial characteristics of different regions of the state. The main contribution of the framework lies in redefining the boundaries of climatic regions. Analysis of region-specific climate data, along with focusing on specific seasons and agricultural activities, produces specialized knowledge compared to the conventional and historical division of climate regions for both states.

## 1.5 Thesis Outline

This thesis is organized as follows:

- **Chapter 2: Background** presents a review of the background necessary for a better understanding of the thesis, such as basic concepts about machine learning and clustering, evaluation metrics, data preprocessing and feature extraction.
- **Chapter 3: Related Works** presents a systematic literature review, in order to map existing solutions for the clustering of meteorological time series, especially analyzing the motivations for using clustering in an agricultural context. Based on

a review protocol, after the initial search, a set of papers was selected for a detailed review, where the algorithms used, the evaluation metrics and the product resulting from the clustering were discussed, in order to extract a snapshot of the state current research on this topic and identify research directions.

- **Chapter 4: Proposed Methodology** presents the methodology in the form of a framework for clustering time series in agricultural applications, detailing its operation flow and the processes involved in each step.

- **Chapter 5: Framework Implementation** presents the implementation details of the proposed framework, such as development environment, language, dependencies and implementation versions, as well as how to use it.

- **Chapter 6: Experiments, Results, and Discussion** presents the experiments carried out, exemplifying the use of the framework, in order to demonstrate its applicability in data sets from real agricultural applications. The results obtained are also discussed, analyzing feasibility, limitations and contributions.

- **Chapter 7: Conclusion** presents the conclusion of this research, summarizing the main aspects of the work, relating the results to the objectives and goals, and highlighting possibilities for future work.

# 2 BACKGROUND

This chapter provides the necessary background for a better understanding of the thesis, presenting basic concepts about the theoretical framework. Initially, the clustering task is introduced, within the scope of machine learning, as well as metrics for computing the similarity of elements and other processes involved in this task, such as data preprocessing steps.

## 2.1 Meteorological Time Series

Time series data refers to a sequence of observations or measurements recorded over equally spaced time intervals. This data format captures the evolution of a variable or phenomenon over time, making it a powerful tool for understanding trends, patterns, and behaviors in various fields. Time series data is prevalent across disciplines, with diverse applications, and plays a fundamental role in forecasting future values, detecting anomalies, and making informed decisions in response to historical trends (HAMILTON, 2020). However, while time series data offers invaluable insights, it presents several challenges regarding inherent temporal dependencies, noise, missing values, seasonality, and trends, that require careful preprocessing and modeling. The volume of data, in cases with high-frequency measurements, also poses challenges for analysis and visualization. In the context of this research, we are interested in a specific and important type of time series, which are MTS.

Meteorological Time Series (MTS) data is a specialized subset of time series data, which, in general, involves the sequential recording of meteorological observations or measurements over time (GANGULY; STEINHAEUSER, 2008). It provides valuable information about how a weather variable changes over time, and it is particularly important in meteorology for understanding and predicting climate conditions. This type of time series data focuses on capturing weather-related parameters, such as temperature, humidity, wind speed, atmospheric pressure, and precipitation, over time, from ground-based weather stations, satellites, radar systems, and other sources to gain insights into the Earth's atmospheric conditions and to inform a wide range of applica-

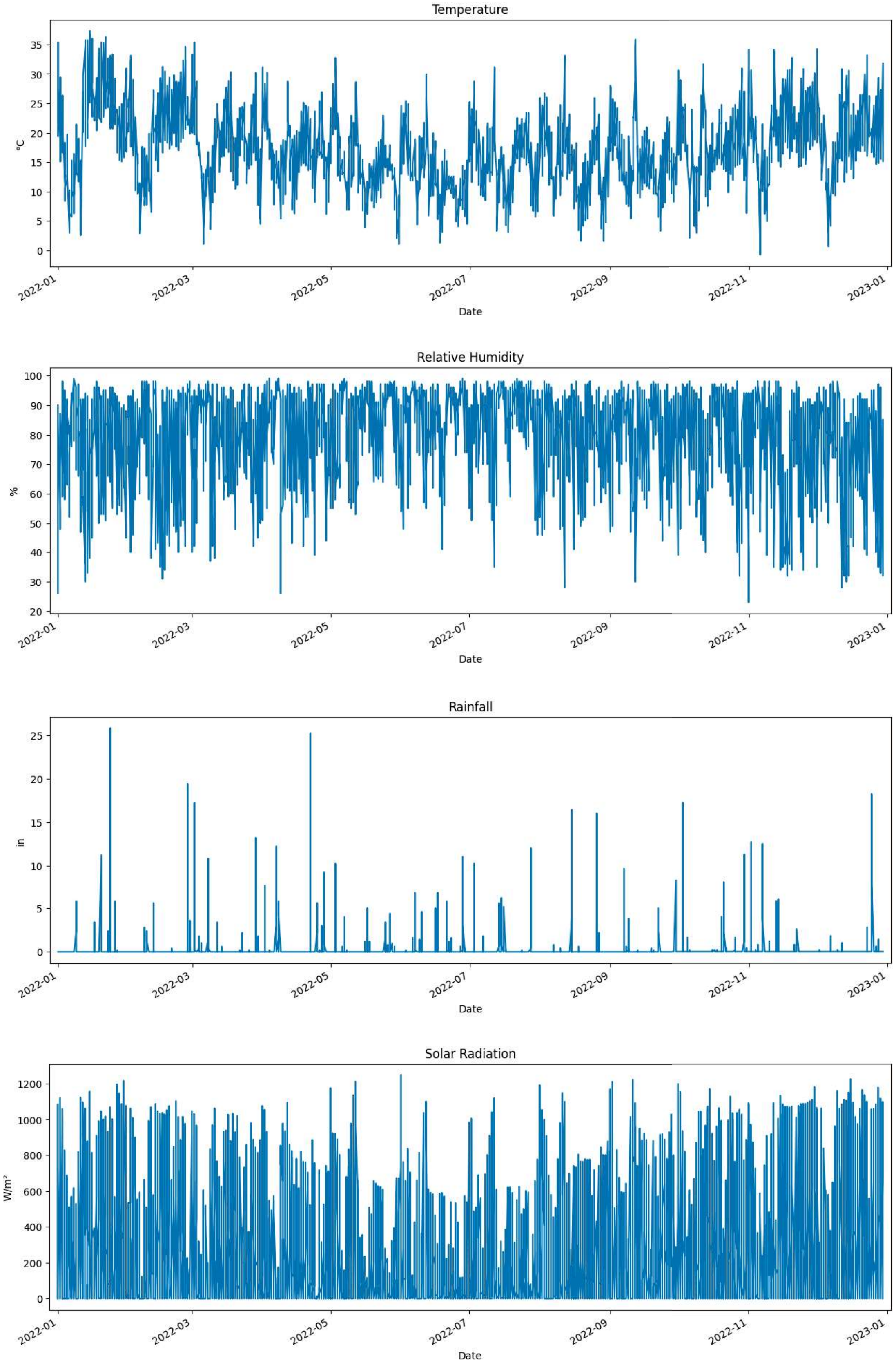


Figure 1 – Examples of meteorological time series (MTS).

tions, including weather forecasting, climate research, and environmental monitoring. Figure 1 shows examples of the MTS used in this research.

MTS data is characterized by high temporal resolution, often recorded at hourly or even finer time intervals. These time series are inherently noisy, as weather conditions are subject to rapid fluctuations and can exhibit both short-term variability and long-term trends (ESLING; AGON, 2012). Analyzing MTS requires specialized techniques that account for seasonality, cycles, and the presence of extreme events like storms or heatwaves. These time series can extend over several decades, and modern meteorological data often comes from a variety of sources with different formats and quality levels. MTS analysis contributes to a deeper understanding of atmospheric dynamics, which is essential for addressing climate-related issues, optimizing resource management, and mitigating the impact of extreme weather events on society.

## 2.2 Machine Learning

Machine Learning (ML) is a dynamic field within artificial intelligence that has revolutionized the way to analyze and understand complex datasets. It involves the development of algorithms and models that enable computers to learn from data, make predictions, and discover patterns without being explicitly programmed (RASCHKA, 2015). ML, compared to traditional programming, has changed the way computational models are designed. Traditional programming and ML represent two distinct paradigms in the world of computer science. In traditional programming, explicit instructions are crafted to solve a specific problem, and the computer executes these commands precisely. This approach relies on a programmer's ability to anticipate and account for all possible scenarios. On the other hand, ML operates on the principle of enabling computers to learn from data and improve their performance over time without being explicitly programmed. Instead of following predefined rules, ML systems analyze patterns and make decisions based on the information they've been trained on. Figure 2 illustrates this difference between these paradigms. The key distinction lies in the flexibility and adaptability of ML, where algorithms can generalize and make predictions on new, unseen data, making it particularly powerful for tasks with complex patterns or large datasets.

ML can be broadly categorized into three main types: Supervised, Unsupervised, and Reinforcement learning, each with distinct objectives and applications. Supervised learning is a paradigm of ML where the algorithm is trained on a labeled dataset, which means that the input data is paired with corresponding output labels. Unsupervised learning is other ML paradigm that deals with unlabelled data, focusing on discovering inherent structures or patterns within the data. Reinforcement learning is another ML training method based on rewarding desired behaviors and punishing undesired ones.

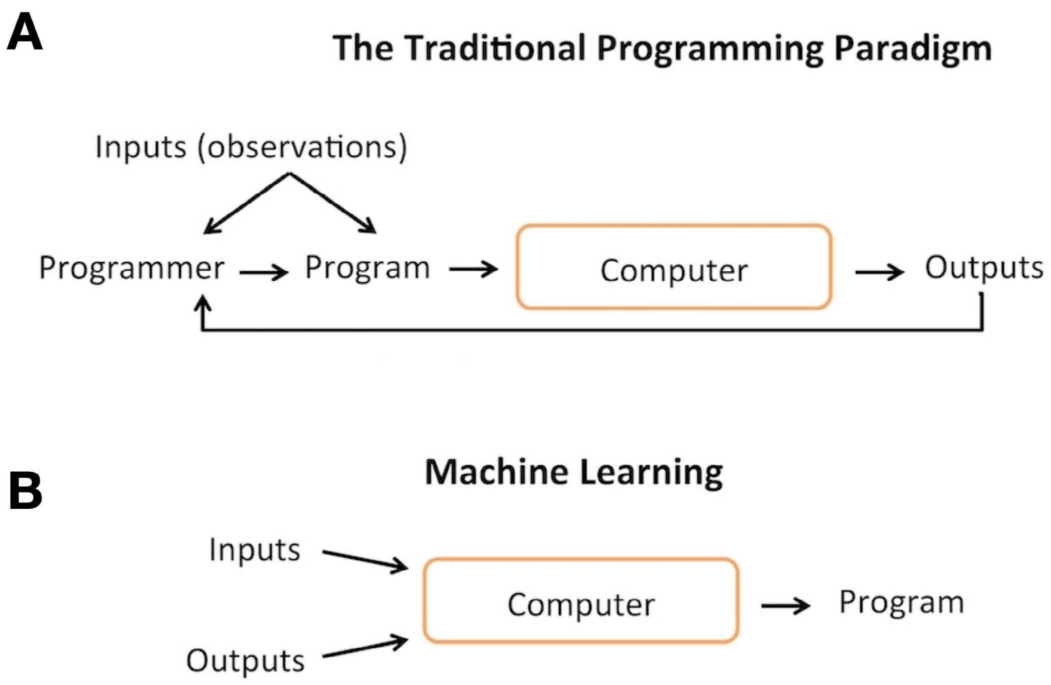


Figure 2 – Traditional Programming (A) vs. Machine Learning (B). (RASCHKA et al., 2022)

In the context of clustering, supervised learning is not typically used, as clustering tasks involve grouping data without prior knowledge of distinct labels or categories (RASCHKA et al., 2022). Instead, unsupervised learning methods are more suitable for clustering applications.

### 2.2.1 Clustering Definition

Clustering is a data analysis technique in ML and data mining that involves the process of grouping a set of data points or objects into groups, where each cluster consists of data points that are more similar to each other than to those in other clusters (JAIN; MURTY; FLYNN, 1999). The primary objective of clustering is to discover underlying patterns, structures, or natural groupings within a dataset without any prior knowledge of the groups. The process of clustering can be thought of as sorting data points into clusters based on their similarities or dissimilarities, where the goal is to minimize the intra-cluster differences and maximize the inter-cluster differences. In other words, data points within a cluster are more homogeneous or similar, while data points in different clusters are more dissimilar. Clustering has a wide range of applications, specially in the context of this research, where it can help identify weather patterns and trends (TIAN et al., 2014). Figure 3 demonstrates three of the most common approaches to the clustering task: (a) Partitioning, (b) Hierarchical and (c) Model-based. These and other methods are detailed below.

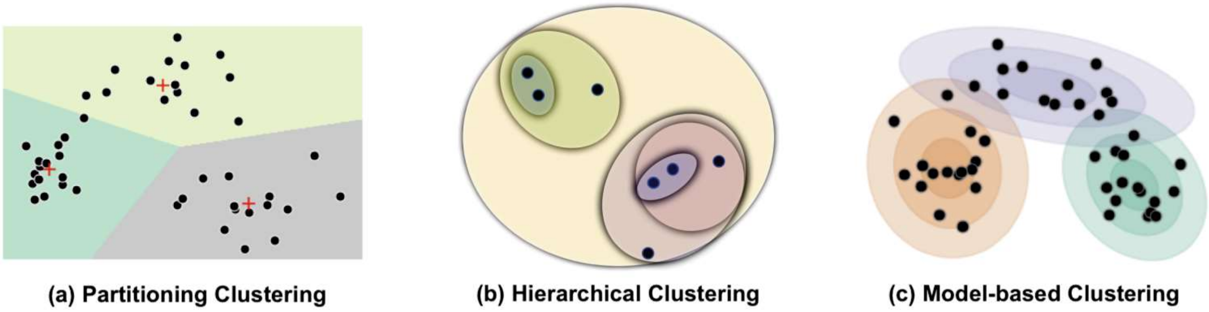

(a) Partitioning Clustering
(b) Hierarchical Clustering
(c) Model-based Clustering

Figure 3 – Popular clustering approaches. (KHOSLA et al., 2019)

### 2.2.2 Clustering Algorithms

Clustering algorithms are used to perform clustering tasks. These algorithms use various similarity measures to determine how data points should be grouped together. Clustering algorithms can be categorized into several main categories (SAXENA et al., 2017) based on their underlying principles and approaches:

- **Partitioning Algorithms:** Partitioning algorithms, such as k-means, are widely used for dividing data into non-overlapping clusters. The goal is to assign each data point to a single cluster to minimize the variance within clusters. These algorithms are computationally efficient and work well for data with well-defined spherical or convex-shaped clusters. However, they may struggle with non-convex clusters, varying cluster sizes, and are sensitive to the initial placement of cluster centroids, requiring multiple runs with different initializations to find the optimal solution.

- **Hierarchical Algorithms:** Hierarchical clustering methods create a hierarchical structure of clusters, often represented as a dendrogram. Agglomerative clustering begins with each data point as its own cluster and merges the closest clusters iteratively. Divisive clustering, on the other hand, starts with all data points in a single cluster and recursively divides them. These algorithms provide a hierarchy of clusters, allowing users to explore the data at different levels of granularity. They are flexible and can handle various cluster shapes and sizes but can be computationally demanding for large datasets.

- **Density-Based Algorithms:** Density-based clustering algorithms, like DBSCAN, focus on identifying clusters as regions of high-density data points separated by areas of lower density. These methods are effective in identifying clusters of arbitrary shapes, handling noise, and accommodating varying cluster sizes. They do not require the user to specify the number of clusters in advance, making them suitable for data where cluster count is not known.

- **Model-Based Algorithms:** Model-based clustering algorithms assume that data points are generated from a statistical model, often a Gaussian Mixture Model (GMM). They use the Expectation-Maximization (EM) algorithm to fit these models to the data. Model-based clustering is effective for capturing the underlying data distribution and works well with multivariate data but may require specifying the number of clusters in advance.

- **Fuzzy Clustering Algorithms:** Fuzzy clustering allows data points to belong to multiple clusters with varying degrees of membership. Fuzzy C-Means (FCM) is a popular example, where each data point is assigned membership values for each cluster. These algorithms are useful when data points exhibit ambiguity in their cluster assignments or when they belong to multiple clusters to varying degrees.
- **Spectral Clustering:** Spectral clustering techniques transform the data into a low-dimensional space and then apply standard clustering algorithms. They are effective at capturing complex structures in the data, such as non-convex clusters and intricate relationships between data points. Spectral clustering is particularly suitable for data that may not conform to traditional distance-based clustering approaches.
- **Self-Organizing Maps (SOM):** Self-Organizing Maps use neural network-based techniques to map high-dimensional data onto a lower-dimensional grid, preserving topological relationships between data points. They are useful for visualizing complex datasets and discovering clusters in high-dimensional spaces, providing an alternative approach to traditional clustering algorithms.

Each category of clustering algorithms has its own strengths and weaknesses, for this reason it is important to choose the clustering method that aligns with the characteristics of the time series data and the specific goals of the analysis. MTS data, which often exhibit cyclical behaviors and may have missing or null values, pose specific challenges for clustering. The choice of clustering algorithm should consider these characteristics.

### 2.2.3 Cluster Similarity Metrics

Similarity metrics are fundamental for clustering evaluation and validation. These metrics can quantify the likeness or resemblance between data objects and are indispensable in clustering algorithms (IRANI; PISE; PHATAK, 2016). They are employed to assess the degree of similarity or dissimilarity between data points, allowing clustering validation. These metrics fall into two primary categories: internal and external similarity metrics.

- **Internal Similarity:** these metrics are used when you do not have ground truth information about the true cluster assignments. They assess the quality of clustering solely based on the characteristics of the data and the clustering results. They help you understand the structure of the clusters within the data without comparing them to any external reference. The most commonly used, for general purposes, are Silhouette Score, Davies-Bouldin Index, Dunn Index and Calinski-Harabasz Index, also known as Variance Ratio Criterion.

- **External Similarity:** these metrics are employed when you have ground truth or reference information about the true cluster assignments. They evaluate how well the clustering results align with this known or expected grouping of data points. They assess the agreement between the predicted clusters and the ground truth clusters. The most commonly used, for general purposes, are Adjusted Rand Index, Normalized Mutual Information, Fowlkes-Mallows Index, Jaccard Index and Adjusted Mutual Information.

There is no one-size-fits-all similarity metric, and the best metric depends on the unique characteristics of your data and the objectives of your clustering task. Choosing the best similarity metric for your specific data and clustering task involves careful consideration of several factors, such as the characteristics of the dataset, clustering objectives, data types, interpretability and domain knowledge. Clustering is often an iterative process in which choices are often refined, evaluated and remade as necessary to gain more insights into the clustering results.

## 2.3 Meteorological Time Series Clustering

The use of ML in MTS analysis has been gaining prominence due to its ability to increase the accuracy, robustness and versatility of meteorological predictions and understanding. ML models are capable of identifying intricate and non-linear relationships within data, which may escape traditional statistical methods, allowing for more accurate information of meteorological parameters. Figure 4 shows an example of clustering in a random set of time series. The challenge is to group those that are most similar into the same cluster, using a similarity metric. Furthermore, the adaptability of ML models makes them suitable for dealing with the diverse and evolving nature of meteorological data. One of the main techniques used when analyzing climate time series aimed to agriculture is clustering, the main focus of this research. Clustering mainly helps in identifying climatic zones in agricultural areas, enabling the specialization of agricultural management tasks, in line with precision agriculture.

Clustering techniques applied to MTS data in agriculture offers a compelling avenue for enhancing precision and efficiency in farming practices. By categorizing MTS into distinct clusters based on similarities in weather patterns, farmers and agricultural stakeholders have information about regionalized climate conditions and its im-

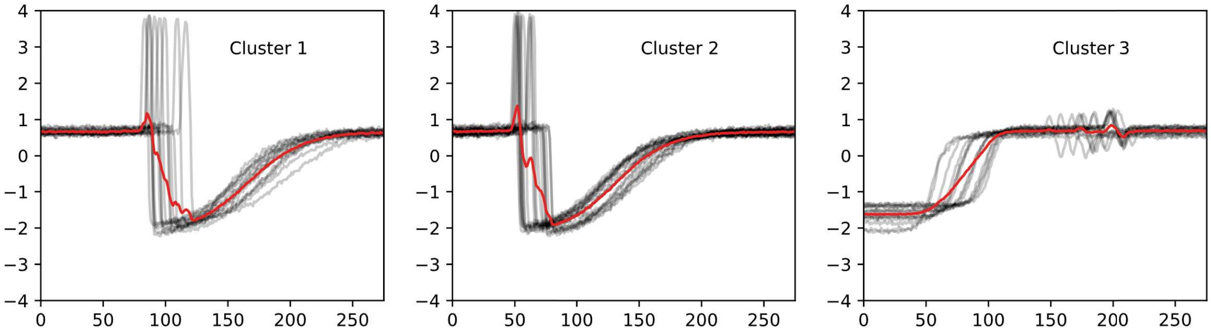


Figure 4 – Example of time series clustering. (TAVENARD et al., 2022)

pact crop growth and yield. This information can inform critical decisions related to planting, irrigation, and crop protection, allowing for the timely adjustment of agricultural activities to optimize resource allocation and reduce the risk of crop losses (ROY; GEORGE K, 2020).

Moreover, the utilization of clustering in agriculture benefits from its capacity to improve decision support systems. By grouping MTS data into clusters and creating region-specific insights, the clustering results can be another input in those systems to improve their recommendations and action plans for farmers. Different clusters may represent diverse microclimates within a given agricultural region, and by tailoring advice to these sub-regions, agriculture stakeholders can adapt their practices to the unique challenges and opportunities presented by the local climate (MUSTARIK; SULTAN; ISLAM, 2021). For instance, clusters characterized by consistent rainfall patterns might require different irrigation strategies than clusters experiencing erratic precipitation. This tailored approach not only improves the efficiency of resource utilization but also promotes sustainable farming practices by reducing water and energy wastage. In summary, the application of clustering techniques to MTS data in agriculture empowers farmers and agricultural decision-makers with actionable insights, fostering more resilient and sustainable agricultural systems in the face of climate variability and change.

However, despite being a powerful alternative, MTS clustering has significant challenges, which makes this process non-trivial. Generally, this task requires a combination of statistical methods for data preprocessing and feature extraction, investigation of efficient ML algorithms, and knowledge of the application domain (GANGULY; STEINHAEUSER, 2008). Encapsulating all of this in a work methodology, with processes automated by software, and presenting results to stakeholders is very challenging and, in this sense, the main contribution of this work is positioned: a framework that assists and automates the task of clustering time series meteorological data used in agricultural applications.

### 2.3.1 Applications in Agriculture

The application of clustering MTS in agriculture holds significant promise for improving agricultural practices, resource management, and crop yield optimization (TIWARI; MISRA, 2011). First, enhancing precision agriculture, where clustering can be used to create microclimatic zones within a farm based on meteorological data. These zones allow for more precise farm activities, such as irrigation, fertilization, and pest control, reducing the overuse of resources and environmental impact. Also, it can help anticipate extreme weather events, such as droughts or storms, on agricultural productivity. By identifying clusters associated with extreme weather conditions, farmers can implement adaptive strategies to mitigate potential crop damage, like introducing drought-resistant crop varieties or enhancing irrigation infrastructure (BEN AYED; HANANA, 2021). Additionally, the clustering of meteorological data can aid in the development of predictive models that anticipate weather-related risks and help farmers make proactive decisions regarding crop selection and risk management.

An efficient way to make use of clustering results in practice is to insert them as complementary input into agricultural decision support systems. Clustering can improve decision support systems recommendations with climate insights, increasing the accuracy of crop forecasts and modeling (SALEEM; POTGIETER; ARIF, 2021). These insights empower stakeholders to make more informed decisions. Also, the integration of MTS clustering within decision support systems strengthens the ability to dynamic adapt to climate variability, updating microregionalization of agricultural areas, in comparison to outdated historical and traditional divisions. By clustering MTS data in real-time, farmers can make timely decisions to optimize resource allocation while minimizing costs and ecological footprints.

## 2.4 Chapter Remarks

This chapter presented basic and fundamental concepts for the discussion about time series, characteristics of time series from meteorological observations, ML, clustering algorithm approaches, and similarity metrics. Knowing that the theory on these subjects is broad and extensive, only introductory aspects were presented here to facilitate the understanding of this thesis, without the intention of covering the matter in its entirety. Based on these basic concepts, in the next chapters some algorithms and metrics are explored in depth, those investigated in the proposed framework for clustering MTS, in order to focus on the scope of the research.

# 3 RELATED WORKS

Clustering of MTS in the agricultural context is extremely useful for improving agricultural decision support systems, mainly through climate zoning. Given the particularities of MTS, the clustering task is complex, usually involving data preprocessing and feature extraction steps, in addition to the need to keep up with the advancement of technology and ML techniques. This chapter presents a descriptive Systematic Literature Review (SLR) on the topic *"Meteorological Time Series Clustering in Agricultural Applications"*, bringing together the main solutions for clustering MTS in agricultural applications, in a context-aware way, mapping the main challenges and seeking to understand the characteristics of the meteorological data, to better understand the applicability of different techniques in the agricultural context. After an initial search, the papers were screened and filtered based on the review protocol, and 26 papers were selected for review. Data about the solutions presented in each paper were then extracted, such as objective, operation, experiments, and evaluation metrics. The main contribution of this research step was the organization of published knowledge on the research topic, in order to identify the state-of-the-art and assist researchers, as well as the discussion and highlighting of possible research directions.

## 3.1 Systematic Literature Review

MTS data, which includes weather variables such as temperature, rainfall, and humidity, have become instrumental in helping farmers make informed decisions. As the agricultural sector increasingly relies on data-driven approaches, a growing body of research has explored the clustering and categorization of MTS data to enhance precision agriculture and address the challenges posed by climate variability. Insights from climate zoning can inform and enhance agricultural strategies. This includes tailoring conditions specific to each crop and anticipating unusual or extreme weather events. Such insights contribute to the improvement of agricultural decision support systems.

With the advancement of ML techniques, in recent years, different solutions for clustering climate data in time series format have been proposed. There has also been an increase in the availability of historical series of climate data, both from public data sources, such as meteorological monitoring institutes, and from private agricultural monitoring initiatives. Therefore, an organization of this knowledge is necessary in order to provide a clear view of the state of the art on this topic, and this is the main objective of this review.

This literature review aims to provide a comprehensive overview of existing research, methods, and challenges in clustering MTS. Its primary objective is to elucidate the current state of knowledge, report, and analyze the different clustering techniques that deal with climate time series data applied in agricultural contexts, and highlight the practical implications for farmers and relevant stakeholders. The main contributions of this review are the following:

- Identify and summarize the state-of-the-art knowledge about clustering meteorological time series in agricultural applications in the last ten years;
- Analyze and discuss the main solutions for this purpose;
- Highlight open perspectives and directions in this subject.

### 3.1.1 Review Methodology

A Systematic Literature Review (SLR) is a comprehensive and structured approach to evaluating existing research on a specific topic or question. It involves systematically searching, selecting, and analyzing relevant studies to provide an unbiased summary of the current state of knowledge (COOPER, 1988). Following the categorization described in (PARÉ et al., 2015), this review presents a specific type of SLR known as *Descriptive Review*. Its main objective is to consolidate knowledge related to the clustering of MTS in agricultural applications.

The adopted protocol for this SLR was that presented in (KITCHENHAM; CHARTERS, 2007) due to its appropriateness for this kind of review and its close alignment with the field of Computer Science. Therefore, this SLR was conducted in three major phases, as outlined in Figure 5. In the first phase, the *Planning* stage, the review protocol was designed, consisting of objective, research questions, search string, research databases, and criteria for inclusion and exclusion of studies. Subsequently, in

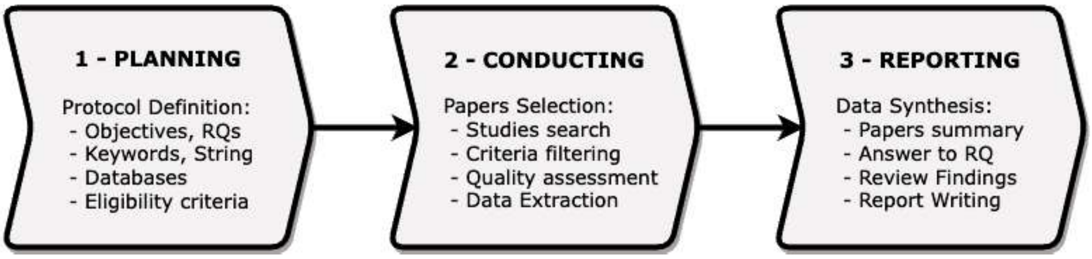


Figure 5 – Systematic Literature Review process flow.

the *Conducting* stage, the pre-defined protocol guides the search for relevant studies and their selection, followed by the assessment of study quality and extraction of pertinent data. Lastly, in the *Reporting* stage, the main review report is generated, offering a condensed summary of information gleaned from the chosen studies. Further elucidation of these review stages is provided below, with a focus on providing in-depth descriptions to enhance the reproducibility of the study. The steps included within the first two phases are detailed in this section. The steps of the third phase are presented in the next section, together with a discussion of the review findings.

To enhance the reliability and credibility of SLRs, authors often employ electronic tools (OKOLI, 2015), such as reference libraries and automation software, to manage the complex process and analyze substantial volumes of data related to the research topic. One notable software tool commonly used for this purpose is *Parsifal*[1], which is an online platform designed to facilitate SLRs. Parsifal offers a collaborative online workspace and documents the entire review process, smoothly integrating different reference management tools such as *Mendeley*[2], used in this SLR. To export and import references into Parsifal, the *BibTex*[3] format was predominantly used. Additionally, spreadsheets were used in this review, primarily for extracting and recording quantitative data, as well as for generating graphical representations.

### 3.1.2 Objective and Research Questions

Based on the motivations briefly presented above, the main objective of this SLR is *to identify the state of the art in solutions for clustering meteorological time series in agricultural applications.*

The *Research Questions (RQ)* were designed to meet the objective of the review, helping to specify the focus for the selection of primary studies as well as data extraction, and to map and analyze existing solutions. Thus, these were the RQs defined for this review:

**RQ1:** *What are the solutions for clustering meteorological time series in the agricultural context?*

**RQ2:** *What are the clustering algorithms used in these solutions?*

**RQ3:** *What data preprocessing or feature extraction techniques are used prior to clustering?*

**RQ4:** *How is the optimal number of clusters defined?*

**RQ5:** *What are the similarity metrics used in these solutions?*

**RQ6:** *What factors motivate the clustering of time series for agricultural purposes and for which crops?*

[1] Parsifal - `https://parsif.al/`

[2] Mendeley - `https://www.mendeley.com/`

[3] BibTex - `http://www.bibtex.org/`

### 3.1.3 Keywords, Search String, and Databases

*Keywords* (and some synonyms) were extracted from the initial definitions of objective, research questions, and research context. They were subsequently used to form the *Search String* for this review. Then, this string was used as the input for querying databases to retrieve studies relevant to the subject of the review. Control articles were used in tests, to verify the string return and, after testing, this was the resulting string:

***Search String:*** `(“time series” OR “temporal series”) AND (“clustering” OR “cluster” OR “grouping”) AND (“climate” OR “meteorological” OR “weather”) AND (“agriculture” OR “agricultural” OR “farm”)`

***Databases:*** The string above was then used to search for primary studies in academic databases, based on their relevance to the field of Computer Science, as follows.

`ACM Digital Library:` http://portal.acm.org

`IEEE Digital Library:` http://ieeexplore.ieee.org

`Scopus:` http://www.scopus.com

`SpringerLink:` https://www.springer.com

The initial search was performed through each database search engine, accessed from the *CAPES*[4] journals portal and using the *CAFe*[5] authentication.

### 3.1.4 Exclusion Criteria

The *Exclusion Criteria (EC)* are designed to filter primary studies that offer direct insight into research questions. Exclusion criteria were established to filter the studies, considering factors of interest, as listed below. The authors' definitions were based on best practices for conducting this type of review, as well as research interest.

***Exclusion Criteria:***

**EC1:** *papers published before 2014*

**EC2:** *duplicate papers on search bases*

**EC3:** *secondary (surveys, reviews) or tertiary studies (meta-analyses)*

**EC4:** *lecture/conference notes and short papers (4 p. or less)*

**EC5:** *not written in English or without access to the full text*

**EC6:** *papers that do not use data in time series format*

**EC7:** *papers focused only on ML tasks other than clustering*

### 3.1.5 Quality Assessment

After filtering and excluding unwanted papers, the next step was to define the Quality Assessment (QA) checklist, in order to evaluate the quality of the accepted studies. QA helps grade papers, supporting authors to observe scientific aspects such as objectives, methodology, experiments and metrics used in the context of each paper. The

[4] CAPES - Portal de Periódicos - https://www.periodicos.capes.gov.br/
[5] CAFe - https://www.rnp.br/servicos/cafe

QA questions used in this review are listed below.

**QA1:** *Does the paper present a solution for clustering time series?*

**QA2:** *Were the time series from meteorological observations?*

**QA3:** *Is the solution aimed at the agricultural context?*

**QA4:** *Is the solution clearly presented (steps, source code, etc.)?*

**QA5:** *Does the paper present experiments?*

**QA6:** *Do experiments include comparisons with other solutions?*

**QA7:** *Does the paper present metrics to evaluate clustering?*

**QA8:** *Does the paper explain the techniques used to calculate the similarity (distance) between elements/centroids in a cluster?*

**QA9:** *Does the paper address limitations of the proposed solution?*

Each question had three possible answers and scores: $Yes - 1.0$; $Partially - 0.5$; $No - 0.0$. From the QA, a score was assigned to each paper and a cut-off point was defined. Since the checklist had nine questions, the maximum score was $9.0$ and the cutoff point was set at $5.5$, based on good SLR practices.

### 3.1.6 Selection Process Overview

The selection process involves the artifacts described so far applied sequentially. Table 1 shows the number of papers excluded at each stage, for each database, up to those accepted. It is important to highlight that the definition of the protocol was carried out jointly by the authors and the review stages, such as screening and quality assessment, were carried out in the peer review format.

First, the *Initial Search* column indicates the number of papers returned in the raw search. Then, the *Excluded Papers* columns indicates the number of papers excluded by each EC, while column *Accepted Papers* shows the number of papers remaining after applying the EC. The ECs were applied sequentially, in ascending order, with the first four (EC1, EC2, EC3, and EC4) being applied one by one, and the other three (EC5, EC6, and EC7) being applied together, in a dynamic review stage analyzing the title and abstract of each paper. After this initial filtering, $80$ papers were accepted for quality assessment. The column *Excluded by QA* shows the number of papers excluded in that phase ($54$), and, finally, column *Selected Papers* indicates the number

Table 1 – Number of papers excluded at each stage and by each exclusion criteria, and those selected for review.

| **Database** | **Initial** | **Excluded Papers** | | | | | **Accepted** | **Excluded** | **Selected** |
|---|---|---|---|---|---|---|---|---|---|
| | **Search** | **EC1** | **EC2** | **EC3** | **EC4** | **(EC5, EC6, EC7)** | **Papers** | **by QA** | **Papers** |
| **ACM** | **376** | 37 | 1 | 41 | 208 | 79 | **10** | 7 | **3** |
| **IEEE** | **42** | 7 | 0 | 1 | 1 | 20 | **13** | 9 | **4** |
| **Scopus** | **139** | 26 | 12 | 3 | 2 | 46 | **50** | 35 | **15** |
| **Springer** | **117** | 24 | 0 | 39 | 1 | 46 | **7** | 3 | **4** |
| **Total** | **674** | **94** | **13** | **84** | **212** | **190** | **80** | **54** | **26** |

Table 2 – Identifier Code, references and titles of the 26 papers selected for SLR.

| ID | Reference | Title |
|---|---|---|
| P1 | (WEISSTEINER et al., 2019) | A Crop group-specific pure pixel time series for Europe |
| P2 | (WANG; JIA; XIAO, 2023) | A Hybrid Approach Based on Unequal Span Segmentation -Clustering for Short-Term Wind Power Forecasting |
| P3 | (AZIMI; GHOFRANI; GHAYEKHLOO, 2016) | A hybrid wind power forecasting model based on data mining and wavelets analysis |
| P4 | (YANG et al., 2020) | A robust method for generating high-spatiotemporal-resolution surface reflectance by fusing MODIS and Landsat data |
| P5 | (SOHOULANDE et al., 2019) | An investigation of seasonal precipitation patterns for rainfed agriculture in the Southeastern region of the United States |
| P6 | (LI; CHEN; CHEN, 2018) | An SNN Ontology Based Environment Monitoring Method for Intelligent Irrigation System |
| P7 | (DENG et al., 2014) | Analyzing Wind Speed Data through Markov Chain Based Profiling and Clustering |
| P8 | (AHER; YADAV, 2021) | Assessment of precipitation trends and its implications in the semi-arid region of Southern India |
| P9 | (FERRELLI et al., 2019) | Climate regionalization and trends based on daily temperature and precipitation extremes in the south of the Pampas (Argentina) |
| P10 | (DE OLIVEIRA et al., 2023) | Clustering Weather Time Series used for Agricultural Disease Alert Systems in Florida |
| P11 | (GUO et al., 2020) | Data mining algorithms for bridge health monitoring: Kohonen clustering and LSTM prediction approaches |
| P12 | (ETIENNE et al., 2016) | Development of a Demand Sensitive Drought Index and its application for agriculture over the conterminous United States |
| P13 | (MA et al., 2020) | Early warning indexes determination of the crop injuries caused by waterlogging based on DHSVM model |
| P14 | (WANG et al., 2021) | Exploring the optimal crop planting structure to balance water saving, food security and incomes under the spatiotemporal heterogeneity of the agricultural climate |
| P15 | (CONRADT; GORNOTT; WECHSUNG, 2016) | Extending and improving regionalized winter wheat and silage maize yield regression models for Germany: Enhancing the predictive skill by panel definition through cluster analysis |
| P16 | (OLIVEIRA-JÚNIOR et al., 2021) | Fire foci in South America: Impact and causes, fire hazard and future scenarios |
| P17 | (BREGAGLIO et al., 2023) | Improving crop yield prediction accuracy by embedding phenological heterogeneity into model parameter sets |
| P18 | (SHU et al., 2019) | On the Selection of Features for the Prediction Model of Cultivation of Sweet Potatoes at Early Growth Stage |
| P19 | (CHEN; ZWART; JIA, 2022) | Physics-Guided Graph Meta Learning for Predicting Water Temperature and Streamflow in Stream Networks |
| P20 | (SATHIARAJ; HUANG; CHEN, 2019) | Predicting climate types for the Continental United States using unsupervised clustering techniques |
| P21 | (CHEN et al., 2016) | Scalable nearest neighbor based hierarchical change detection framework for crop monitoring |
| P22 | (ZHANG; LI; ZHANG, 2020) | Short-term wind power forecasting approach based on Seq2Seq model using NWP data |
| P23 | (REYES et al., 2023) | Soil properties zoning of agricultural fields based on a climate-driven spatial clustering of remote sensing time series data |
| P24 | (MULLAPUDI et al., 2023) | Spatial and Seasonal Change Detection in Vegetation Cover Using Time-Series Landsat Satellite Images and Machine Learning Methods |
| P25 | (ROBINSON et al., 2021) | Temporal Cluster Matching for Change Detection of Structures from Satellite Imagery |
| P26 | (OLIVEIRA-JÚNIOR et al., 2021) | Wet and dry periods in the state of Alagoas (Northeast Brazil) via Standardized Precipitation Index |

of papers that obtained a QA score above the cutoff point ($26$), and were therefore selected for review.

Table 2 presents initial information about the $26$ papers selected for SLR, such an: identifier code (`ID` column) assigned to each paper in aphabetical order by title, for future textual references; the bibliographic reference of each paper (`Reference` column); and, the titles of the selected papers (`Titles` column)

### 3.1.7 Data Extraction

To extract data from the selected papers, a form was created to record the information in a structured way. The Parsifal tool has an interface for this and allows the definition of a specific data type for the answers (string, boolean). To save space, the form will not be fully detailed here, but it contained questions about authors, institution, country, publication date, motivation, meteorological variables, time range, study location, datasets, data aggregation or normalization, clustering techniques, cluster evaluation, metrics, agricultural activity and crops, devices, application layer, and implementation. The extracted data was then used to compose the results presented in the next section.

## 3.2 Reporting and Discussion

After the planning and conducting phases, the report of the primary studies was conducted. The papers were reviewed and summarized according to the data extraction form, defined in the protocol. This section presents a summary of the SLR results, containing the answers to the research questions and the main outcomes of this review. The main challenges faced in their research on methodological time series clustering are also described and, subsequently, the threats and limitations of this review are discussed.

### 3.2.1 References Synthesis

After selecting the studies, the 26 selected papers were carefully reviewed and summarized in order to enable the extraction of insights from the data. Table 3 presents a summarization of the information extracted from selected papers, in order to help compare and relate the works. The first column (`ID/Ref`) contains an identifier assigned to each paper, based on the titles in ascending alphabetical order, and the bibliographic reference. The second column (`Ctry`) indicates the country of institution of the first author of each paper. The third (`Study Region`) and fourth (`Time Range`) columns refer to the study cases presented in the papers, indicating the geographic location and the time interval in which the data were collected. The fifth column (`Agric. Interest`) presents the agricultural aspect of interest for each paper, that is, what was under

Table 3 – Overview summary of the reviewed papers, highlighted (shading) those entirely fitted to the topic of this RSL – agricultural context and dealing with meteorological time series.

| ID | Ctry | Study Region | Time Range | Agric. Interest | Target Crop | Time Series Source | Cluster Alg. |
|---|---|---|---|---|---|---|---|
| P1 | ITA | European Union | 2001-2017<br>17 years | Crop Phenology | winter, spring, and summer crops | NDVI | GMM |
| P2 | CHN | Inner Mongolia (CHN) and Yalova (TUR) | 2015, 2018<br>2 years | - | wind farm* | Wind Speed | Fuzzy C-means |
| P3 | IRN | 16 USA States | 2008-2012<br>5 years | - | wind farm* | Temperature, Wind Speed and Direction | TSB K-means |
| P4 | CHN | Heilongjiang (CHN) | 2017-2018<br>2 years | Land Cover | vegetation | Surface Reflectance | Fuzzy C-means |
| P5 | USA | Southeastern USA | 1960-2017<br>58 years | Water | cotton, peanuts corn, soybean | Precipitation | K-means |
| P6 | CHN | China | 2017<br>1 year | Soil | all crops | Soil Moisture and Temperature, Air Temperature, Air Humidity, and Precipitation | K-means, Spectral, GenClus |
| P7 | NZL | Cape Canaveral (USA) | 2011, 2013<br>2 years | - | wind farm* | Wind Speed | Spectral |
| P8 | IND | Southern India | 1980-2013<br>34 years | Water | all crops | Rainfall | Hierarchical |
| P9 | ARG | South Pampas (ARG) | 1970-2017<br>47 years | Water | all crops | Temperature, Precipitation | Hierarchical |
| P10 | BRA | Florida (USA) | 2005-2023<br>18 years | Disease Control | strawberry, citrus, blueberry | Temperature, Relative Humidity | K-means |
| P11 | CHN | Hubei (CHN) | 2017-2018<br>2 years | - | - | Temperature, Humidity, Wind | Kohonen NN |
| P12 | USA | USA | 1949-2010<br>62 years | Drought | corn, soybeans, hay, wheat, barley, sorghum, rice, cotton | Temperature, Rainfall, Wind Speed | K-means |
| P13 | CHN | Jianli County, Hubei (CHN) | 1970-2015<br>45 years | Flood | cotton | Rainfall, Soil Moisture | K-means |
| P14 | CHN | Liaoning (CHN) | 1999-2016<br>18 years | Planting | maize, sorghum, rice, wheat | Temperature, Precicipation Wind Speed, Humidity, and Sunshine Duration | K-means |
| P15 | DEU | Germany | 1901-2010<br>110 years | Yield | winter wheat, silage maize | Temperature, Precipitation, and Solar Radiation | Hierarchical, PAM, EM |
| P16 | BRA | South America | 1998-2018<br>21 years | Fire Foci | all crops | MFDI | Hierarchical |
| P17 | ITA | Apulia, Tuscany, and Veneto (ITA) | 2007-2019<br>13 years | Planting | barley, maize | NDVI | Hierarchical |
| P18 | TWN | Taiwan | 2010-2014<br>5 years | Planting | sweet potato | Air and Soil Temperature, Dew Point, Wind Speed, Evaporation, Precipitation, Relative Humidity, and Sunshine Hours | Hierarchical |
| P19 | USA | Delaware River Basin (USA) | 1980-2020<br>41 years | Water | all crops | Air Temperature, Precipitation, and Solar Radiation | K-means |
| P20 | USA | USA | 1946-2015<br>70 years | Weather | all crops | Temperature and Precipitation | K-means, DBSCAN, BIRCH |
| P21 | USA | Iowa and Dakota (USA) | 2001-2011<br>11 years | Crop Phenology | all crops | NDVI | Hierarchical |
| P22 | CHN | Galicia (ESP) | 2016<br>1 year | - | wind farm* | Wind Speed and Direction, Air Pressure, Temperature, and Humidity | K-means |
| P23 | ITA | Umbria and Lazio (ITA) | 1984-2021<br>38 years | Soil | herbaceous and tree crops | NDVI | K-means |
| P24 | IND | Akole (IND) | 2016/17/21/22<br>4 years | Land Cover | vegetation | NDVI | K-means, ISODATA |
| P25 | USA | Delaware, Maryland, and Virginia (USA) | 2011-2020<br>10 years | - | poutry barns* | Satellite Imagery | K-means |
| P26 | BRA | Alagoas (BRA) | 1960-2016<br>57 years | Water | all crops | Rainfall | Hierarchical |

*Not an agricultural activity in itself, but it can be related to agricultural crops or tasks.

monitoring. The sixth column (`Target Crop`) shows the agricultural crop to which the solution was focused. The seventh column (`Time Series Source`) presents the source of the time series clustered in each paper. The eighth column (`Cluster Alg.`) presents the clustering algorithm(s) investigated in each paper.Data extraction sought to organize the information in a way that makes it easier to identify relationships between the papers, but strictly following the nomenclatures that the authors indicated in the original papers.

From this table and the extraction of data on the publication of the papers, it is possible to carry out some initial general analyzes on the papers reviewed. First, no predominant or preferred publication channel was identified. The $26$ selected papers were published in $24$ different periodicals, including peer-reviewed journals and conference proceedings. Then, For a temporal analysis of publications, Figure 6 shows the distribution of papers by year of publication. It is possible to observe the growth in the number of publications on the topic of this review over the last ten years, showing that the clustering of MTS is currently a topic of interest to the scientific community. The advancement of technology (IoT and meteorological sensors) and the popularization of ML techniques (languages and libraries) certainly contributes to this number of publications in recent years. The recent exception was 2022, which can be attributed to the gap in scientific conferences during the Covid pandemic period.

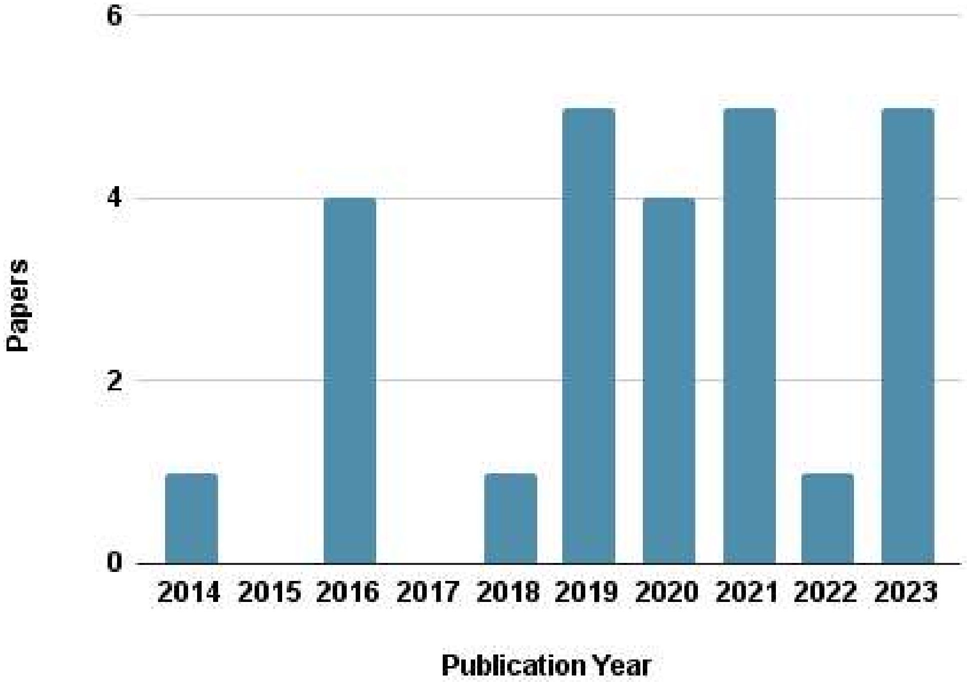


Figure 6 – Distribution of reviewed papers by publication year.

Based on Table 3, from the `Study Region` column it is possible to analyze the global spatial distribution of papers, observing which areas the papers are directed to. Figure 7 graphically presents the distribution of papers for each country, on a colored scale

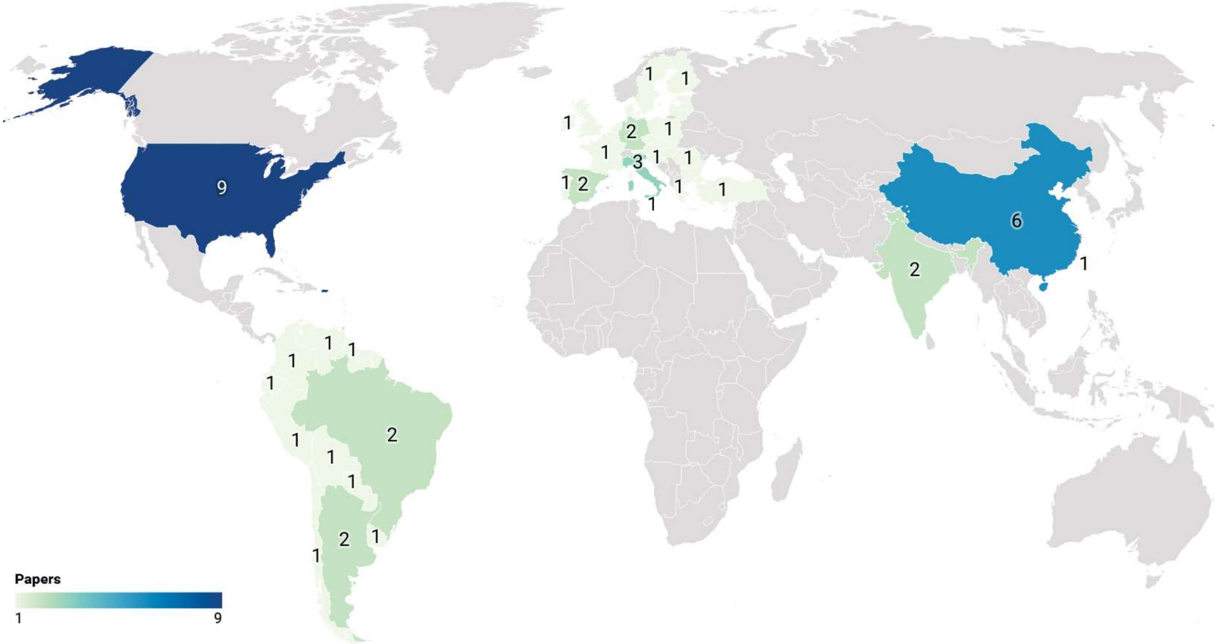


Figure 7 – Distribution of reviewed papers by area of study.

between 1 and 9. It is important to highlight that this distribution does not refer to the country of the authors or their institutions, nor the country of publication of the paper, but is based on the target study region of each paper. The country with the most studies focused on it, on the topic of RSL, is the USA (9 articles), followed by China (6). The other areas appear in smaller numbers, with a few studies analyzing data from more than one country, covering an entire continent, for example. The distribution of the graph follows the order of world agricultural production, led by countries such as the USA, China, India, and Brazil (SERVICE, 2023).

### 3.2.2 Answer to Research Questions

To answer the research question **RQ1 (What are the solutions for clustering meteorological time series in the agricultural context?)**, Table 3 can be used, by searching among the reviewed papers those that deal specifically with the subject of this review: *clustering of meteorological time series in an agricultural context*. This is important because not all papers selected for review fit this entire condition, even if they were accepted in the review filters. Therefore, the papers that make up the answer to this first research question are papers $P5, P6, P8, P9, P10, P12, P13, P14, P15, P18, P19$ and $P20$, which appear highlighted in light gray in Table 3. Some papers deal with time series clustering, but are not focused on the agricultural context (null value in `Agric. Interest` column - $P2$, $P3$, $P7$, $P11$, $P22$ and $P26$); others are focused on the agricultural context, however, they do not manipulate time series that represent meteorological variables (`Time Series Source` column - $P1$, $P4$, $P16$, $P17$, $P21$, $P23$, $P24$ and $P25$), but other data types, such as agricultural indexes derived from satellite images or other sources.

By analyzing the twelve highlighted papers, which are those entirely related to the topic of this review, as mentioned before, it is possible to answer **RQ2 (What are the clustering algorithms used in these solutions?)**. The algorithms used by these papers are predominantly *K-means* and *Hierarchical* clustering. In all highlighted papers, one of the two was used as the main solution or comparison baseline. Both are the best-known algorithms for the clustering task, and this justifies their appearance in so many papers. Regarding hierarchical clustering, in all reviewed papers the agglomerative approach was used, where each item in the set starts as a small cluster and, subsequently, the items merge into larger clusters. With unique occurrences, in different papers, the following algorithms were also investigated: *Spectral Clustering*, *GenClus* (genetic algorithm), *PAM* (Partition Around Medoids), *EM* (expectation-maximization), *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise), and *BIRCH* (Balanced Iterative Reduction and Clustering using Hierarchies). However, they were briefly presented in the papers, in order to compare them with the main algorithms, so the analysis did not delve into how they work.

To complement the answer to this and other RQs, Table 4 presents a general summary of the methodology used in the twelve papers mentioned above, which fully meet the scope of this SLR, separated by the clustering algorithm used (columns `Algorithm` and `Paper`). It presents the techniques used for data preprocessing (column `Preprocessing`) and feature extraction (column `Feature Extraction`), as well as the method used to define the number of clusters (column `N of Clusters`), and the similarity metrics used to define the degree of membership of elements to clusters (column `Similarity Metrics`). Furthermore, it presents the clustering implementation environment (column `Implementation`) and the agricultural purpose (column `Agric. Purpose`).

Research question **RQ3 (What data preprocessing or feature extraction techniques are used prior to clustering?)** is answered based on Table 4, observing columns `Preprocessing` and `Feature Extraction`. The first presents preprocessing techniques used in some papers, mostly related to: data quality checking, in order to deal with common problems in the manipulation of datasets such as missing values, nulls and outliers; and date filtering, such as the seasons of interest for each crop. It is noted that few authors used data preprocessing techniques, which may be related to the quality and robustness of the data sources, where possibly those who do not preprocess the data are because they do not observe significant data quality problems or perform aggregations or transformations that overcome this type of problem. Regarding Feature Extraction, the other column related to this RQ, an intense use of techniques for this purpose is observed. This step is crucial in training ML models and there are many techniques, mainly statistical and focused on time series, that can be

Table 4 – Summary of algorithms operation, preprocessing, feature extraction, metrics and implementation.

| Algorithm | Paper | Preprocessing | Feature Extraction | N of clusters | Similarity Metric | Implementation | Agricultural Purpose |
|---|---|---|---|---|---|---|---|
| K-means | P5 | Filling missing values, Filtering (season) | Varimax Rotation (PCA) | Jaccard Index | Jaccard Index | - | Identify precipitation homogeneous regions |
| | P6 | - | - | known (context) | Euclidean Distance | - | Improve accuracy of soil sensors data |
| | P10 | Quality Checking (missing and null values), Filtering (season) | Data Aggregation | Elbow Method, Silhoutte Score | Dynamic Time Warping | Python | Identify agroclimatic regions to monitor fungal diseases |
| | P12 | - | Data Aggregation | Silhouette Score | - | - | Classify drought regions |
| | P13 | - | Statistics | known (context) | - | ENVI Software | Classify watterloggins |
| | P14 | - | - | known (context) | - | SPSS Software | Optimize planting structure and water footprints |
| | P19 | - | Data Aggregation and Augmentation | Prediction RMSE | Contrastive Loss | - | Identify similar river segments |
| | P20 | - | Data Aggregation | Homogeneity, V-measure | Euclidean Distance | Python | Classify climate regions |
| Agglomerative Hierarchical Clustering | P8 | - | - | Dendogram, Scree Plot | Square Euclidean Distance, Ward's | Statistica Software | Identify precipitation trends for rainfall monitoring |
| | P9 | Filling missing values, Quality and Homogeneity Controls (PCA) | Dimension Reduction (PCA) | Elbow Method | Euclidean Distance, Ward's | R | Identify similar regions by water availability |
| | P15 | - | Interpolation | Silhouette Score | Euclidean Distance | R | Improve the predictive power of crop models |
| | P18 | Filtering (planting days) | Colinearity Feature Grouping | Dendogram, Silhouette Score | Ward's | - | Identify agroclimatic conditions after planting |

used. It is observed that the most used were: *Principal Component Analysis (PCA)*, analyzing summary time series indices; *Data Aggregation*, representing values from multiple timestamps of a time series into average or cumulative values; *Data Augmentation*, to increase the size of the existing dataset; and general *Statistical Analyses*, such as interpolation, statistical tests and feature grouping. The use of these techniques reinforces the importance and need for extracting valuable information from a dataset, seeking to improve the performance of clustering algorithms.

To answer **RQ4 (What methods are used to define the optimal number of clusters?)**, another column of the same table is analyzed (`N of clusters`). Defining the number of clusters to be grouped in the dataset is a fundamental step in the clustering task and has a direct relationship with the performance of algorithms, such as K-means. There is a range of techniques that can be used and in the papers analyzed the *Silhouette Score* is most often used. The *Elbow Method* and *Dendogram* analysis also appear more than once. It was also observed that some papers did not present a discussion on how they defined the optimal number of clusters, since, based on the context, it had already been defined into how many groups the data should be categorized.

Regarding **RQ5 (What are the similarity metrics used in these solutions?)**, the `Similarity Metrics` column presents a summary of the metrics used in the papers. These metrics indicate aspects such as the distance between elements in a cluster and their centroid, and the degree of membership of elements in a cluster. The most used metrics in the articles reviewed were: *Euclidean Distance*, which basically represents the length of a line segment between the two points; and *Ward's Method*, which bases the choice of the pair of clusters to be merged at each step based on the optimal value of an objective function. In the same way as the definition of an optimal number of clusters, these similarity metrics represent generalist aspects and are often used in more than one complementary way.

To answer question **RQ6 (What factors motivate the clustering of time series for agricultural purposes and for which crops?)**, Tables 3 and 4 are referred again. In Table 3, column `Agric. Interest` indicates the aspect of agricultural interest, classifying the papers based on the element that was monitored, helping to relate works that address the same agricultural aspect. Therefore, greater interest was observed among these papers in aspects such as *soil* and *cover land*, *water availability*, and *weather conditions*, mainly interested in extreme events (fire, flooding, drought). Additionally, column `Target Crop` indicates which culture is related to that analysis, since, in the same location, under the same weather conditions, different cultures may have different behaviors.

Furthermore, the `Agric. Purpose` column in Table 4 presents the purpose of using clustering in the agricultural context, from a more technical perspective, related to the application of the algorithm. The purposes can be grouped into four main categories: monitoring crops, soil, water and weather. For all of them, with the aim of zoning different variables, to serve as a basis for agricultural practices. Figure 8 presents an infographic summarizing the main motivations of the reviewed papers, the related agricultural activities and the valuable insights from them. The majority of papers used clustering to identify areas with similar conditions, especially regarding climatic variables. The main objective of climate zoning, in all papers, is to improve decision-making, using the results of clustering to support agricultural decisions. It was identified that the main contribution of clustering in this context is to assist agricultural stakeholders, such as producers and researchers, extracting valuable insights from meteorological data, to enhance productivity, area management and sustainability.

## 3.3 Main Outcomes and Challenges

The results obtained from the SLR, especially the answers to the research questions, provide compelling findings that can be applied to further research on clustering MTS in agricultural context. Analyzing the objective of each solution, the operation steps, the data source and the algorithms used, challenges were observed that all authors faced, when dealing with this type of time series, in this context. In summary, it can be stated that the main outcomes of this SLR are:

- **Research Framework:** among the 26 reviewed papers, only one ($P14$) proposed a research procedure, such as a framework or a guide with well-defined steps.

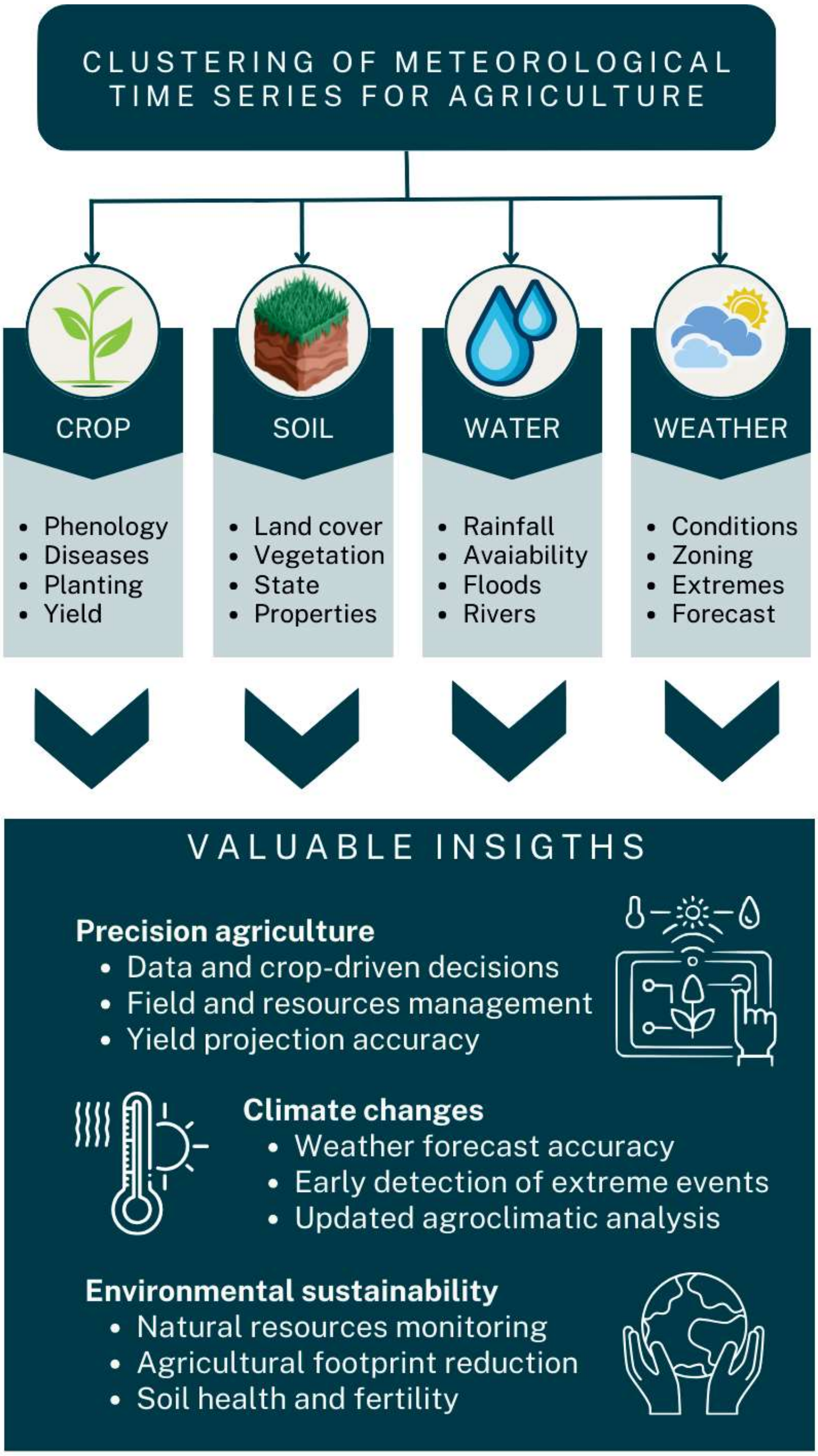

CLUSTERING OF METEOROLOGICAL TIME SERIES FOR AGRICULTURE
CROP
Phenology
Diseases
Planting
Yield
SOIL
Land cover
Vegetation
State
Properties
WATER
Rainfall
Avaiability
Floods
Rivers
WEATHER
Conditions
Zoning
Extremes
Forecast
VALUABLE INSIGTHS
Precision agriculture
Data and crop-driven decisions
Field and resources management
Yield projection accuracy
Climate changes
Weather forecast accuracy
Early detection of extreme events
Updated agroclimatic analysis
Environmental sustainability
Natural resources monitoring
Agricultural footprint reduction
Soil health and fertility

Figure 8 – Main motivations for using MTS clustering in agricultural applications.

The other papers are based on very specific solutions, without exploring or discussing whether the preprocessing, feature extraction, and clustering steps can be generalized. This does not diminish the scientific contribution of these papers, however, it leaves room for research in this direction. The existence of a framework for working with MTS in agricultural applications could boost the development of future researches, bringing together good practices and a set of context-aware methods and metrics.

- **Implementations reusability:** the reviewed papers are implemented in different ways (languages and softwares). The implementations do not discuss reusability aspects, being quite specific for each case. Even though they make use of libraries and functionalities of these software, the authors compose a mosaic with existing alternatives, which is not always efficient or does not meet the entire need. The lack of a software interface, such as a library or API, which brings together, for example, the implementation of functions for preprocessing, feature extraction and clustering, could also speed up implementation, bringing together suitable implementations, and even enabling collaborative coding.

- **Data Preprocessing and Feature Extraction:** There are many existing techniques for data preprocessing, such as filtering and quality checking, as well as for extracting features, with statistical analyzes. When it comes to meteorological temporal data, some definitions could be established based on a research protocol, such as a framework, bringing together good practices for preprocessing this data, in a way that is aware of the agricultural activities, for example, filtering by crop season, rain season, or planting period. In the same way with feature extraction, bringing together the most used statistical techniques for analysis of time series components, aggregation, dimensionality reduction and data augmentation. These analyzes appear unevenly in the works, some using preprocessing techniques, others extracting features, sometimes without exploring all the possibilities. The use of these techniques can make the results of the solutions more robust and increase the reliability of the studies.

- **Evaluation Metrics:** the reviewed papers present different metrics for evaluation for each case. Because each one has a different purpose, what you want to evaluate differs. Even so, there are some general metrics that can be useful in most cases, to assess aspects such as the similarity of elements in a cluster, the distance between elements and their cluster centroid, and the degree of belonging to a cluster. However, they are not always used due to the need to implement it manually or because the software in use does not have this metric implemented.The implementation of a set of performance metrics, suitable for the agricultural context, could also help researchers, enabling the evaluation of solutions in a more comparative way. Furthermore, the metrics observed in the papers generally refer to techniques based on statistics and, due to their generalist nature, they rarely present an excellent result, which is why more than one technique is often used. A more local and specific approach can be useful in creating specific metrics for agricultural applications.

- **Brazil/South America Data Exploration:** Analyzing the global scenario of agricultural production, where Brazil is one of the five largest producers, and from the perspective of the authors of this SLR, researchers from Brazilian institutions, it is observed a very small number of papers that deal with data from Brazil and the South America. Only two papers analyze data from Brazil, one of which deals with the continent as a whole. This is very little, given the country's relevance in the market, the size of the country and the importance of agriculture in the country's economy. Therefore, there is a lot of room for research in this direction, with interesting economic potential. Given the particularities and characteristics of each country, research on MTS in the local agricultural context can even result in the development of research methodologies applied especially to the Brazilian agricultural environment, which differs from other regions of the world in terms of availability of natural resources and technological.

Of course, there are more aspects that could be analyzed, but for the purpose and scope of this SLR, these were the main challenges and possible outcomes extracted from the review process. This note is expected to assist researchers and direct future research towards these directions, in order to contribute to the advancement of the state of the art in the clustering of MTS in agricultural applications.

### 3.3.1 Limitations and Threats to Validity

The SLR presented here, even though it follows a robust and well-established methodology, is subject to threats to its validity, like any scientific work. The authors' main concern at all stages of review was about bias, such as poor understanding of papers, due to the reviewers' prior knowledge (or lack thereof). Therefore, some precautions were taken to avoid bias problems, both when searching and filtering studies.

Regarding the search process, the most recurrent keywords on the topic and four relevant databases were used to ensure the reliability of the search. As there was a possibility that some relevant work had not been returned in the search, in order to minimize this threat, a search string calibration phase was carried out, which resulted in the adding synonyms for keywords in the search string, to increase search coverage. Undetected problems in the database search engines could also have affected this process, so the search was repeated by all researchers in different browsers, looking for differences between results, in order to mitigate this threat as well, and none difference was identified.

Regarding selection and review steps, the concern is due to the understanding of a paper could vary according to the reviewer's background. Therefore, to ensure the reliability of the screening process, all reviews were carried out by pairs of authors, and every time there was disagreement regarding the application of any exclusion criteria, quality assessment, or understanding of the solution presented in one paper, it was reviewed and discussed by everyone, until a consensus understanding was reached. This way, there was less room for erroneous or dubious interpretations and this reduced the threat of bias at this stage as well. Finally, also regarding the selection, it was observed that some papers selected for review were not fully related to the interests of this review (answer to RQ1). However, it was decided to report them because they described the use of clustering techniques and were interesting to know.

## 3.4 Chapter Remarks

This chapter presented a systematic descriptive review on MTS clustering in agricultural applications, following a well-defined protocol widely used in computer science research. Starting from an initial search of 674 papers, after screening, exclusion filters and quality assessment, 26 papers were selected for the complete review. From the selected papers, existing solutions for clustering MTS in the agricultural context were then summarized through comparative tables and figures, in order to answer the research questions. Among the main information extracted are data about study regions, clustering techniques, similarity metrics, and motivation for using the techniques in the agricultural context.

# 4 PROPOSED METHODOLOGY

This chapter presents the proposed methodology for clustering meteorological time series in in the format of a framework, from its conception, design, and general flow of operation. The framework consists of a series of steps designed to facilitate the clustering of MTS in agricultural applications, going through data extraction, filtering and quality checking, analyzing the features for agricultural applications, and then proceeding to clustering and validation. The research decisions taken during the design of this framework, such as definition of work steps, sequence, choice of clustering algorithms and similarity metrics, are strongly based on the results of the systematic literature review, reported in chapter 3.

## 4.1 Overview

A framework refers to a structured and organized way of approaching a particular topic or problem (GUILLÉN et al., 2016). In research, it provides a conceptual structure or theoretical foundation for conducting research and helps researchers organize their thoughts and methods. Frameworks serve as guides for how research questions are formulated, data is collected and analyzed, and conclusions are drawn. There are different types of frameworks, depending on the nature of the study and the research goals. Within the scope of this thesis, a methodological framework is proposed. A methodological framework outlines the research methods, tools, and techniques that will be used to collect and gather data during the research (MCMEEKIN et al., 2020), serving as a blueprint for data collection and research design.

Frameworks are essential in research because they provide structure and guidance, and facilitate the organization of data and findings. They also make it easier for other researchers to understand and evaluate the research and its contributions to the field. Reporting research steps is fundamental as scientific work, in order to allow the reproducibility of this research.

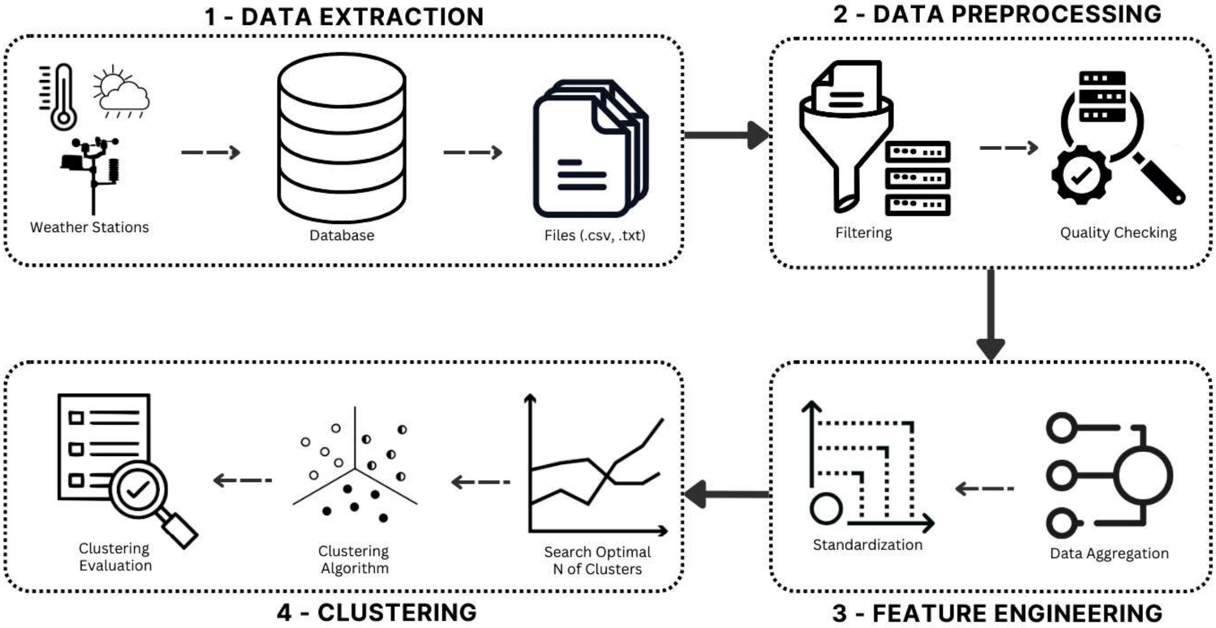


Figure 9 – FMTSClust Overview: Proposed framework process flow.

## 4.2 General Structure

Thus, within the scope of this thesis, a methodological framework is proposed, called **FMTSClust** supported by an API (Application Programming Interface). The name stands for **F**ramework for **M**eteorological **T**ime **S**eries **Clust**ering. FMTSClust proposes a well-defined structure, in order to guide the procedures and assist research on clustering of MTS in agricultural applications. It includes data analysis steps, from data collection, preprocessing, feature engineering and clustering. Figure 9 presents an overview of the entire framework process, which will be detailed below.

## 4.3 Procedures

The proposed research framework provides a sequence of procedures in the design of the MTS clustering, ensuring that their data collection and analysis are consistent with the overall goals in the agricultural context, and also helping to interpret findings. The framework proposes four major procedures, Data Extraction, Data Preprocessing, Feature Engineering and Clustering, which are broken down into smaller specific tasks, as follows.

### 4.3.1 Data Extraction

Data extraction is the first step of this methodology and is based on obtaining data collected by weather stations. These stations have a set of sensors, with equipment such as: thermometers, to measure temperature; barometers, to measure atmospheric pressure; anemometers, to measure wind speed and direction; hygrometers, to measure humidity; pyranometers, to measure solar radiation; rain gauges, to measure precipitation; and, soil moisture sensors, to measure soil moisture content. This equipment is on all the time, recording meteorological observations periodically, usually every few minutes.

#### *4.3.1.1 Data Source*

Weather stations store the collected data on-site using data loggers or other data storage devices. These data loggers record the measurements and store them in digital files. Some modern weather stations may also have the capability to transmit data wirelessly to a central server for storage. The raw data collected from the sensors can be processed to derive additional information, such as dew point, wind chill, heat index, or evapotranspiration. This processed data provides a more comprehensive view of the local weather conditions. The data is then transmitted to a server for storage and dissemination. This can be done through wired or wireless connections. Common methods of data transmission include Wi-Fi, cellular networks, radio frequencies, or satellite communication (DEBAUCHE et al., 2022).

#### *4.3.1.2 Files Format*

The collected and stored meteorological data is made available for agricultural applications through various means, the most common being through web-based platforms. In these systems, anyone can access meteorological data, downloading them in files (in formats such as `.csv, .xls, .txt`), or viewing them with applications that provide historical and real-time meteorological information, with personalized features for agriculture, such as crop-specific weather forecasts, disease risk models, and irrigation recommendations.

Data can also be made available through *Application Programming Interfaces (APIs)* integration, where agricultural applications and software can integrate with weather data using APIs provided by weather service providers. This allows for real-time data retrieval and automated decision-making based on weather conditions.

### 4.3.2 Data Preprocessing

From the extracted data, the next step is its preprocessing, to make it more suitable for data analysis. In this framework, two basic preprocessing steps are proposed, filtering and quality checking, in order to select stations, variables and time periods of interest for agriculture applications, as well as ensuring data quality, as detailed below.

#### *4.3.2.1 Filtering*

When extracting climate data from the perspective of agricultural applications, it is important, first, to define the set of weather stations of interest for analysis. In some cases, not all weather stations are of interest, as agricultural production is conducted in a specialized manner. Some regions concentrate some crops, while others are more conducive to other crops, and the perspective from which you want to cluster the data influences this stage.

Then, it is important to select the variables of interest and the time period, from the perspective of the analysis. In the agricultural context, this phase is generally related to some agricultural activity, such as planting, harvesting or spraying inputs, and the period of time in which this activity occurs. For planting, for example, it may be necessary to identify the rainfall pattern during the period indicated for this activity, while for crop growth it may be necessary to pay more attention to extreme temperatures throughout the production cycle, and for disease control it may be necessary monitor temperature and relative humidity during the flowering period.

Furthermore, the data volume must be checked for each station where data is available. Because weather network stations are not all deployed at the same time, it is common for some stations to have more data than others. Also, some stations may be turned off after a period of activity, others may be very recent, with insufficient data volume for climate analysis. This verification aims to ensure that all stations under analysis have a volume of data equal to or at least representative of the period and variables that are desired to be clustered.

It is important to highlight that the filtering of variables and time window is not arbitrary or exclusively up to those who will apply the framework or perform the clustering. This definition comes from agricultural stakeholders, knowledge of the domain, and agricultural aspect of interest, therefore it is suggested that these definitions be made among all actors in the context.

#### *4.3.2.2 Quality Checking*

Quality checking in weather data helps to ensures the accuracy, reliability, and integrity of meteorological information. This proccess involves examination and validation of meteorological observations, to identify and correct errors, anomalies, or inconsistencies that may arise from instrumentation issues or data transmission. There are different complementary ways to perform quality checking, and the choice of which methods to use may depend on available resources.

Usually, the quality check starts by checking for missing values. Gaps in climate datasets are common, caused by various reasons, such as sensor malfunction, hardware degradation, configuration errors, erroneous human inputs or communication failures between the sensor devices and the central server. Depending on the specific circumstances surrounding each missing data point, corrective measures may be considered or, alternatively, the data points may be removed from the dataset if their prevalence does not pose a substantial threat to the overall integrity of the dataset (GANDIN, 1988).

Another important quality check is the detection and analysis of outliers, to identify discrepant values with the distribution of each variable. These outliers possibly represent measurement errors, such as sensor failures or failures in sending data to the server. There are some strategies that can be applied, and in this framework the outlier capping method is proposed using Interquartile Ranges (IQR) (FRERY, 2021). The IQR is a statistical measure used to describe the spread or dispersion of a data set and quantifies the range of values in the central part of the data, providing a more reliable measure of variability compared to the full range or standard deviation, which are sensitive a Extreme values. To calculate the IQR, for each variable, the first quartile ($Q1$) at the 25th percentile and the third quartile ($Q3$) at the 75th percentile are needed, so $IQR = Q1 - Q3$. Therefore, based on the IQR, if a value is more than $1.5 * IQR$ above the top quartile (Q3) or less than $1.5 * IQR$ below the bottom quartile (Q1), the value is considered an outlier. Limiting means setting an upper and lower limit for outliers and replacing the outliers with the bounded values.

### 4.3.3 Feature Engineering

Feature engineering is the step of preparing data for ML models. It involves selecting, transforming, and creating new features (variables) from the initial dataset to improve the model's performance by making the data more informative and relevant. The choice of feature engineering techniques is guided by the context, so, for meteorological data used in agricultural applications are proposed two techniques, aggregation and standardization, as described below.

#### *4.3.3.1 Aggregation*

Data aggregation is the process of summarizing and condensing large sets of data into a more manageable and comprehensible form by combining or reducing the information within the dataset. Sometimes, raw data from meteorological observations may be too granular and aggregating it over time intervals, such as daily or monthly averages, can provide a more concise representation of the weather conditions while preserving important information. This typically involves performing mathematical operations, such as averaging, summing, counting, or finding the maximum/minimum values. Aggregation also serves to reduce data complexity and facilitate analysis by providing a higher-level view of the data, making it easier to discern patterns, trends, and key metrics within the information-rich dataset.

Generally, climate datasets have a wide range of measurements, with dozens of variables, measured in intervals of minutes or hours. This high frequency implies a large volume of data, which may not contribute to the training of mahine learning models. Considering the scope of agricultural applications, it may be viable and interesting to aggregate these data into average values for each day of the season being monitored, that is, identify average values of these variables for the first day of the season, second day of the season and so on. To do this, it is possible to use representative values of *"Day of the Year"* (*DOY*), and later aggregate them based on the average for each *DOY*. The definition of the size of the aggregation interval can vary according to the perspective of the analysis, as well as the form of aggregation, which can be the average value for variables such as temperature and relative humidity, or the daily accumulation for rain and solar radiation.

#### *4.3.3.2 Standardization*

In the context of climate data, the range of values for each variable is quite different. The temperature variable is expressed in degrees Celsius, while relative humidity is expressed in percentages, rainfall in inches, and solar radiation generally in watts per square meter. Furthermore, some variables have daily cyclical behavior, fluctuating throughout the day, but having a constant average, while cumulative variables can increase the measurement more significantly, or even others have the majority of values around zero. Based on these characteristics, the comparison or training of ML models based on differences between variables may be biased.

Standardization is the process of transforming data so that it has a standard or common scale. It involves rescaling the data, making it centered around a common reference point and ensuring that the units of measurement are consistent. This process is particularly useful when dealing with features that have different measurement scales, as it is the case of weather variables, since it allows fair comparisons between them, helping ML models converge faster and perform more effectively. In this framework, the *z-score* (DUBES; JAIN, 1980) technique is proposed, which transforms the variables so that they have a mean of 0 and a standard deviation of 1. This transformation is achieved by subtracting the mean of the data from each data point and then dividing by the standard deviation, effectively normalizing the data to a standardized distribution.

### 4.3.4 Clustering

The clustering procedure also involves smaller tasks, it is the step of the framework where the optimal number of clusters is sought, then the clustering is actually carried out and, finally, evaluated. These tasks are detailed here, starting with the methods for searching the number of clusters, as well as the clustering algorithms and metrics for evaluating the results.

#### *4.3.4.1 Find the Optimal Number of Clusters*

Find the optimal number of clusters is crucial in clustering analysis since it directly influences the interpretability of the results. Too few clusters may oversimplify the data, while too many clusters can make it difficult to extract meaningful insights. Clustering is often used as a pre-processing step for various data analysis tasks, such as climate zoning. The right number of clusters ensures that these subsequent tasks can be performed effectively, leading to more informed decision-making. Also, different numbers of clusters can lead to different levels of model performance because in large datasets clustering can be computationally expensive. In this framework, two well-known techniques are suggested for this task, the *Elbow Plot* and the *Sillhouete Score*, detailed below.

- **Elbow Plot:** This method operates by iteratively applying the clustering algorithm to the dataset with varying numbers of clusters, ranging from a small value to a relatively large one (CUI et al., 2020; THORNDIKE, 1953). For each number or clusters, the Within-Cluster Sum of Squares (WCSS) is calculated, which quantifies the squared distances between data points and their respective cluster centroids. WCSS gauges the compactness of clusters, and lower WCSS values indicate better clustering, yet it declines as the number of clusters increases. This information is then visualized on a line plot, with the number of clusters along the x-axis and the corresponding WCSS values along the y-axis. As the number of clusters grows, WCSS generally decreases, forming a declining curve. The key aspect of this method lies in identifying the *"elbow"* point on the curve. The elbow signifies the juncture at which the rate of decline in WCSS begins to slow down, creating a distinct inflection point resembling an elbow. This inflection point denotes the optimal number of clusters, as adding more clusters beyond this juncture fails to yield significant improvements in clustering quality. This optimal number of clusters represents a balance, allowing for the capture of meaningful data patterns while avoiding excessive cluster numbers that could lead to overfitting or increased complexity.

- **Silhouette Score:** This technique goes beyond just quantifying the number of clusters; it evaluates the quality of these clusters. By assessing the cohesion of data points within clusters and the separation between clusters, the silhouette score provides a more holistic understanding of the clustering's effectiveness (ROUSSEEUW, 1987). To employ this method, the clustering algorithm is applied to the dataset with varying cluster counts. For each configuration, the silhouette score for every data point is computed. These scores, ranging from -1 to 1, reflect the degree of fit within clusters and distinctiveness between clusters. Higher silhouette scores signify well-defined clusters, while lower scores indicate potential overlaps or less clear clustering. The crux of this approach lies in identifying the number of clusters that maximizes the average silhouette score across all data points. This optimal number of clusters is the one that results in the highest average silhouette score, signifying coherent and well-separated clusters.

#### *4.3.4.2 Cluster Algorithms*

Clustering algorithms play a important role in the analysis of weather data, allowing meteorologists, researchers, and decision-makers to uncover valuable insights and patterns within complex and multidimensional meteorological datasets. The choice of clustering algorithm depends on the specific goals of the analysis, the nature of the weather data, and the desired level of granularity in pattern recognition. Several clustering algorithms are commonly used in weather data analysis, and for this research were chosen three of the most populars: K-means, K-medoids, and Hierarchical Agglomerative Clustering, as detailed below.

- **K-means:** K-means is a popular unsupervised ML algorithm used for clustering data. It partitions a dataset into K clusters, where K is a user-defined number, by iteratively assigning data points to the cluster whose centroid (mean) is closest to them (AHMED; SERAJ; ISLAM, 2020; STEINLEY, 2006). Figure 10 illustrates the clustering process with K-means in four steps. The algorithm starts with initial random centroids (step 1) and refines them in an iterative (steps 2 and 3) process until convergence (step 4). It aims to minimize the sum of squared distances between data points and their assigned cluster centroids. K-means is computationally efficient and widely applied for various data clustering tasks, such as customer segmentation, image compression, and data analysis, where grouping similar data points into clusters is essential for data exploration and pattern discovery. The most common metrics for this type of calculation are *Euclidean Distance* (SERRÀ; ARCOS, 2014), *DTW (Dynamic Time Warping)* (SAKOE; CHIBA, 1978) and *Soft-DTW* (CUTURI; BLONDEL, 2017). In this framework, the metric chosen for cluster assignment was Euclidean Distance, well recommended for time series of the same size and for having better performance in test runs and model calibration. Euclidean distance is a measure of the straight-line distance between two points in Euclidean space (ABANDA; MORI; LOZANO, 2019). In simpler terms, it's the length of the shortest path between two points in a plane. Figure 11 shows an example, with two time series T (red) and S (blue), where the Euclidean distance is the sum of the point-to-point distances (gray lines), along all the time series. Finally, a random number was used as a seed and 10 initializations were made for each run.

- **K-medoids:** K-medoids is another unsupervised clustering algorithm used to partition data into K clusters, but it differs from K-means in that it focuses on

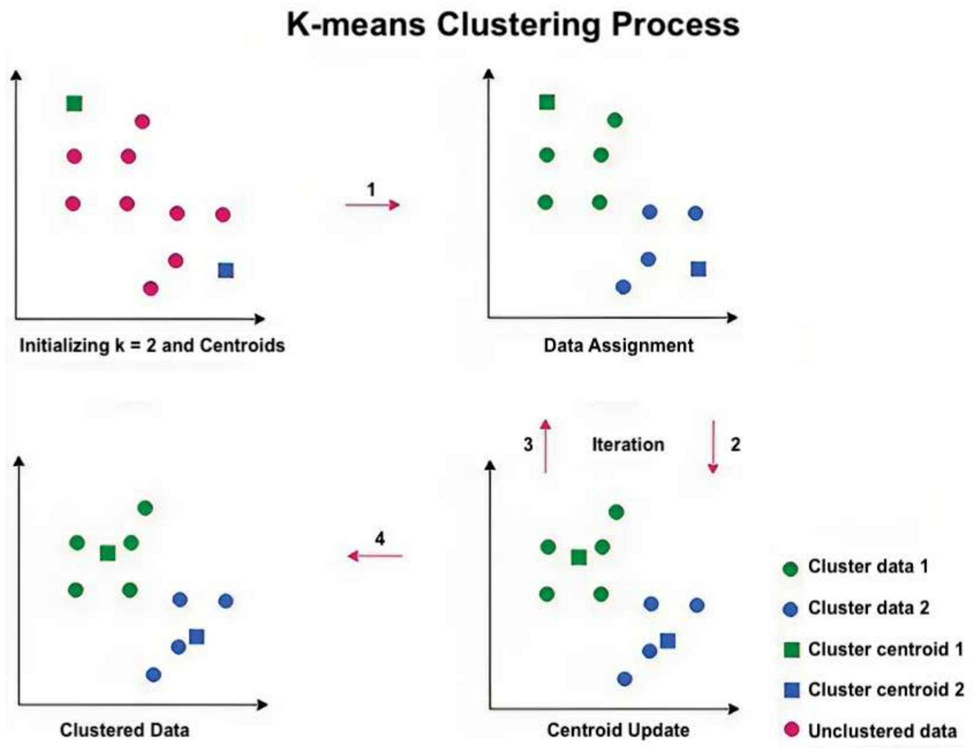


Figure 10 – K-means Clustering Process.

Image source: https://www.e-consystems.com/blog/camera/technology/what-is-k-means-clustering-algorithm/

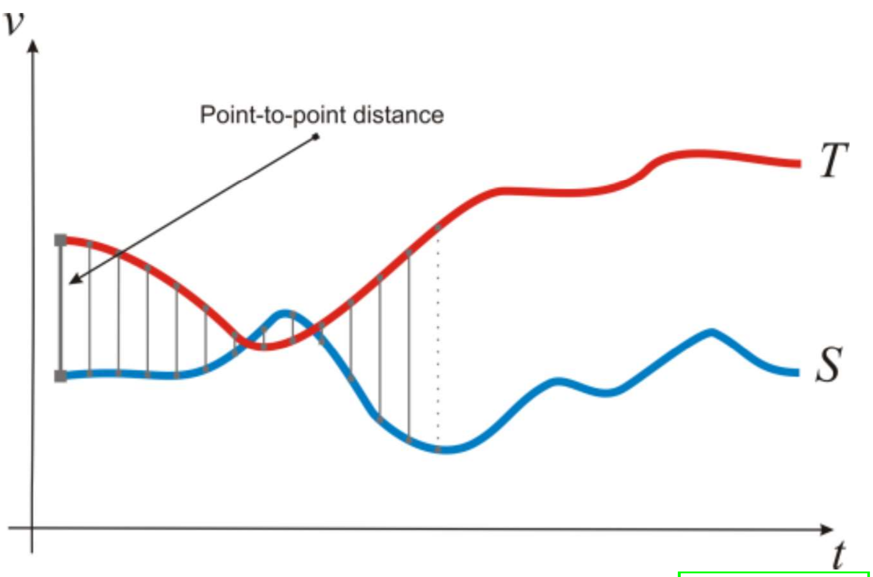


Figure 11 – Euclidean Distance between two time series T and S. (CASSISI et al., 2012)

finding representative data points (medoids) as cluster centers rather than the mean (RDUSSEEUN; KAUFMAN, 1987; KAUFMAN; ROUSSEEUW, 1990). Figure 12 illustrates the clustering process using the K-medoids algorithm, also in four steps. It begins by selecting K initial data points as medoids (step 1) and iteratively optimizes the medoids to minimize the total dissimilarity between data points and their nearest medoid within each cluster (steps 2 and 3), until convergence (step 4). K-medoids is robust to outliers and works well with a wide range of dissimilarity measures, making it suitable for applications where choosing an actual data point as a cluster center is more appropriate than using the mean. In this research, the standard parameters of the algorithm were used, also with *Euclidean Distance* as time series distance metric, under the same previous motivation.

- **Hierarchical Agglomerative Clustering:** Hierarchical Agglomerative Clustering (HAC) is a hierarchical clustering algorithm used in unsupervised learning to build a tree-like structure (dendrogram) of data points or clusters (MURTAGH; LEGENDRE, 2014; LUKASOVÁ, 1979). It starts by treating each data point as an individual cluster and then repeatedly merges the closest clusters into larger clusters, continuing this process until all data points belong to a single cluster at the root of the dendrogram. HAC does not require the user to specify the number of clusters in advance, making it a versatile method for exploring hierarchical relationships within data. It is widely employed in various fields, such as biology, social sciences, and document retrieval, to reveal nested structures and relationships in datasets by forming a hierarchy of clusters based on similarity or dissimilarity metrics. An important parameter in hierarchical clustering is the linkage criterion between clusters. There are different linkage methods, such as *Single Linkage (or Nearest Neighbor)* (SIBSON, 1973), *Complete Linkage (or Furthest Neighbor)* (DEFAYS, 1977), *Average Linkage* (SOKAL; MICHENER, 1958), *Ward's Method* (WARD, 1963), and *Centroid Linkage* (SCHÜTZE; MANNING; RAGHAVAN, 2008). In this framework, the *Ward's* variance minimization metric, illustrated in Figure 13, was used. This method aims to merge clusters in a way that minimally increases the total within-cluster Sum of Squares Error (ESS) and tends to produce more compact and spherical clusters compared to other linkage criteria. Ward's method is less sensitive to the shape of clusters compared to single-linkage or complete-linkage, making it more robust in cases where clusters have varying shapes. Also, this method often produces well-balanced hierarchical structures in the dendrogram, making it easier to interpret the relationships between clusters at different levels of the hierarchy and can be less influenced by noise and outliers compared to some other methods.

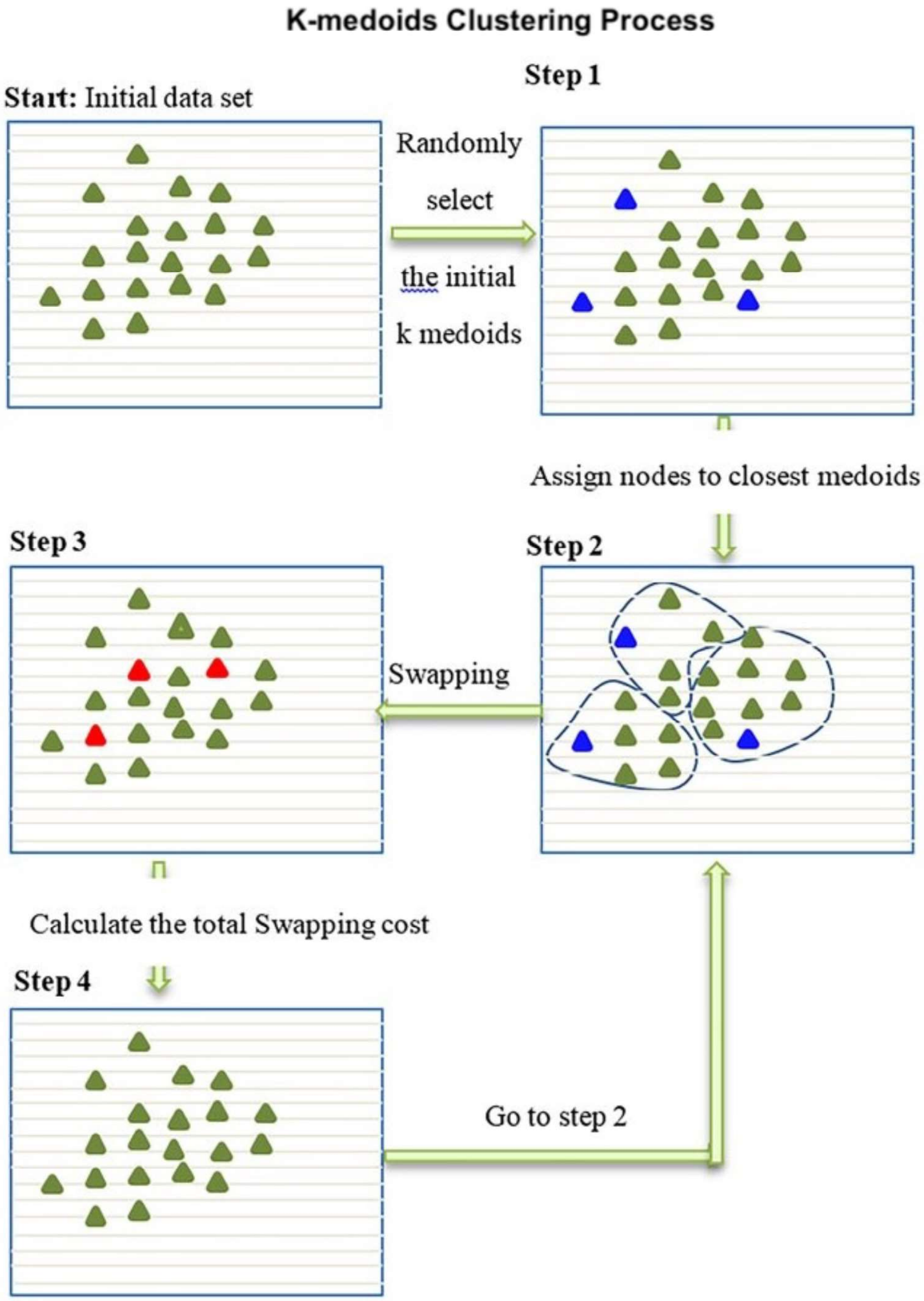

K-medoids Clustering Process
Start: Initial data set
Step 1
Randomly select the initial k medoids
Assign nodes to closest medoids
Step 2
Step 3
Swapping
Calculate the total Swapping cost
Step 4
Go to step 2

Figure 12 – K-medoids Clustering Process. (HAJLAOUI et al., 2019)

### *4.3.4.3 Cluster Evaluation*

To evaluate clustering, there are many generic statistical metrics, which are applicable to a large number of problems, but without specializing the evaluation based on context aspects. Internal metrics, like the Silhouette, gauge the clustering quality by examining the characteristics of the data and the resultant clustering structure. In the current context, where silhouette scores have already been computed, it becomes evident that despite selecting the optimal number of clusters, the overall quality of the clusters appears to be moderate. This observation arises from the fact that the sil-

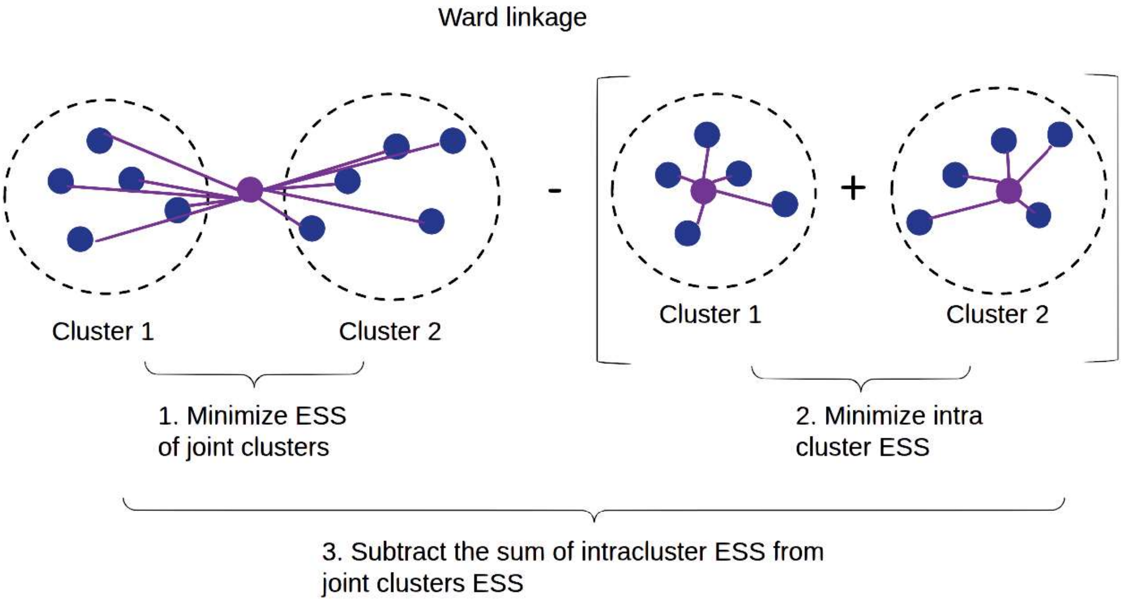

Ward linkage
Cluster 1
Cluster 2
-
+
Cluster 1
Cluster 2
1. Minimize ESS of joint clusters
2. Minimize intra cluster ESS
3. Subtract the sum of intracluster ESS from joint clusters ESS

Figure 13 – Ward's Linkage Method. (WARD, 1963)

Image source: https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/

houette scores tend to be closer to 0, indicating an average quality of clustering, as opposed to the ideal score of 1, which would represent the best possible clustering solution.

Therefore, in this research, within the context of climate data, it was decided to implement a complementary and more specific evaluation metric, in order to quantify the similarity of elements in the same cluster, using supervised ML and based on the *"Leave-one-out"* validation approach. In this method, each data point in the dataset is temporarily removed from the cluster and the other elements in that cluster are used to predict the removed one, assessing the error between real and predicted values. Thus, from cluster $C$, composed of stations $s$ ($s_1$, ..., $s_n$), a station $s_i$ is removed and a Linear Regression model is trained to predict the values of $s_i$ using the remaining stations in cluster $C$. Then, the *RMSE (Root Mean Squared Error)* of the prediction is calculated, comparing the real data with the predicted data. The prediction is then made for all stations, through this process of removing a station from the cluster, predicting it and adding it back to the cluster, and after that, each station is assigned an RMSE. The prediction is then made for all stations, through this process of removing a station from the cluster, predicting it and adding it back to the cluster, and after that, each station is assigned an RMSE (WONG, 2015). Then, an average RMSE is calculated for cluster $C$, based on the RMSEs of stations $s_1$, ..., $s_n$. With the RMSEs of each cluster, the average RMSE of that entire cluster is calculated.

The best performance expected here is from the algorithm that presents a lower Total RMSE, as it indicates greater similarity between the stations in these clusters. The cluster evaluation results of this study are presented in the following chapter.

## 4.4 Chapter Remarks

This chapter presented the proposed methodology for clustering meteorological time series for agricultural applications, in the format of a framework called **FMTSClust**. The details bring aspects of conception, design and execution flow, presenting step by step the operation of the framework. The goal is to guide and automate the clustering, specifically focused on agricultural aspects. The statistical and ML techniques were also detailed, as well as the evaluation metrics used in each phase.

# 5 FRAMEWORK IMPLEMENTATION

This chapter reports on the implementation of codes related to the automation of the framework procedures. The development was done in two versions. Version I was implemented in the form of an auxiliary python library with functions for MTS clustering. Version II, being an evolution of the first, was implemented in the form of an API. The complete source code of this research, consisting of the implemented library, notebooks with case studies and the API is available at the project public repository: `https://github.com/marcosjr06/FMTSClust`.

## 5.1 Development Environment

The development environment used was Jupyter Lab[1], a web-based interactive development environment, within the Anaconda[2] platform. The implementation was done in the Python language, version `3.11.0`, packaged by `conda-forge`, using Jupyter notebooks (KLUYVER et al., 2016). Table 5 shows a list of used auxiliary libraries, versions, and their respective home pages. The use of each one in different stages of implementation is detailed below.

Table 5 – Dependencies and versions of all Python libraries[a] used to instantiate the framework.

| **Library** | **Version** | **Home page** |
|---|---|---|
| `pandas` | `2.1.2` | `http://pandas.pydata.org` |
| `glob2` | `0.7` | `http://github.com/miracle2k/python-glob2` |
| `numpy` | `1.23.5` | `http://www.numpy.org` |
| `scipy` | `1.10.1` | `http://www.scipy.org` |
| `scikit-learn` | `1.2.2` | `http://scikit-learn.org` |
| `scikit-learn-extra` | `0.3.0` | `http://github.com/scikit-learn-contrib` |
| `tslearn` | `0.6.1` | `http://tslearn.readthedocs.io` |
| `fastdtw` | `0.3.4` | `http://pypi.org/project/fastdtw` |
| `matplotlib` | `3.8.0` | `http://matplotlib.org` |

[a] All links accessed in November, 16 2023.

[1] JupyterLab - `https://jupyter.org/`
[2] Anaconda - `http://www.anaconda.com/`

## 5.2 Version I - *FMTSClust* Library

A library is a collection of pre-written code or modules that can be imported and used in other programs. Libraries provide reusable functions and routines that help simplify the development process by abstracting complex tasks into manageable components (STANČIN; JOVIĆ, 2019). Within the scope of this research, the library format was chosen to implement the first version, especially because it provides code reusability, allowing developers to leverage existing solutions rather than starting from scratch and functionality extension, providing additional capabilities to existing libraries.

Figure 14 shows the sequence and flow of the framework, with the main functions implemented for each step and respective outputs. The implementation of the first version of the framework was done in a single file named `FMTSClust.py`. This file concentrates the implementation of auxiliary functions for data import, preprocessing, feature extraction, clustering and validation. This file has more implemented functions, to be used in used in intermediate stages of the framework. The implementation was basically documented using *docstrings*, to enable the use of the library by other researchers. Also, it is expected to encourage collaboration and knowledge sharing. Other developers can contribute to improving and expanding the library, reporting problems or suggesting features. For textual convenience, only the function definitions and docstrings will be presented here, and the complete codes are available in the project repository.

### 5.2.1 Data Extraction Functions

The first group of functions to be introduced is *Data Extraction*, as shown in Listing 1, referring to the first stage of the framework: `getFAWNdata` and `getSIMAGROdata`, as specializations of traditional *getData* function. These functions are used to load data from the FAWN and SIMAGRO-RS databases, respectively. As a parameter they receive the local path to the folder containing the `.csv` files for each database. As the databases have `.csv` files with different headers, each one needs to be imported in a different way. Both functions then load the data, calculate the *DOY*, as previously introduced, and exclude the leap day from leap years.

### 5.2.2 Data Preprocessing Functions

The second group of functions is dedicated to the data preprocessing stage, the second of the framework, containing the functions: `filterByDate`, `check_data_availability_stations`, `QC_missing_null_values`, and `QC_outliers`. The parameters and returns of each are detailed in Listing 2. The first applies a date filter according to the agricultural season to be analyzed (e.g. crop production period, disease control period), in order to eliminate data from undesirable periods. The sec-

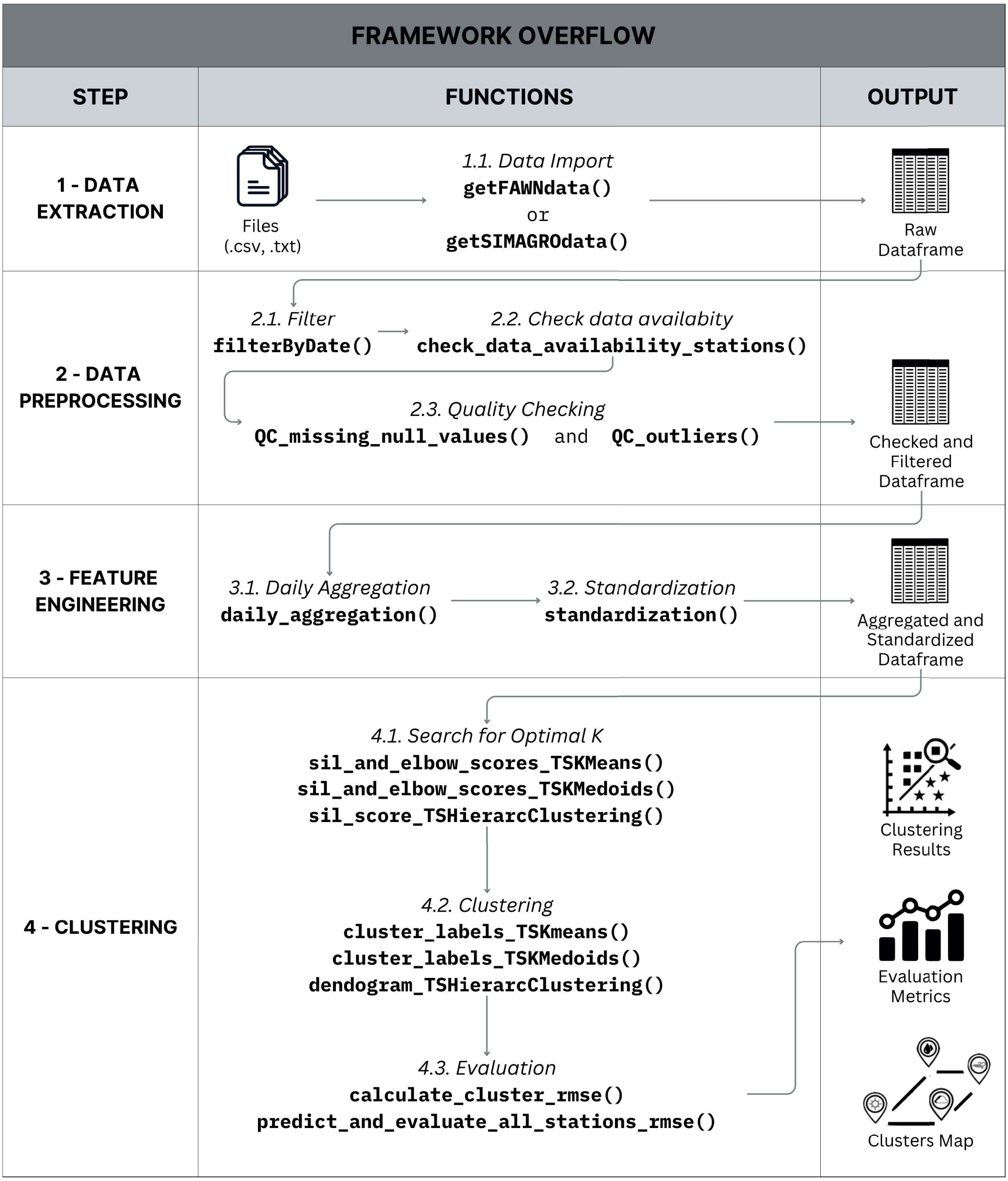


Figure 14 – FMTSClust main functions implemented for each step and outputs.

ond checks the existence of data for all meteorological stations in that database, based on a minimum number of years that the station must have data collected to be considered in the analysis. Finally, the two other functions perform the quality checking stage, one referring to missing and null data, and the other detecting and capping outliers.

```
# FMTSClust - Step 1 - DATA EXTRACTION FUNCTIONS

def getFAWNdata(path):
    """
    Imports FAWN data from .csv files into pandas dataset.

    Parameters:
    - path (str): path to folder with FAWN files.

    Returns:
    - dataframe: FAWN dataset (variables T, RH, TF and RF),
    with DOY and leap days excluded.
    """


def getSIMAGROdata(path):
    """
    Imports SIMAGRO-RS data from .csv files into pandas dataset.

    Parameters:
    - path (str): path to folder with SIMAGRO-RS files.

    Returns:
    - dataframe: SIMAGRO-RS dataset (variables T, RH, TF and RF),
    with DOY and leap days excluded.
    """

```

Listing 1 – Data Extraction functions definitions.

### 5.2.3 Feature Engineering Functions

The third group of implemented functions is aimed at the feature engineering phase. Listing 3 presents the definition, as well as the parameters and returns of each one. The first, `daily_aggregation`, aggregates the values of the meteorological variables, generally collected on an hourly basis, into daily average values. The second, `standardization` uses the z-score technique, introduced previously, to standardize the values of the dataset, in order to seek a representation of the variables on the same scale.

### 5.2.4 Clustering Functions

The functions implemented for the fourth step of the framework, clustering, are focused on the tasks of searching for the optimal number of clusters and clustering, producing, at the end of each algorithm execution, a set of cluster labels for each

```python
# FMTSClust - Step 2 - DATA PREPROCESSING FUNCTIONS

def filterByDate(df, yearStart, yearEnd, monthDayStart, monthDayEnd):
    """
    Filters a dataset according to a time window for analysis (season).

    Parameters:
    - df (dataframe): Dataset.
    - yearStart (str): Year of start of the filter ("YYYY").
    - yearEnd (str): Year of end of the filter ("YYYY").
    - monthDayStart (str): Start month/day of the time window ("MM-DD").
    - monthDayEnd (str): End month/day of the time window ("MM-DD").

    Returns:
    - dataframe: Filtered dataset.
    """

def check_data_availability_stations(df, nYears):
    """
    Check and filters stations that have data for at least 'n' years.

    Parameters:
    - df (dataframe): Dataset.
    - nYears (int): Minimum number of years for analysis.

    Returns:
    - dataframe: Dataset with data only from the last 'n' years.
    - ids_filtered: List of station IDs with data in the last 'minYears'.
    """

def QC_missing_null_values(df,dfName):
    """
    Calculates and prints missing values.

    Parameters:
    - df (dataframe): Dataset.
    - dfName (str): Database name ("fawn" or "simagro").
    """

def QC_outliers(df):
    """
    Identifies, prints and caps outliers using IQR method.

    Parameters:
    - df (dataframe): Dataset.
    """

```

Listing 2 – Data Preprocessing functions definitions.

```python
# FMTSClust - Step 3 - FEATURE ENGINEERING FUNCTIONS

def daily_aggregation(df,agg_columns):
    """
    Aggregates data into representative daily values.

    Parameters:
    - df (dataframe): Dataset.
    - agg_columns (list): List of columns to be aggregated.

    Returns:
    - dataframe: Daily aggregated dataset.
    """

def standardization(df, std_columns):
    """
    Standardizes data using the z-score technique.

    Parameters:
    - df (dataframe): Dataset.
    - std_columns (list): List of columns to be standardized.

    Returns:
    - dataframe: Standardized dataset.
    """

```

Listing 3 – Feature Engineering functions definitions.

meteorological station in the dataset. Listing 4 presents, first, the functions for the K-means algorithm: `sil_and_elbow_scores_TSKmeans` and `cluster_labels_TSKmeans`. The first calculates the Silhoutte Score and generates the Elbow Plot for all variations in the number of clusters, up to a maximum number entered as a parameter, using K-means algorithm. The second executes the K-means algorithm with a specific number of clusters informed as a parameter and returns the set of cluster labels.

With other algorithms, the nomenclature and definitions of functions are similar. For the K-medoids algorithm, the functions are `sil_and_elbow_scores_TSKmedoids` and `cluster_labels_TSKmedoids`, as also shown in Listing 4. Again, the first calculates the Silhouette Score and the Elbow Plot and the second the cluster labels, but then using K-medoids algorithm.

Similarly, the functions presented in Listing 5 were implemented for the Hierarchical Agglomerative Clustering algorithm. The `sil_scores_TSKmedoids` function calculates the Silhouette Scores, from one to the maximum number of clusters, which in this case is the number of existing meteorological stations. Function

```
# FMTSClust - Step 4 - CLUSTERING - K-means

def sil_and_elbow_scores_TSKMeans(df_array, max_cluster,
                                  n_init_c, max_iter_c):
    """
    Calculates and prints the Silhouette Score and the
    Elbow Plot using K-means algorithm.

    Parameters:
    - df_array (ndarray): Dataset in 3D-ndarray format (number
    of stations, time series size, number of variables).
    - max_cluster (int): Maximum number of clusters.
    - n_init_c (int): Initializations with different centroid seeds.
    - max_iter_c (int): Iterations for a single run.
    """

def cluster_labels_TSKmeans(df_array, n_clusters_c, n_init_c,
                            max_iter_c):
    """
    Runs K-means algorithm with specific number of clusters.

    Parameters:
    - df_array (ndarray): Dataset in 3D-ndarray format (number
    of stations, time series size, number of variables).
    - n_clusters_c (int): Number of clusters.
    - n_init_c (int): Initializations with different centroid seeds.
    - max_iter_c (int): Iterations for a single run.
    Returns:
    - list: Cluster labels for each meteorological station.
    """

# FMTSClust - Step 4 - CLUSTERING - K-medoids

def sil_and_elbow_scores_TSKMedoids(df_array, max_cluster, max_iter_c):
    """
    Calculates and prints the Silhouette Score and the Elbow Plot
    using K-medoids algorithm.

    Parameters:
    - df_array (ndarray): Dataset in 3D-ndarray format (number
    of stations, time series size, number of variables).
    - max_cluster (int): Maximum number of clusters.
    - max_iter_c (int): Iterations for a single run.
    """

def cluster_labels_TSKMedoids(df_array, n_clusters_c, max_iter_c):
    """
    Runs K-medoids algorithm with specific number of clusters.

    Parameters:
    - df_array (ndarray): Dataset in 3D-ndarray format (number
    of stations, time series size, number of variables).
    - n_clusters_c (int): Number of clusters.
    - max_iter_c (int): Iterations for a single run.

    Returns:
    - list: Cluster labels for each meteorological station.
    """
```

Listing 4 – K-means and K-medoids Clustering functions definitions.

```
# FMTSClust - Step 4 - CLUSTERING - Hierarchical Agglomerative Clustering


def sil_score_TSHierarcClustering(df_array):
    """
    Calculates and prints the Silhouette Score and the Elbow Plot
    using Hierarchical Agglomerative Clustering algorithm.

    Parameters:
    - df_array (ndarray): Dataset in 3D-ndarray format (number
    of stations, time series size, number of variables).
    """


def dendogram_TSHierarcClustering(df_array, num_clusters,
                                  list_names_dendogram):
    """
    Runs Hierarchical Agglomerative Clustering algorithm
    with specific number of clusters.

    Parameters:
    - df_array (ndarray): Dataset in 3D-ndarray format (number
    of stations, time series size, number of variables).
    - n_clusters_c (int): Number of clusters.
    - list_names_dendogram (list): Meteorological stations names.

    Returns:
    - list: Cluster labels for each meteorological station and
    linkage matrix.
    """

```

Listing 5 – Hierarchical Agglomerative Clustering functions definitions.

`dendogram_TSHierarcClustering` performs the clustering, generating the complete dendrogram, and returning the cluster labels according to the number of clusters entered as a parameter. For the hierarchical algorithm, elbow plot is not calculated, as the algorithm approach is different.

Finally, for this fourth stage of the framework, two functions were implemented to evaluate the clustering results, based on the supervised ML technique of linear regression and the "leave-one-out" approach, as explained in the previous section. Listing 6 presents the definitions of the functions implemented to evaluate clustering. Function `calculate_cluster_rmse` calculates the average RMSE of the entire clustering prediction, for all clusters and stations. This functions uses the other function, called `predict_and_evaluate_all_stations_rmse`. The first function uses this second func-

```python
# FMTSClust - Step 4 - CLUSTERING - EVALUATION


def calculate_cluster_rmse(data_df, cluster_labels):
    """
    Calculate the average RMSE for each cluster.

    Parameters:
    - data_df (dataframe): Dataset.
    - cluster_labels (list): Clustering labels result.

    Returns:
    - avg_rmse: Clustering total average prediction RMSE.
    """


def predict_and_evaluate_all_stations_rmse(data_df):
    """
    Predicts the variables of a station based on other
    stations in the same cluster (leave-one-out approach).

    Parameters:
    - data_df (dataframe): Dataset.

    Returns:
    - prediction_RMSE: Predicition RMSE within the cluster.
    """

```

Listing 6 – Clustering Evaluation functions definitions.

tion, in cyclic iterations, to predict and calculate the RMSE for each station in a cluster. These two functions then provide the main clustering evaluation metric, printing the RMSE results for all stations, all clusters and for the entire clustering of an algorithm. This is the main metric used to evaluate the results of clustering.

### 5.2.5 Extra Functions

Additionally, other functions were implemented to assist with mid-process tasks and visualization of clustering results. The definitions of these functions are described in Listing 7. The first five are for “Time Series Visualization,” varying the way of calling, by index, by station name and all variables on separate axes and on the same axis. Then, the other five functions are for “Map Visualization” and “Clustering Results Visualization,” to visualize meteorological stations distributed according to geolocations, as well as the result of clustering, coloring stations from the same clusters. Finally,

```
# FMTSClust - Extra Functions


# Time Series Visualization
def plot_first_TS(array, station_list):

def plot_all_TS(array, station_list):

def plot_TS_by_index(array, station_list, index):

def plot_TS_by_index_title(array, station_list, index, title):

def plot_TS_by_index_separated_variables(array, station_list, index):


# Map Visualization
def plot_map_stations(stations):

def plot_map_stations_IDs(stations):


# Clustering Results Visualization
def plot_map_clustering(stations, cluster_labels):

def plot_map_clustering_names(stations, cluster_labels, title):

def plot_map_clustering_IDs(stations, cluster_labels, title):


# Data Transformation
def transform_dfTS_to_3Darray_DOYsorted(df, columns, doy_lim1, doy_lim2):

def flatten_multivariate_time_series(data):

```

Listing 7 – Extra functions definitions.

there are also two auxiliary functions for “Data Transformation.” The first is used prior to clustering, just to transform the dataset into the format of a three-dimensional array, for computing purposes, while the second, flattens a three-dimensional array to a two-dimensional pandas dataframe, used before the clustering evaluation functions, both for computing reasons.

## 5.3 Version II - FMTSClustAPI

The second version of the framework is implemented in the format of an Application Programming Interface (API) (BIEHL, 2015). An API is a set of rules and tools that allows different software applications to communicate with each other. It defines the methods and data formats that applications can use to request and exchange information. APIs are used to enable the integration of different software systems, allowing them to work together and share data seamlessly. Commonly, they are applied in web development, allowing web applications to interact with external services, databases, or other web applications (MANIKAS, 2016). For these reasons, the API format was chosen to implement the second version of FMTSClust, in order to enable access to its functions more quickly and easily.

APIs can be categorized based on different criteria, one of the main ones being the integration approach, *RESTFull* or *SOAP*. REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) represent two distinct approaches to designing web services with notable differences. REST (MASSE, 2011), an architectural style, emphasizes simplicity and statelessness, utilizing standard HTTP methods for operations on resources and employing lightweight data formats like JSON for communication. It is flexible, scalable, and well-suited for scenarios prioritizing speed and ease of integration. On the other hand, SOAP (MUELLER, 2002) is a protocol defining a strict set of standards for message structure, often using XML. SOAP offers a more formalized contract between client and server, supporting various transport protocols beyond HTTP and providing features like transactions and security. While REST is chosen for its simplicity and compatibility with existing web infrastructure, SOAP is preferred in scenarios requiring a more rigid structure, extensive functionality, and adherence to established standards. Therefore, in the context of the framework proposed in this work, the REST approach was understood as more appropriate to implement it.

RESTFul is an adjective derived from REST and is used to describe an implementation that adheres to the principles of REST. This approach emphasizes a stateless client-server interaction, where each request from a client contains all the information needed to understand and process the request. It leverages standard HTTP methods (`GET, POST, PUT, DELETE`) for operations on resources, and the resources themselves are identified by URIs (Uniform Resource Identifiers). Data is typically exchanged in formats like JSON or XML. For this type of API, there are web frameworks for implementation, which speed up the work, such as *FastAPI*.

### 5.3.1 FastAPI

FastAPI[3] is a modern and high-performance web framework for building APIs with Python 3.7 and above (LATHKAR, 2023). It is designed to be easy to use, fast, and to leverage the advantages of Python type hints, combining the simplicity of frameworks like Flask with the performance benefits of asynchronous programming. One of its standout features is automatic generation of OpenAPI and JSON Schema documentation, making it effortless for developers to understand, test, and interact with their APIs. FastAPI is built on top of Starlette and Pydantic, utilizing the latest advancements in Python to provide features like dependency injection, data validation, automatic serialization, and asynchronous support (PERALTA, 2023). Its seamless integration with tools like Uvicorn and automatic validation based on type hints make it a compelling choice for building robust and efficient APIs.

The implementation of the API was also carried out in the Jupyter Lab environment, integrating FastAPI and using the Uvicorn [4] server. UVicorn is an ASGI (Asynchronous Server Gateway Interface) server implementation specifically designed for running asynchronous web applications in Python. It is built on top of the high-performance UVloop and HTTP protocols. The primary purpose of UVicorn is to serve applications that leverage asynchronous programming, providing better performance and scalability. In the context of FastAPI, UVicorn is commonly used as the server to run FastAPI applications. UVicorn and FastAPI work together to provide a high-performance, asynchronous web framework for building APIs in Python.

When running the server and the API, Uvicorn mounts the FastAPI application at a local address (Ex.: `http://127.0.0.1:8000/`), and the endpoints (functions) are accessible from a browser or for requests, adding the url with the endpoint name. The documentation, automatically generated by FastAPI, is available, for example, at `http://127.0.0.1:8000/docs`.

### 5.3.2 Endpoints

In the context of web development and APIs, an endpoint refers to a specific URL or URI (Uniform Resource Identifier) that an API exposes for performing certain operations. Endpoints represent the various functions or resources that clients can interact with through HTTP requests. Each endpoint typically corresponds to a specific functionality or action that the API can perform. The choice of FastAPI was to speed up the API design and also reuse the python code from the first version, refactoring the functions of the first version into API endpoints. Since the functioning and purpose of an API is different from a python library, the functions implemented in `FMTSClust.py` are not necessarily endpoints of the `FMTSClustAPI`.

[3] FastAPI - `https://fastapi.tiangolo.com/`
[4] Uvicorn - `https://www.uvicorn.org/`

The main difference is that in the API format the return of the endpoints, that is, the output of each function, is in *JSON*[5], as a request response. JSON, or JavaScript Object Notation, is language-agnostic format, meaning it can be used across various programming languages, fostering interoperability between different systems and technologies. As a web standard, JSON has become the preferred data format for web APIs, where it facilitates data exchange between clients and servers in web applications. Its ease of parsing and serialization in most programming languages, coupled with a wide range of available libraries.

Then, a FastAPI application was created, by defining an instance of the FastAPI class and some endpoints were defined, by utilizing FastAPI decorators such as `@app.get` or `@app.post`. These decorators allow to map the function to a specific HTTP method and endpoint path. Parameters and query parameters for the functions can be seamlessly integrated by specifying them in the endpoint functions, as FastAPI automatically handles the conversion of query parameters, path parameters, and request bodies into function arguments. Once the endpoints were defined, the FastAPI application could run on the web server UVicorn, enabling interaction with the API for testing and documentation purposes. The dependencies are the same as the first version, plus the `fastapi (0.104.1)` and `uvicorn (0.24.0.post1)` packages. As before, for textual convenience, here only the definition of the endpoints will be described.

Regarding endpoints, six were created, two for each algorithm, referring to the functions for search the optimal number of clusters and, then, clustering MTS. Listing 8 presents the endpoints of FMTSClustAPI for K-means algorithm: `sil_elbow_scores_TSKMeans` and `cluster_TSKmeans`. As detailed in the comment strings, below the definition of the endpoints, both receive the name of the database as an argument, and the second also receives the number of clusters to run the clustering. The return of both is in JSON format, with the first returning a dictionary of all Silhouette Scores and WCSS, according to the cluster variation, while the second returns a dictionary with the clustering result and the total RMSE of the clustering, used for clustering evaluation.

Similarly, Listing 9 present the endpoints for the K-medoids and Hierarchical Clustering algorithms. For each algorithm, two equivalent endpoints were created with similar names for all, only the ending varying:: `sil_elbow_scores_TSKMedoids` and `cluster_TSKmedoids`, for K-medoids; and `sil_score_TSHClustering` and `cluster_TSHClustering`, for Hierarchical Agglomerative Clustering. The operation, parameters and returns of these endpoints are the same as the two K-means, explained previously, remembering that for hierarchical clustering, the elbow plot is not computed. The endpoints differs basically in the algorithm used for clustering

As mentioned earlier, FastAPI automatically generates a documentation for the API,

[5] JSON - `https://www.json.org`

```python
# FMTSClustAPI - ENDPOINTS - K-MEANS

@app.get('/sil_elbow_scores_TSKMeans')
async def sil_elbow_scores_TSKMeans(dfName):
    """
    Calculates the Silhouette Score and the
    Elbow Plot WCSS using K-means algorithm.

    Parameters:
    - dfName (str): Database name ("fawn" or "simagro").

    Returns:
    - dict: json object with Silhouette and WCSS scores.
    """



@app.get('/cluster_TSKmeans')
async def cluster_TSKmeans(dfName,nClusters):
    """
    Runs K-means algorithm with specific number of clusters.

    Parameters:
    - dfName (str): Database name ("fawn" or "simagro").
    - nClusters (int): Number of clusters.

    Returns:
    - dict: json object with clustering results and RMSE.
    """

```

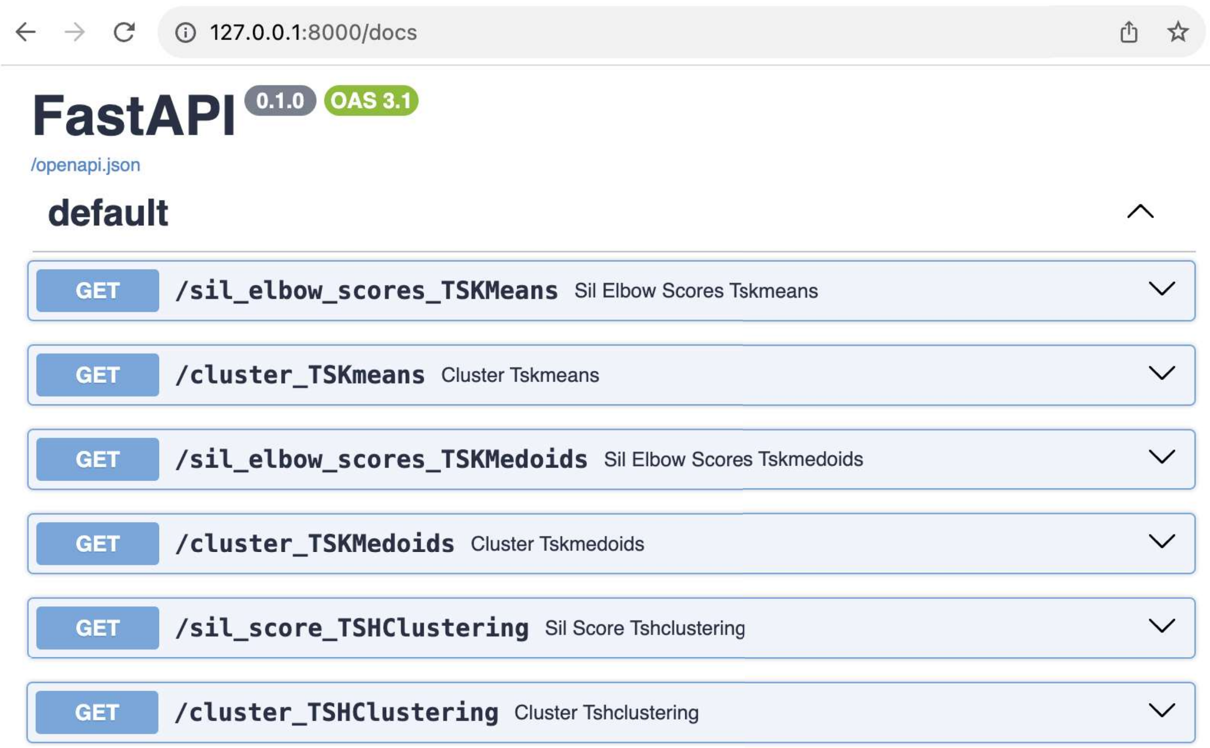
Listing 8 – FMTSClustAPI endpoints - K-means algorithm.

available in the `docs` endpoint (`http://localhost:8000/docs`). Figure 15 shows this documentation home page, generated for FMTSClustAPI. This documentation interface includes information about all your API endpoints, request and response models, when clicking on each endpoint, and even allows you to interact with your API directly from the documentation. This automatic generation of documentation is possible because FastAPI leverages Python's type hints and function annotations. It uses this information to generate the OpenAPI schema, which is then used to create the documentation. This feature simplifies the process of documenting your API and also helps ensure that the documentation stays up-to-date when modifying API endpoints.

The way to interact with the FMTSClustAPI, then, is through requests to endpoints. For testing and demonstration, a python notebook was implemented (`FMTSClustAPIDemo.ipynb`), also available in the project repository, with example re-

```
# FMTSClustAPI - ENDPOINTS - K-MEDOIDS

@app.get('/sil_elbow_scores_TSKMedoids')
def sil_elbow_scores_TSKMedoids(dfName):
    """
    Calculates the Silhouette Score and the
    Elbow Plot WCSS using K-medoids algorithm.

    Parameters:
    - dfName (str): Database name ("fawn" or "simagro").

    Returns:
    - dict: json object with Silhouette and WCSS scores.
    """

@app.get('/cluster_TSKMedoids')
def cluster_TSKMedoids(dfName,nClusters):
    """
    Runs K-medoids algorithm with specific number of clusters.

    Parameters:
    - dfName (str): Database name ("fawn" or "simagro").
    - nClusters (int): Number of clusters.

    Returns:
    - dict: json object with clustering results and RMSE.
    """


# FMTSClustAPI - ENDPOINTS - HIERARCHICAL CLUSTERING

@app.get('/sil_score_TSHClustering')
def sil_score_TSHClustering(dfName):
    """
    Calculates the Silhouette Score using
    Hierarchical Clustering algorithm.

    Parameters:
    - dfName (str): Database name ("fawn" or "simagro").

    Returns:
    - dict: json object with Silhouette and WCSS scores.
    """

@app.get('/cluster_TSHClustering')
def cluster_TSHClustering(dfName,nClusters):
    """
    Runs Hierarchical Clustering algorithm with specific number of clusters.

    Parameters:
    - dfName (str): Database name ("fawn" or "simagro").
    - nClusters (int): Number of clusters.

    Returns:
    - dict: json object with clustering results and RMSE.
    """
```

Listing 9 – FMTSClustAPI endpoints - K-medoids and Hierarchical Clustering algorithms.



127.0.0.1:8000/docs

FastAPI 0.1.0 OAS 3.1

/openapi.json

default

GET /sil_elbow_scores_TSKMeans Sil Elbow Scores Tskmeans

GET /cluster_TSKmeans Cluster Tskmeans

GET /sil_elbow_scores_TSKMedoids Sil Elbow Scores Tskmedoids

GET /cluster_TSKMedoids Cluster Tskmedoids

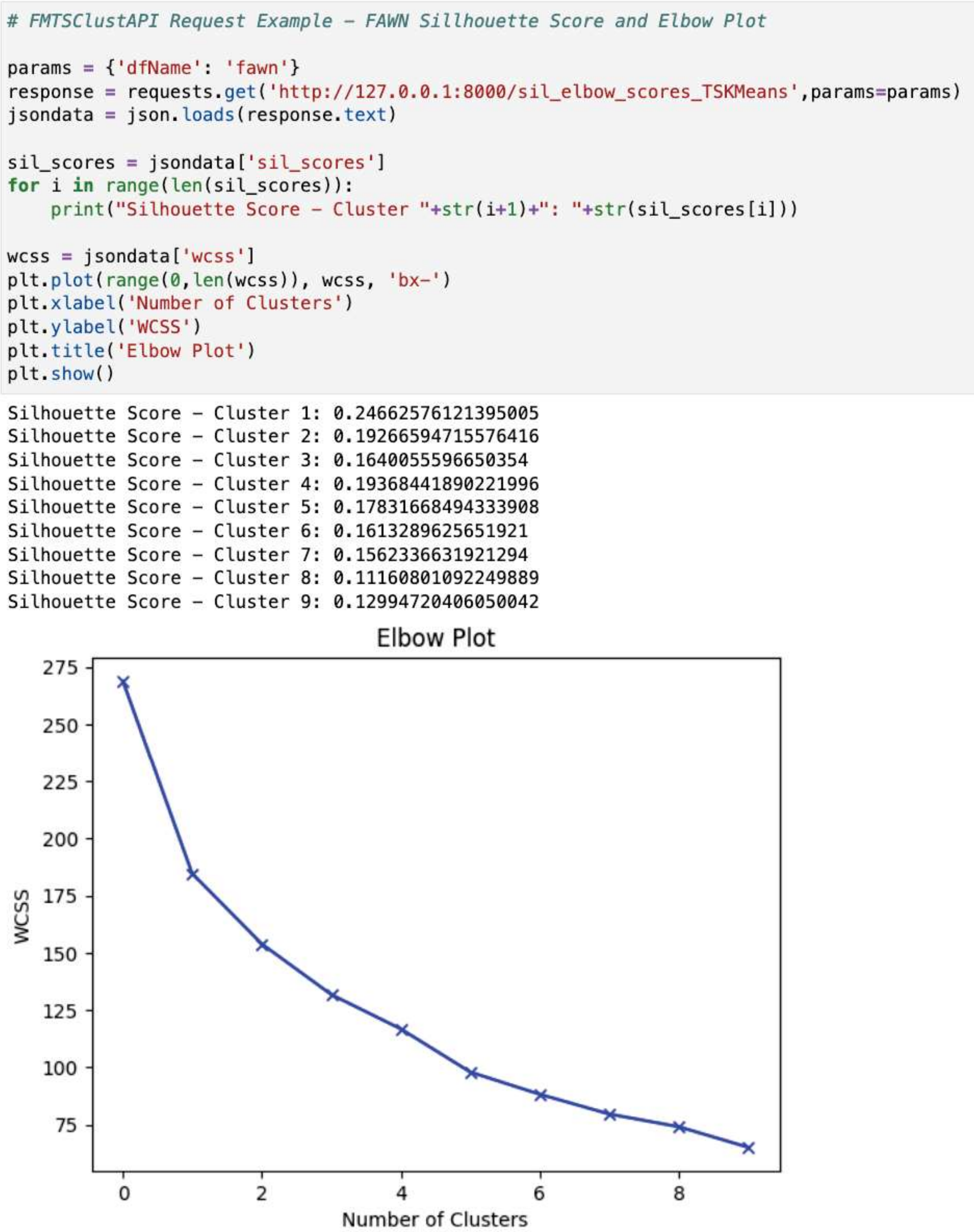
GET /sil_score_TSHClustering Sil Score Tshclustering

GET /cluster_TSHClustering Cluster Tshclustering

Figure 15 – FMTSClustAPI documentation page generated by FastAPI.

quests to endpoints. Figure 16 shows a code snippet from this notebook, which implements a request for the `sil_elbow_scores_TSKMeans` endpoint (using the local server set up at the url: `http://127.0.0.1:8000`). This endpoint calculates silhouette scores and WCSS using the K-means algorithm, as detailed previously. For that, first, the parameters are defined, using a dictionary variable, called `params`. In this case, only one parameter is needed, to indicate the database to be used (`‘fawn’`). Then, a request is made to the endpoint, and its response is stored in the `response` variable. This response is then parsed, following the JSON format, and the values are retrieved using *“keys”*, in this case `‘sil_scores’` and `‘wcss’`, which match the endpoint keys return. And finally, for demonstration purposes, the data returned by the endpoint is printed, the Silhouette Score in text form and the WCSS in graphic form, the Elbow Plot.

Similarly, in Figure 17 another code snippet from the same notebook is presented. In this example, a request is made to endpoint `cluster_TSKMeans`. This endpoint runs clustering with the K-means algorithm, as previously presented. The first three lines work in the same way as the previous example, with one more argument, the number of clusters (in this case ‘2’) in addition to the dataset (`‘fawn’`). After the endpoint returns, the values are retrieved using the keys `‘rmse’` and `‘clustering’`, the same ones implemented by the API. The first corresponds to the total RMSE of the clustering, while the second brings a dictionary, with the assignment of a cluster to each station, using the station ID as the dictionary key. Also, as a simple demonstration, these values are only printed on screen, according to the ID of each meteorological station,



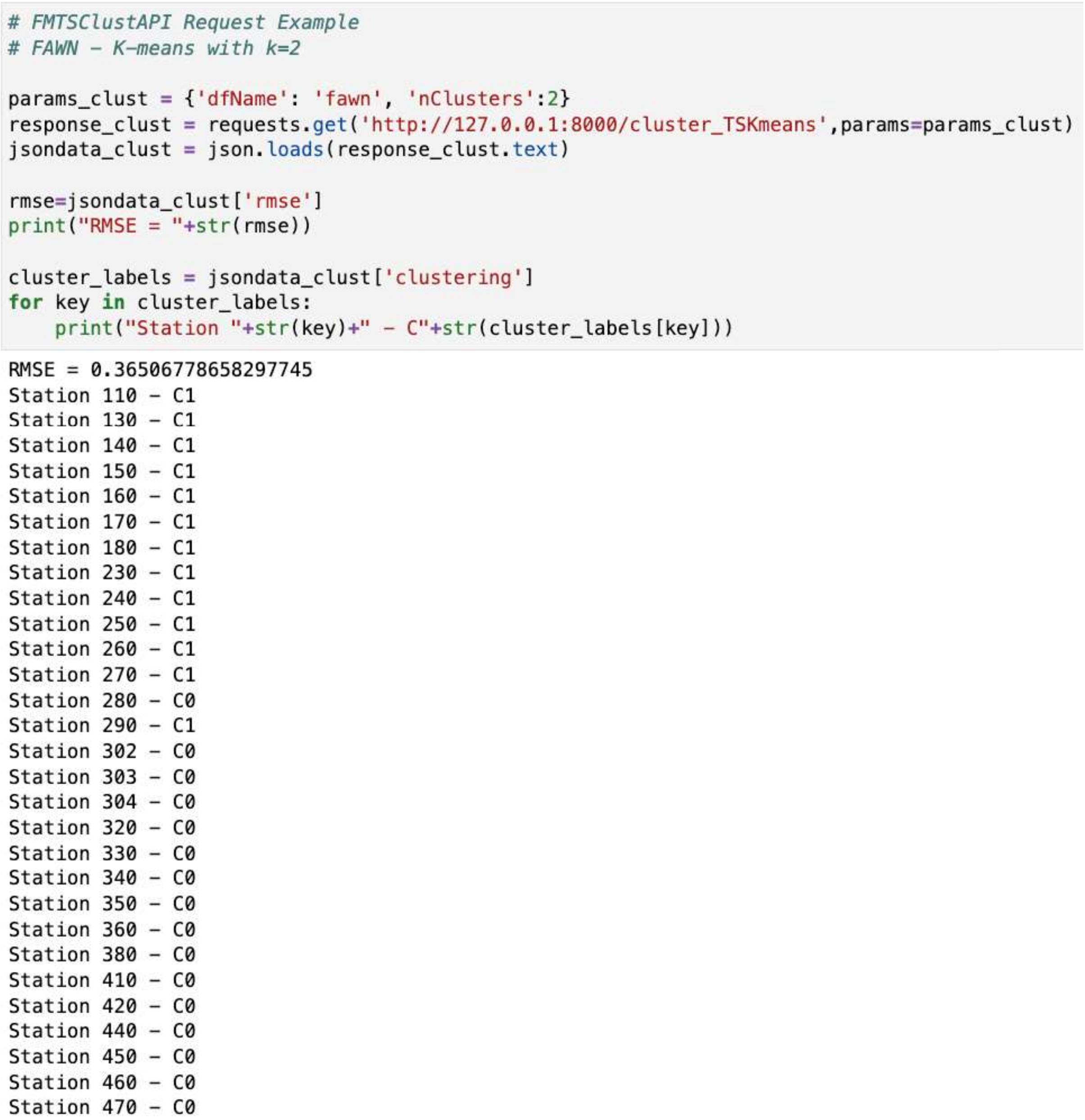
```
# FMTSClustAPI Request Example - FAWN Sillhouette Score and Elbow Plot

params = {'dfName': 'fawn'}
response = requests.get('http://127.0.0.1:8000/sil_elbow_scores_TSKMeans',params=params)
jsondata = json.loads(response.text)

sil_scores = jsondata['sil_scores']
for i in range(len(sil_scores)):
    print("Silhouette Score - Cluster "+str(i+1)+": "+str(sil_scores[i]))

wcss = jsondata['wcss']
plt.plot(range(0,len(wcss)), wcss, 'bx-')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.title('Elbow Plot')
plt.show()
```

```
Silhouette Score - Cluster 1: 0.24662576121395005
Silhouette Score - Cluster 2: 0.19266594715576416
Silhouette Score - Cluster 3: 0.1640055596650354
Silhouette Score - Cluster 4: 0.19368441890221996
Silhouette Score - Cluster 5: 0.17831668494333908
Silhouette Score - Cluster 6: 0.1613289625651921
Silhouette Score - Cluster 7: 0.1562336631921294
Silhouette Score - Cluster 8: 0.11160801092249889
Silhouette Score - Cluster 9: 0.12994720406050042
```

Figure 16 – Code snippet exemplifying a request to the `sil_elbow_scores_TSKMeans` endpoint.

but from here they could be consumed for different purposes.

Finally, it is worth highlighting that as it is an API, FMTSClustAPI does not necessarily need to be consumed in the Python language. Requests, using the http protocol, can be made by applications implemented in any programming language and the returns consumed equally, as they are encoded using JSON standards.



```
# FMTSClustAPI Request Example
# FAWN - K-means with k=2

params_clust = {'dfName': 'fawn', 'nClusters':2}
response_clust = requests.get('http://127.0.0.1:8000/cluster_TSKmeans',params=params_clust)
jsondata_clust = json.loads(response_clust.text)

rmse=jsondata_clust['rmse']
print("RMSE = "+str(rmse))

cluster_labels = jsondata_clust['clustering']
for key in cluster_labels:
    print("Station "+str(key)+" - C"+str(cluster_labels[key]))
```

```
RMSE = 0.36506778658297745
Station 110 - C1
Station 130 - C1
Station 140 - C1
Station 150 - C1
Station 160 - C1
Station 170 - C1
Station 180 - C1
Station 230 - C1
Station 240 - C1
Station 250 - C1
Station 260 - C1
Station 270 - C1
Station 280 - C0
Station 290 - C1
Station 302 - C0
Station 303 - C0
Station 304 - C0
Station 320 - C0
Station 330 - C0
Station 340 - C0
Station 350 - C0
Station 360 - C0
Station 380 - C0
Station 410 - C0
Station 420 - C0
Station 440 - C0
Station 450 - C0
Station 460 - C0
Station 470 - C0
```

Figure 17 – Code snippet exemplifying a request to the `cluster_TSKMeans` endpoint.

Both Version I (`FMTSClust.py` library) and II (`FMTSClustAPI`) are available in the project's GitHub repository. All implementation was done locally, and the API is not yet online. The concern, at that moment, was with the implementation and validation of the functions, and then it will be made available online for access by other applications.

## 5.4 Complementary Documentation

The documentation serves as a guide or reference for users, developers, and other stakeholders involved in the software development process. In the previous sections, conceptual and technical details about the implementation of FMTSClust, to the point of introducing the framework's general purpose and basic usage. In order to complement the explanation of the implemented framework, as well as encourage collaboration, complementary documentation artifacts are presented below, which can be useful in different situations, both for better use and for correction/extension of FMTSClust.

One useful artifact in this scenario is a UML Deployment Diagram (KOBRYN, 2000), as shown in Figure 18. A deployment diagram serves as a visual representation of the architecture of a software system, illustrating how software components are distributed across nodes in a computing environment. It provides a comprehensive overview of the deployment configuration. These diagrams help stakeholders understand the spatial arrangement of a system's elements, facilitating the planning and management of deployment strategies.

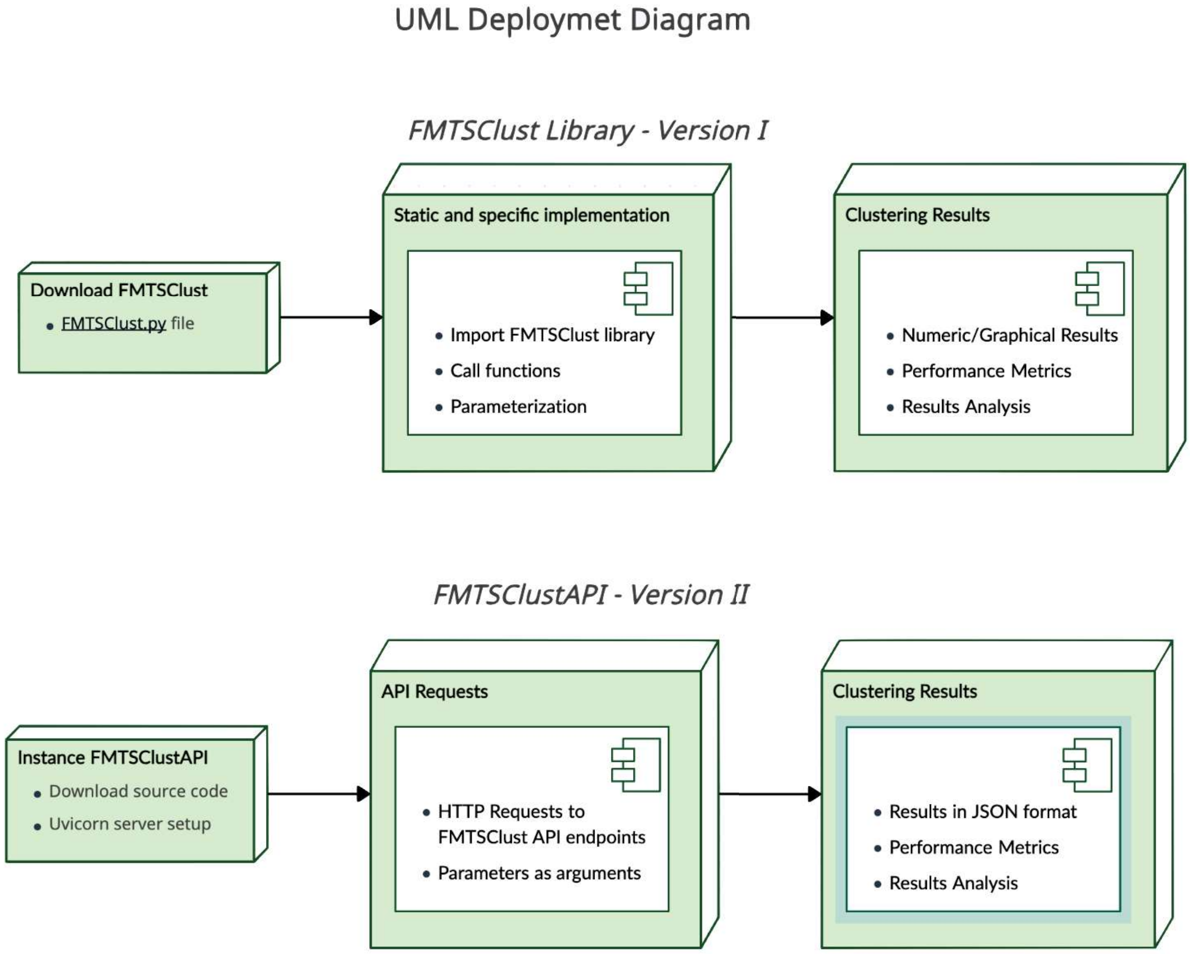
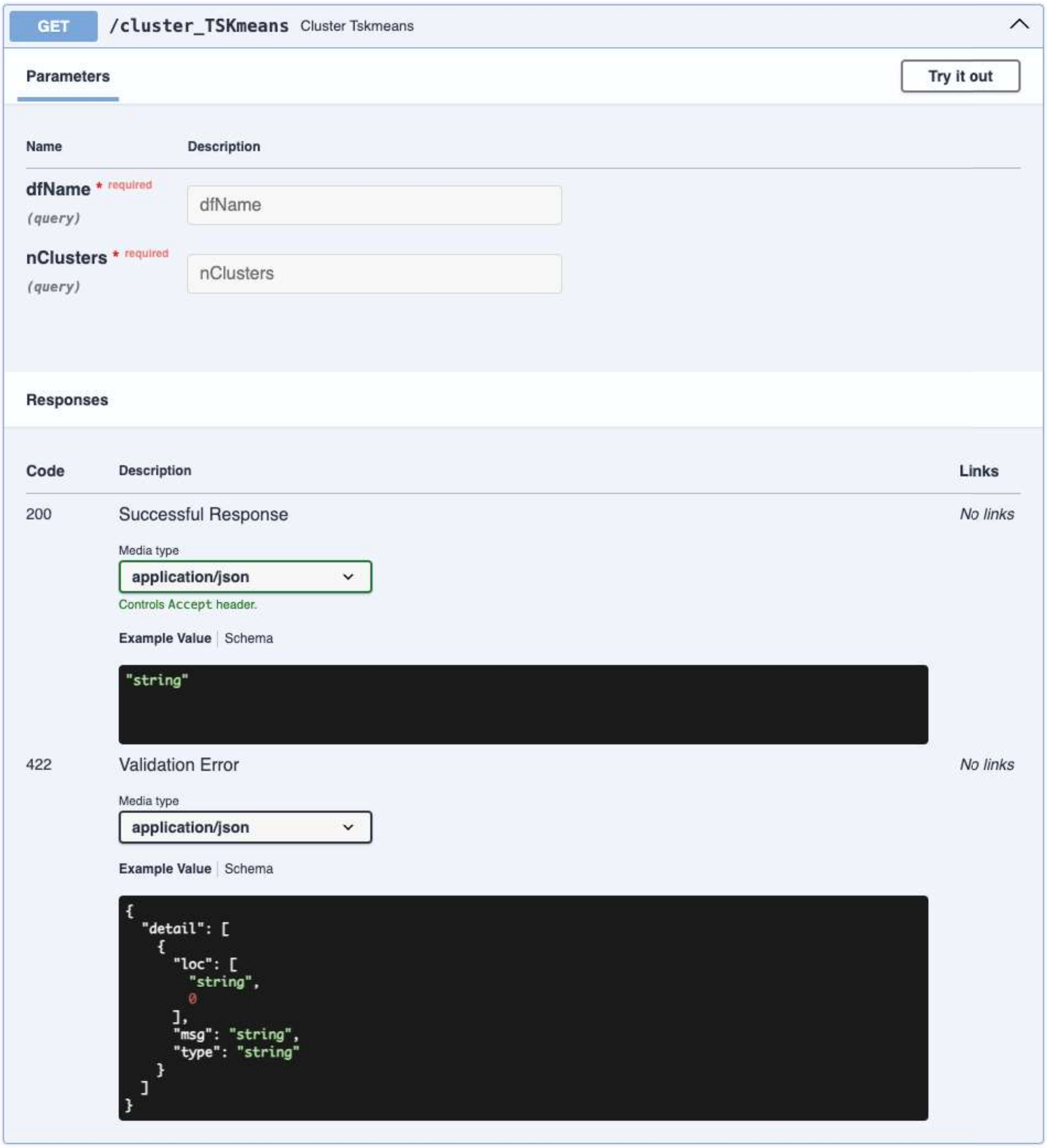


Figure 18 – FMTSCLust UML Deployment Diagram.

In the figure, two deployment diagrams are presented, one for each version, visually explaining how to instantiate FMTSClust. In the first, the instantiation starts with the download of the FMTSClust python library, followed by its import and call of functions with the desired parameters, ending with the analysis of numerical and graphical results, provided by the return of the functions. In the second, the use of the framework starts from the instantiation of the API, with the basic and standard setup of the

Uvicorn server being suggested, then the HTTP requests to the API, with the parameters passed as arguments, and finally the analysis of results from the returns from FMTSClustAPI in JSON format.

Furthermore, in line with the concern with documentation, the use of the FastAPI framework was motivated precisely for this reason, as it provides API documentation automatically and efficiently. An interface for testing requests is even provided, as exemplified in the next figures. Figure 19 shows the request test interface, generated by FastAPI for the TSKmeans endpoint, with the `dfName` and `nClusters` parameter fields blank, as well as the response fields.



| Name | Description |
|---|---|
| dfName * required<br>(query) | dfName |
| nClusters * required<br>(query) | nClusters |

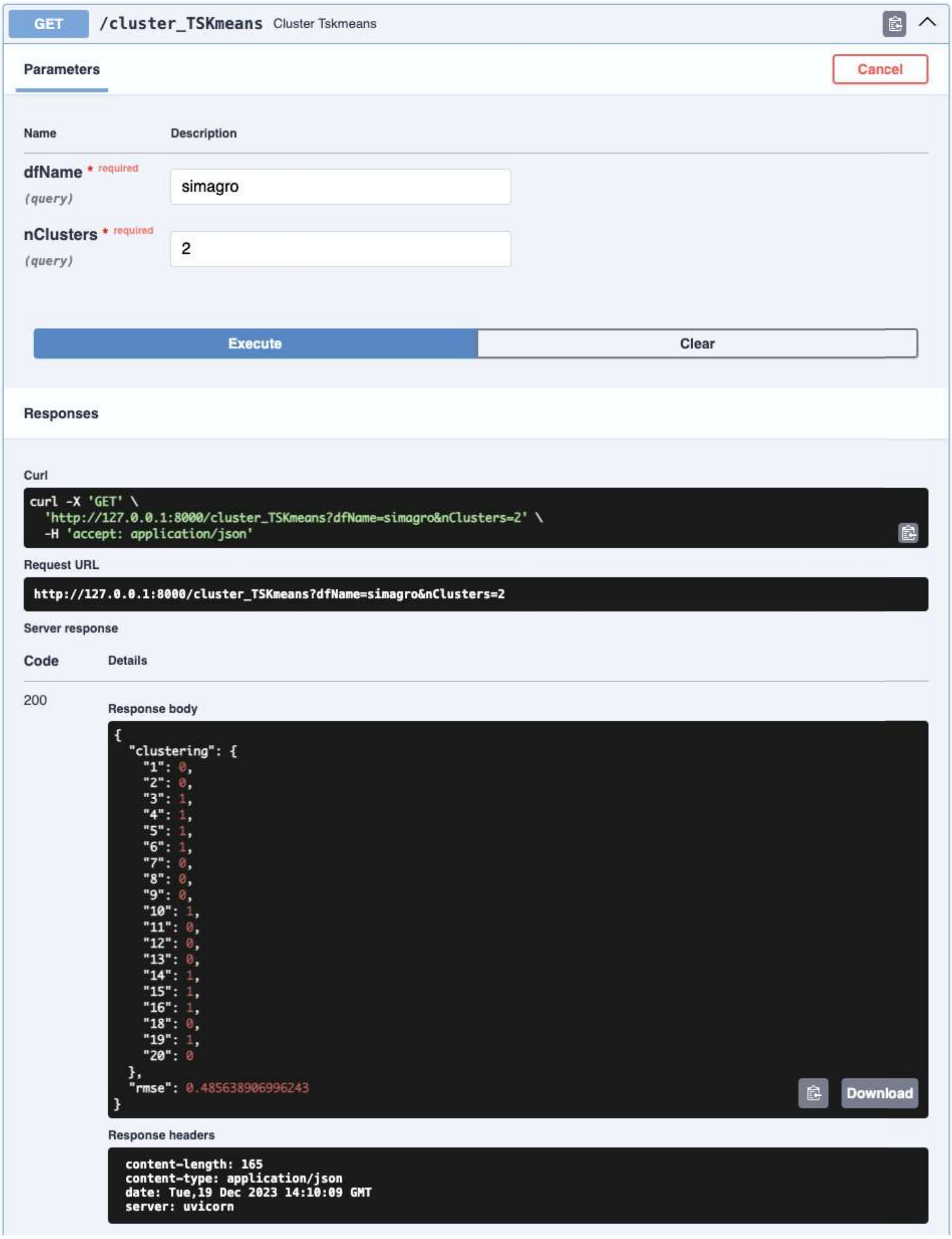
| Code | Description | Links |
|---|---|---|
| 200 | Successful Response | No links |
| 422 | Validation Error | No links |

```
"string"
```

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

Figure 19 – Documentation interface provided by FastApi for FMTSClust - TSKMeans endpoint.

By clicking on *“Try it out”* button and entering valid parameters in the corresponding fields (such as `simagro` for `dfName` and 2 for `nClusters`), as shown in Figure 20, and executing the request, FMTSClust operates and its result is returned, in JSON format, as shown in the response fields.



/cluster_TSKmeans Cluster Tskmeans

Parameters

Name | Description

dfName * required (query): simagro

nClusters * required (query): 2

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/cluster_TSKmeans?dfName=simagro&nClusters=2' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/cluster_TSKmeans?dfName=simagro&nClusters=2
```

Server response

Code | Details

200

Response body

```
{
  "clustering": {
    "1": 0,
    "2": 0,
    "3": 1,
    "4": 1,
    "5": 1,
    "6": 1,
    "7": 0,
    "8": 0,
    "9": 0,
    "10": 1,
    "11": 0,
    "12": 0,
    "13": 0,
    "14": 1,
    "15": 1,
    "16": 1,
    "18": 0,
    "19": 1,
    "20": 0
  },
  "rmse": 0.485638906996243
}
```

Response headers

```
 content-length: 165
 content-type: application/json
 date: Tue,19 Dec 2023 14:10:09 GMT
 server: uvicorn
```

Figure 20 – Example of request using the documentation interface provided by FastApi for FMTSClust - TSKMeans endpoint.

In this case, the clustering of the dataset referring to the simagro database (detailed in the next chapter) was generated into 2 clusters, with the corresponding cluster number being returned for each station (represented by its ID), $0$ or $1$. This type of automated documentation that provides a test interface is very helpful in the software development phase, especially in the API format that requires testing requests, and is therefore very convenient.

As this is an ongoing project, it is intended to continue producing FMTSClust documentation artifacts, as far as possible and necessary, in order to monitor the evolution of implementation, corrections, adjustments and extensions. To ensure updated documentation, it is recommended to observe the project's public repository on github, as all artifacts will be made available there.

## 5.5 Chapter Remarks

This chapter presented the framework (FMTSClust) implementation. Version I was implemented in the form of a pyhton library, containing auxiliary functions implemented for each FMTSClust operation step. Version II was implemented in the form of an API, describing the endpoints implemented to execute the framework. Both versions are accompanied by python notebooks to demonstrate their use. The framework implementation is available in the project's GitHub repository, in order to assist in the development of other projects and also encourage contributions.

# 6 EXPERIMENTS, RESULTS, AND DISCUSSION

This chapter presents the case studies developed, through experiments with data from real agricultural applications, to demonstrate examples of the framework's usefulness. The case studies not only validated, but presented practical applications, offering contributions to the design of the framework with real analyses. Initially, the first case study is detailed, consisting of a round of experiments carried out with a dataset from SIMAGRO-RS, a network of meteorological stations in the state of Rio Grande do Sul, used to monitor agroclimatic conditions for general agricultural purposes, from the perspective of grain production of the main crops in RS. Afterwards, a second case study is reported, composed of another cycle of experiments with a different dataset, owned by FAWN, from the state of Florida/USA, used in an agricultural disease control systems. Furthermore, an analysis and discussion of the results is presented in order to validate the proposed framework for clustering MTS in agricultural applications.

## 6.1 Case Study I - SIMAGRO-RS

In the state of Rio Grande do Sul (RS), the southernmost state in Brazil, which has a largely agricultural economy, some efforts seek to assist producers to collect and analyze climate data, in order to extract useful insights for different seasons and crops. The SIMAGRO-RS (Sistema de Monitoramento e Alertas Agroclimáticos)[1] is the Rio Grande do Sul Agroclimatic Monitoring and Alerts System, and is one project that has emerged in this direction. With weather stations deployed in different locations in RS, SIMAGRO-RS records weather data and monitors the state's climate, providing information through a comprehensive and dynamic online platform, with meteorological maps and a repository of RS weather data, to help the agricultural sector with planning activities and guide short, medium, and long-term actions (GOVERNO DO ESTADO DO RIO GRANDE DO SUL, 2022).

The SIMAGRO-RS project is an initiative of the RS state government, through the Department of Agriculture, Livestock, and Rural Development (Secretaria de Agricul-

[1] SIMAGRO-RS - http://www.simagro.rs.gov.br

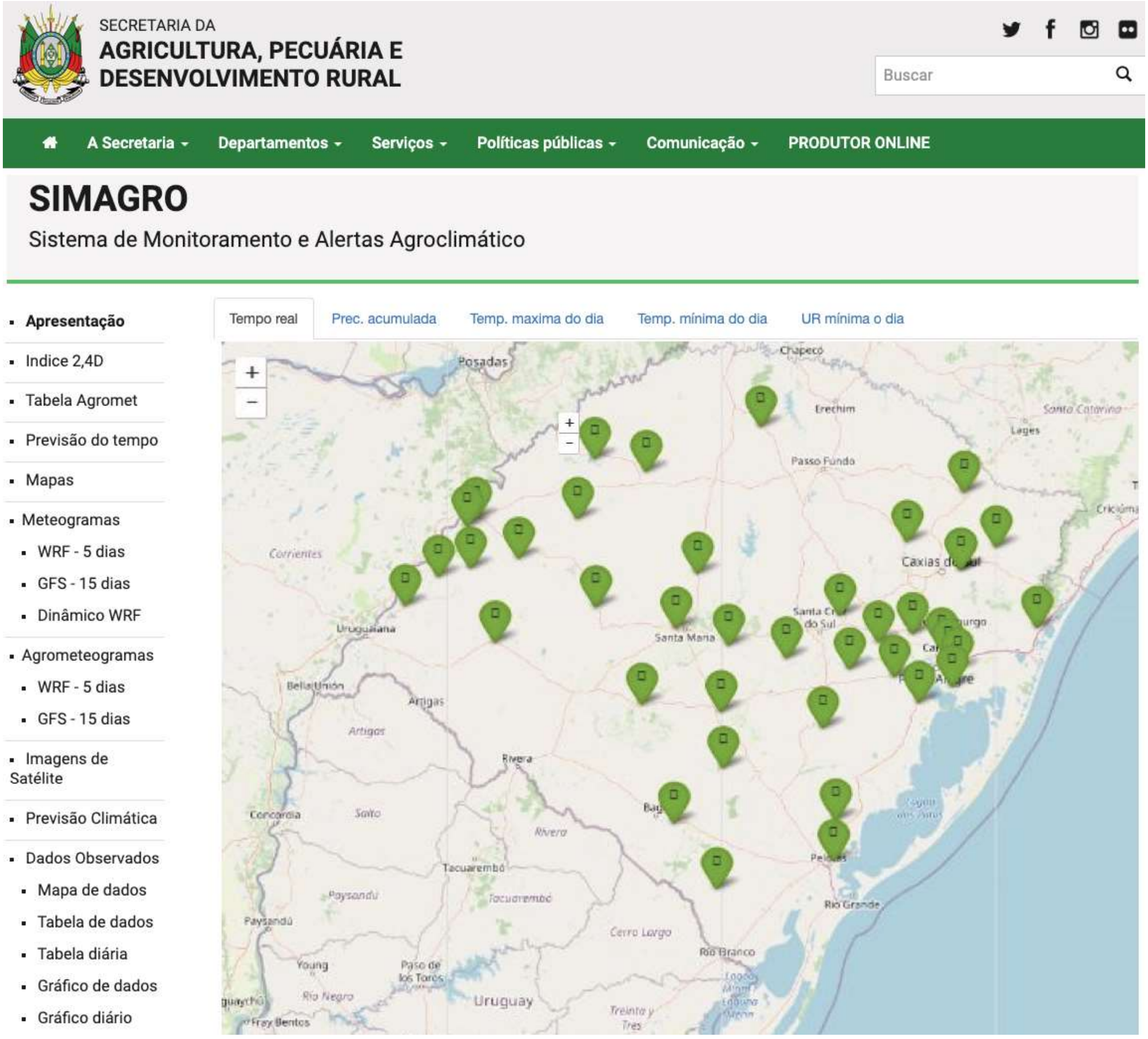


Figure 21 – SIMAGRO-RS web interface, with a map of meteorological stations in the RS state.

tura, Pecuária e Desenvolvimento Rural do RS), and beyond its role as a repository of meteorological data, provides a critical foundation for the advancement of precision agriculture and sustainable resource management. Started in 2019 and founded on the principle of providing historical and real-time climate information, SIMAGRO-RS serves as an important tool for understanding the intricate interplay between weather dynamics and agricultural activities. Since this system is quite recent and with the deployment of meteorological stations still in progress, the data produced by SIMAGRO-RS are starting to be used in climate studies in the state of RS, providing a great opportunity for research such as the one presented here. As the agricultural sector faces increasing challenges posed by climate change and fluctuating weather patterns, the insights drawn from SIMAGRO-RS are poised to play an instrumental role in shaping resilient and adaptive agricultural practices that can secure food production in the face of a dynamic climate landscape.

Figure 21 shows an example of data visualization on the SIMAGRO-RS platform,

Table 6 – SIMAGRO-RS Meteorological Stations used in this study.

| ID | Station (City) | Latitude (Deg) | Longitude (Deg) |
|---|---|---|---|
| 1 | Pinheiro Machado | -31.577629 | -53.383749 |
| 2 | Piratini | -31.447456 | -53.103847 |
| 3 | São Sepé | -30.171961 | -53.571799 |
| 4 | Itaqui | -29.129403 | -56.551439 |
| 5 | Maçambara | -29.147451 | -56.067150 |
| 6 | Rosário do Sul | -30.250008 | -54.916827 |
| 7 | Getúlio Vargas | -27.888607 | -52.228262 |
| 8 | Ilópolis | -28.926757 | -52.126125 |
| 9 | Barra do Ribeiro | -30.299519 | -51.304567 |
| 10 | Cachoeira do Sul | -30.032233 | -52.893182 |
| 11 | Canguçu | -31.398944 | -52.677631 |
| 12 | Herval | -32.028922 | -53.394227 |
| 13 | Lavras do Sul | -30.810365 | -53.899211 |
| 14 | São Borja | -28.659901 | -56.009982 |
| 15 | Bossoroca | -28.731386 | -54.903388 |
| 16 | Jaguari | -29.491148 | -54.689576 |
| 17 | Caxias do Sul | -29.161301 | -51.191962 |
| 18 | Porto Vera Cruz | -27.734706 | -54.899875 |
| 19 | Sobradinho | -29.415668 | -53.027771 |

through the state map with pins representing the weather stations. The platform also has options for viewing data in table format, as well as extracting data in file format. With its expansive coverage and high-frequency data collection, SIMAGRO-RS offers an excellent interface for researchers and producers to gain insight into the state agro-climatic conditions. This way, SIMAGRO-RS empowers agricultural stakeholders with a new layer of information to make informed decisions related to crop selection, planting times, irrigation strategies, pest control measures, and other agricultural activities.

### 6.1.1 SIMAGRO-RS Dataset

SIMAGRO-RS database records a diverse array of more than twenty meteorological variables, every half hour, such as temperature, dew point, precipitation, wind speed, wind direction, solar radiation, and relative humidity, collected through an extensive network of meteorological stations distributed across the state. The database enables the monitoring of climate over time and offers a granular view of weather patterns on regional and local scales.

For this case study, weather data were first extracted from the SIMAGRO-RS database in `.txt` files. Data were extracted from $19$ meteorological stations, which represent the first stations installed since the beginning of the project, resulting in a $58MB$ dataset. Table 6 presents basic information on the $19$ meteorological stations used in

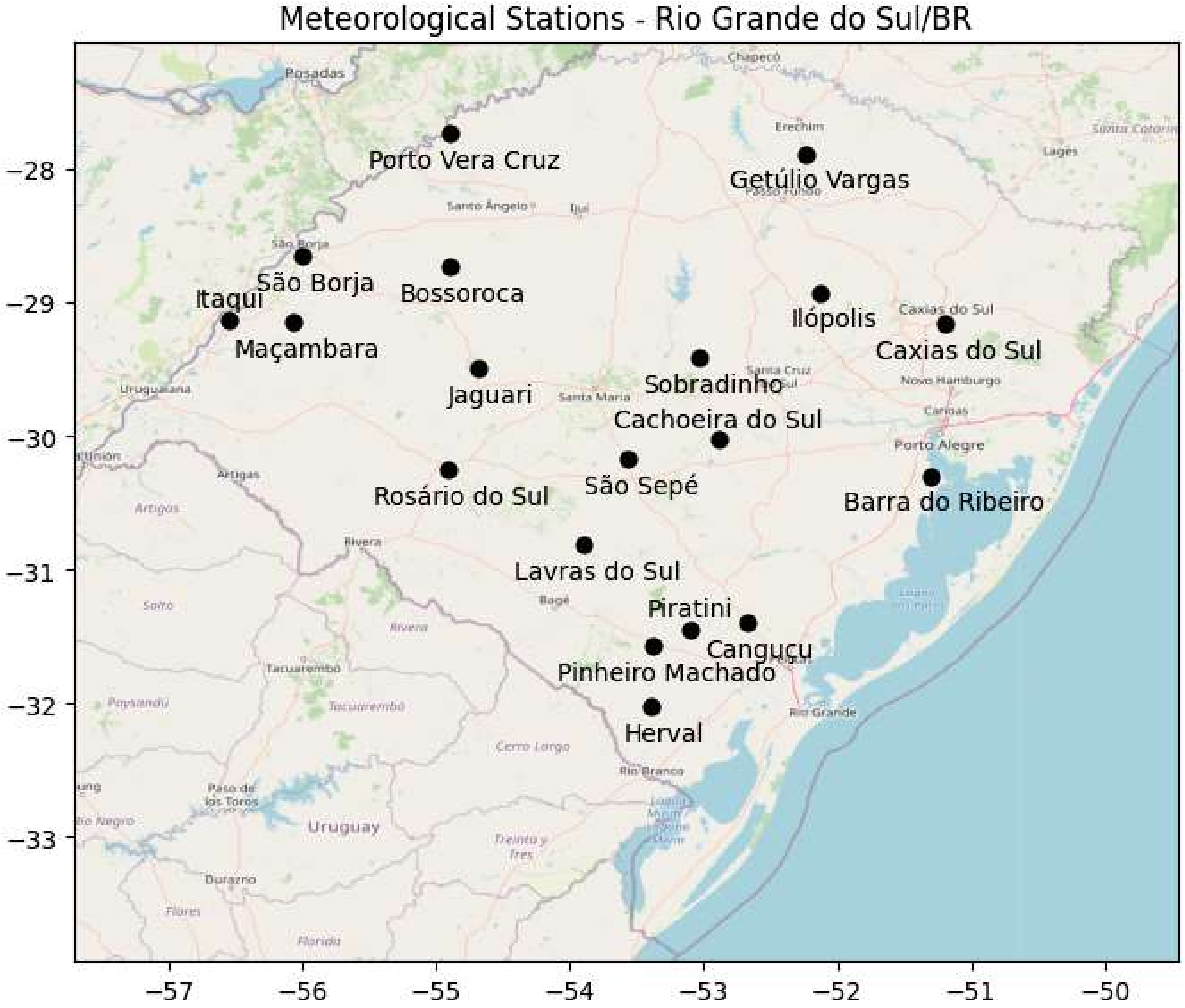


Figure 22 – SIMAGRO-RS Meteorological Stations.

this case study, and Figure 22 graphically presents the distribution of stations across the state of RS. Currently, more stations are in operation, however, as the project is recent, many of the stations are quite new, implemented in the current year, and, consequently, do not have the historical data needed for clustering tasks.

### 6.1.2 Data Preprocessing

In order to prepare the data to be used in clustering tasks, following the proposed framework, some preprocessing steps were performed, such as filtering, quality checking, aggregation, and standardization. Below are detailed the results of this preprocessing phase for this first case study.

#### *6.1.2.1 Filtering*

After extraction, the data were filtered by desired variables and the period of time to be observed. To select the variables of interest, the relevance in agricultural production of RS crops was considered, and the variables Temperature (T), Relative Humidity

(RH), Rainfall (RF), and Solar Radiation (SR) were selected. Temperature affects the growth and development of crops and extreme heat or cold can stress crops and lead to reduced yields. Also, since RS experiences distinct wet and dry seasons, adequate and well-distributed rainfall is important for crop growth. Sunlight is also essential for photosynthesis, enabling plants to grow and produce crops. Likewise, relative humidity in the air can impact crop health and susceptibility to diseases, with high humidity increasing fungal diseases risks and low humidity leading to water stress in plants. Therefore, these were the variables selected for this case study.

Regarding the time period, the agricultural production season in the state of RS was considered, for the main grain crops, such as soybeans, which mainly comprise spring and summer (POTT et al., 2021). Monitoring the climate during the grain production season is a fundamental aspect of modern agriculture. It helps farmers optimize crop management, reduce risks, and adapt to changing environmental conditions, ultimately contributing to higher yields and sustainable agriculture. Thus, data were selected from the entire grain production season in RS, between October 1st and March 31st, therefore totaling 182 days, from the last three agricultural production seasons (20/21, 21/22 and 22/23).

#### *6.1.2.2 Quality Checking*

According to the proposed framework, with regard to quality checking, first, a scan for missing and null values was performed. For the $19$ stations used in this study, $475$ missing values and $12$ null values were detected, including all variables, which represent about $4.5\%$ of the dataset. Given the high frequency of sensing and the ability to use aggregated values, it was decided to disregard these missing and null values, and aggregate all variables into daily values.

Next, an outlier analysis was performed, using the outlier capping strategy with Interquartile Ranges (IQR), as detailed previously, to identify discrepant values with the distribution of each variable. After checking for outliers, the following number of measurements were identified, together with upper and lower outliers: $136$ outliers for T; $65$ for RH; $2,126$ for RF; and, $22$ for SR. These values were then replaced by the upper and lower limit, respectively.

### 6.1.3 Feature Extraction

Feature extraction was conducted in SIMAGRO-RS dataset as proposed in the framework, based on two steps: Aggregation and Standardization. In agricultural grain production, as it is the case, aggregating weather data into daily averages can help identify trends and patterns, which are often more relevant for decision-making than the minute-to-minute variations. Furthermore, standardization helps to combine and compare data from different sources or locations. In combination, aggregation and

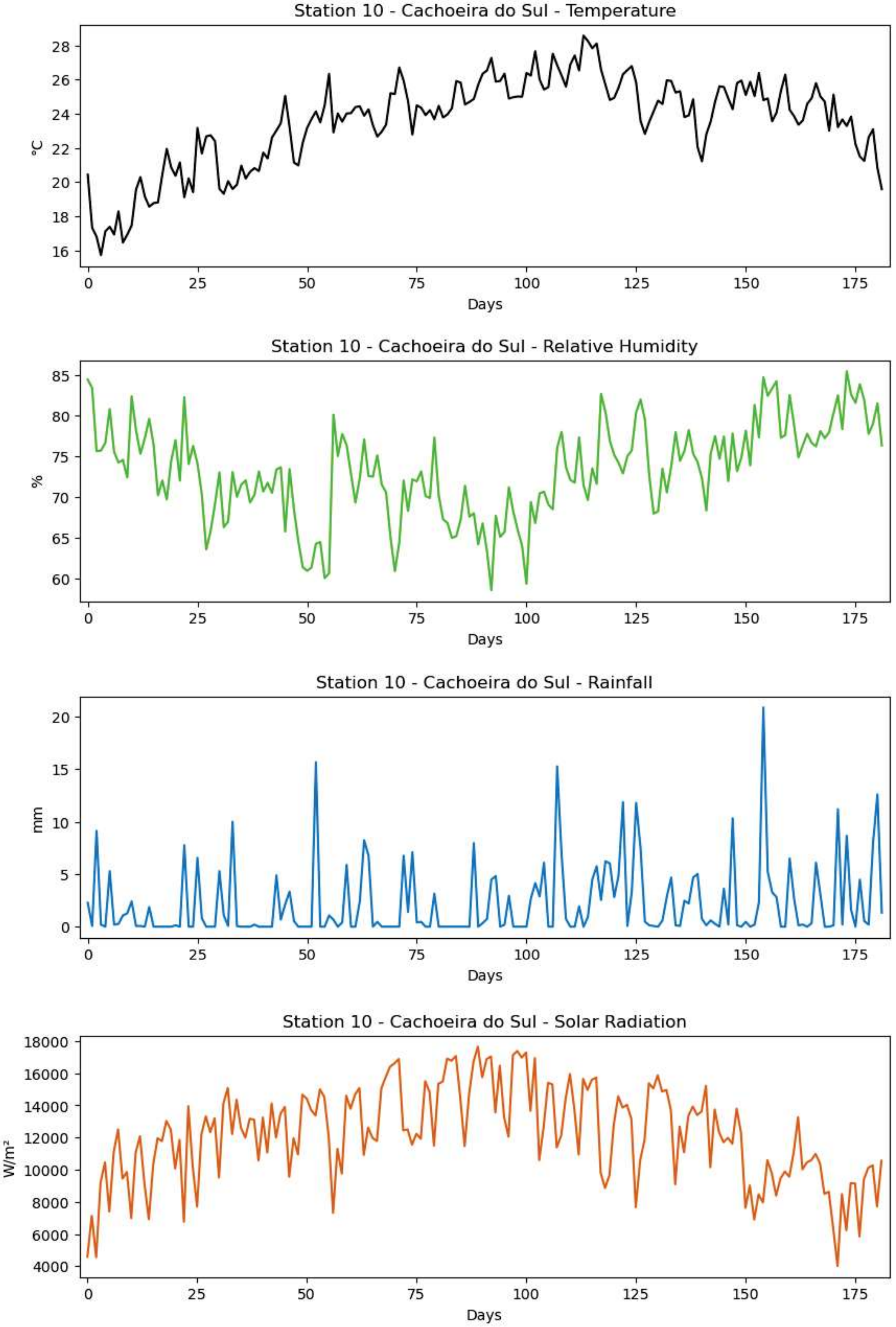


Figure 23 – Visualization of the variables of interest in SIMAGRO dataset (Temperature - black; Relative Humidity - green; Rainfall - blue; Solar Radiation - orange), separately.

standardization enhance the quality of weather data by simplifying it, reducing variability, and making it more suitable for various analytical methods, such as clustering. Next, the details of the application of these two steps in the SIMAGRO-RS dataset are reported.

#### *6.1.3.1 Aggregation*

For this case study, the data were aggregated into daily values, considering good aggregation practices for each variable in agricultural context. Initially, for the variables T and RH, daily averages were used, while for RF and SR, daily accumulated values were used. After this first aggregation, given the three seasons under analysis (20/21, 21/22 and 22/23), the average *DOY* values of T, RH, RF and SR were calculated for each day of these seasons, in order to identify a single value for each day of the season that represents the data collected in the last three years. There is a data gap for stations $14$ to $19$, in the first two months (October and November) of the first season (20/21). For these stations, in this period, only data from the second and third seasons were used to compute the aggregated values. Then, this process resulted in an aggregated dataset containing $3,458$ (rows) values for each variable ($182$ days of the season multiplied by the $19$ weather stations).

Figure 23 presents the aggregated data for each one of the four variables of this study, separately: first graph presents the Temperature, in black; second graph presents the Relative Humidity, in green; third graph presents Rainfall, in blue; and, fourth graph presents Solar Radiation, in orange. The plotted data is from meteorological station number 10, in the city of Cachoeira do Sul, chosen at random, as an example. In this image it is possible to observe the average values, aggregated daily (during the 182 days, between October and March), for the last three seasons of agricultural production.

#### *6.1.3.2 Standardization*

After aggregation, according to the framework, the next step was standardization. Standardized values were calculated for the four variables, T, RH, RF, and SR. Thus, all of them now have a new representation, with a smaller range of variation, in order to enable a fairer correlation and comparison. Figure 24 shows the distribution of variables, plotted on the same scale, before (top) and after (bottom) standardization. Then, it is possible to clearly observe the difference between the original and standardized datasets, looking especially at the vertical axes. When plotted under the same axis without standardization, like the top graph, only one variable stands out, the one with the greatest amplitude, while the others are flattened. After the standardization, they are all represented around the vertical axis 0, like the graph at the bottom of this figure.

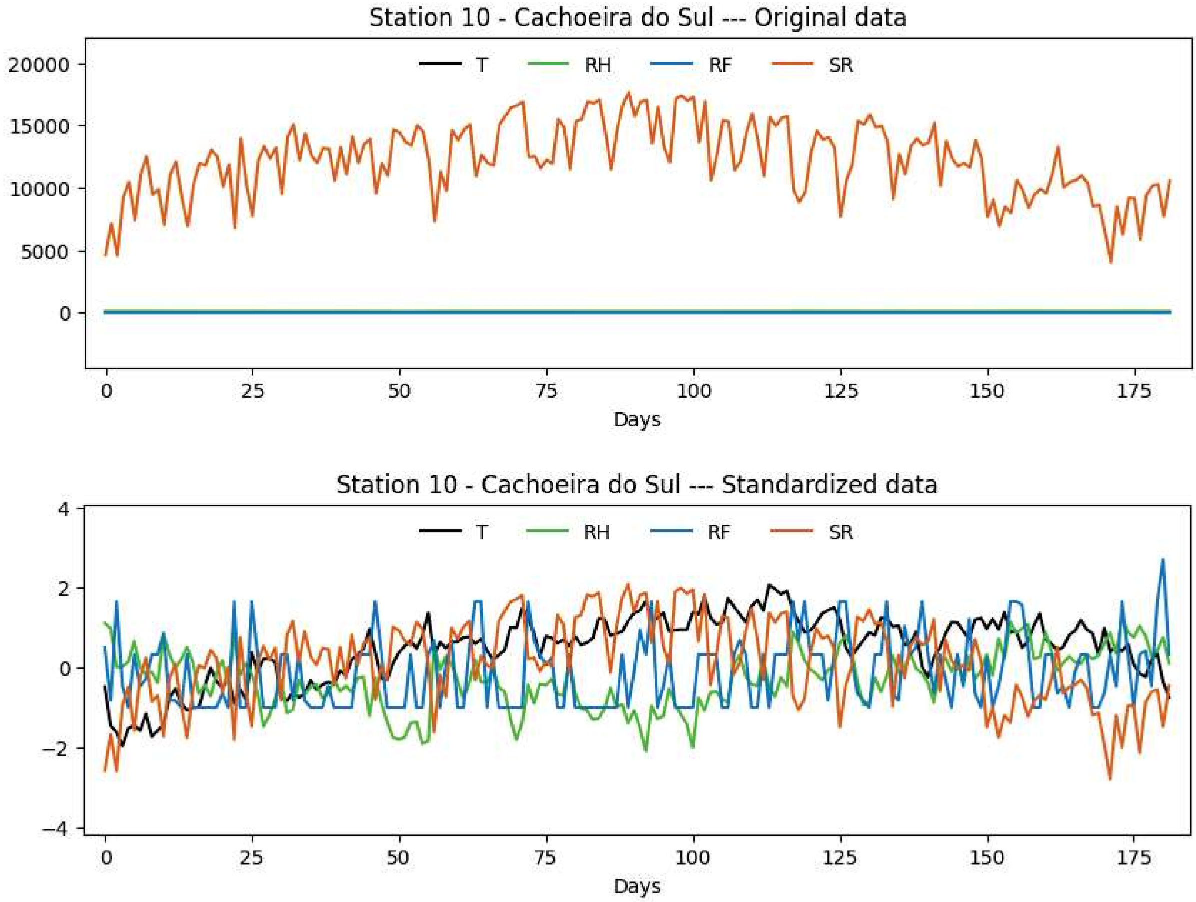


Figure 24 – Data distribution after standardization (Temperature - black; Relative Humidity - green; Rainfall - blue; Solar Radiation - orange).

### 6.1.4 Clustering Results

After all the preprocessing steps, the dataset prepared for clustering was composed of $19$ standardized multivariate time series (one for each meteorological station) of the same size, containing 182 timestamps each (days). In each timestamp, the time series stores 4 values, representing the climatic variables used in this study. So, first a search was made for the optimal number of clusters, then clustering was carried out and, finally, the results were evaluated, as detailed below.

#### *6.1.4.1 Number of clusters*

From the dataset prepared for clustering, the *Elbow Plot* and *Silhouette Score* methods were used to define the optimal number of clusters, according to the proposed framework. The search for an optimal number of clusters in this case study was performed in the range of 2 to 10, according to the reasonableness of the context, within a viable limit for the agricultural context and the number of stations in use. First, the elbow plots were generated for the K-means and K-medoids algorithms, as shown in Figure 25. These graphs shows the elbow plot for the dataset used in this case study, calculated from the difference in distances between the time series and the barycentric

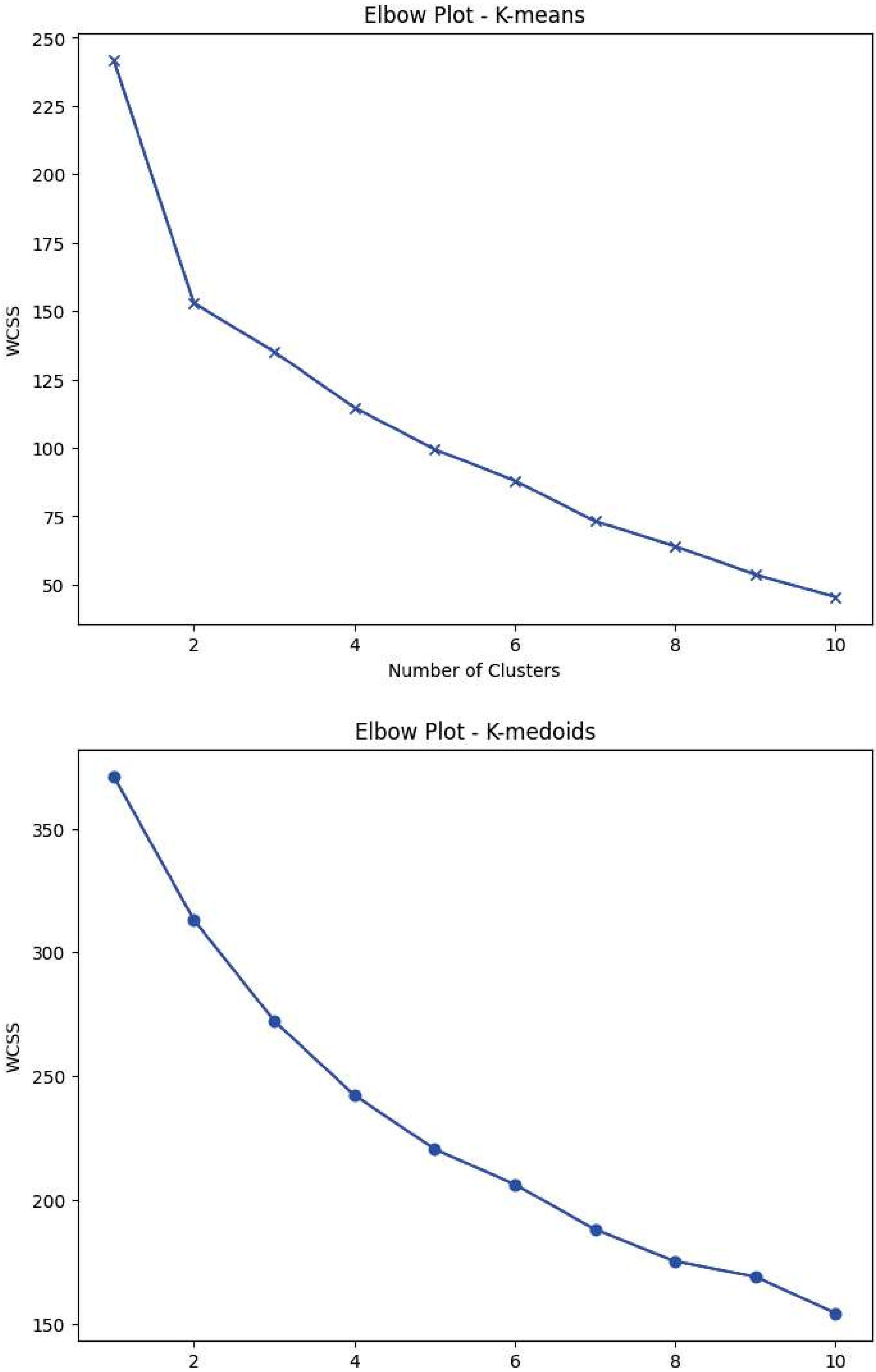


Figure 25 – Elbow Plots for the K-means and K-medoids algorithms on the SIMAGRO-RS dataset.

Table 7 – Silhoutte Score for SIMAGRO-RS dataset.

| N of clusters | K-means | K-medoids | Hierarc. Clust. |
|---|---|---|---|
| **2** | **0.262** | **0.276** | **0.249** |
| 3 | 0.160 | 0.168 | 0.165 |
| 4 | 0.136 | 0.161 | 0.141 |
| 5 | 0.148 | 0.175 | 0.148 |
| 6 | 0.150 | 0.122 | 0.150 |
| 7 | 0.158 | 0.134 | 0.158 |
| 8 | 0.125 | 0.093 | 0.139 |
| 9 | 0.135 | 0.044 | 0.135 |
| 10 | 0.130 | 0.044 | 0.130 |

time series of each cluster. From these graphs, it is observed that for the K-means algorithms there is some indication of the optimal number of clusters, with the elbow at the value of $k = 2$, whereas for the K-medoids there is no apparent elbow. Analyzing these graphs, we have a possible value for $k$, but they are not conclusive enough. This is common when searching for an ideal number of clusters, due to the difficulty of a single metric defining this precisely. Therefore, the other metric was used in a complementary way.

The *Silhouette Score* metric quantifies how similar an object is to its own cluster compared to other clusters, varying between -1 and 1, where a higher score indicates better-defined clusters. Table 7 presents the scores for each algorithm and cluster number, with the highest score highlighted for each algorithm. Based on these results, analyzing the *Elbow Plot* and the *Silhouette Score* in a complementary way, the optimal number of clusters for this study with the SIMAGRO-RS dataset was defined as $k = 2$.

It is important to note that even though the silhouette score points to an optimal number of clusters, the highest values (around $0.25$) of the coefficient indicate that the clustering quality is closer to the average $(0.0)$ than to the maximum value $(1.0)$. This demonstrates that the data in the dataset is not easy to divide, reinforcing the importance of investigation and clustering.

### *6.1.4.2 Clustering Algorithms Results*

After finding the optimal number of clusters for the dataset of this case study, the K-means, K-medoids and Hierarchical Agglomerative Clustering algorithms were used to group the SIMAGRO-RS meteorological stations. The basic result of clustering is the spatial representation, on the map, of stations divided into their clusters. Figure 26 shows the clustering obtained by the K-means algorithm, Figure 27 shows the clustering obtained by the K-medoids algorithm, and Figure 28 shows the clustering results for Hierarchical Agglomerative algorithm.

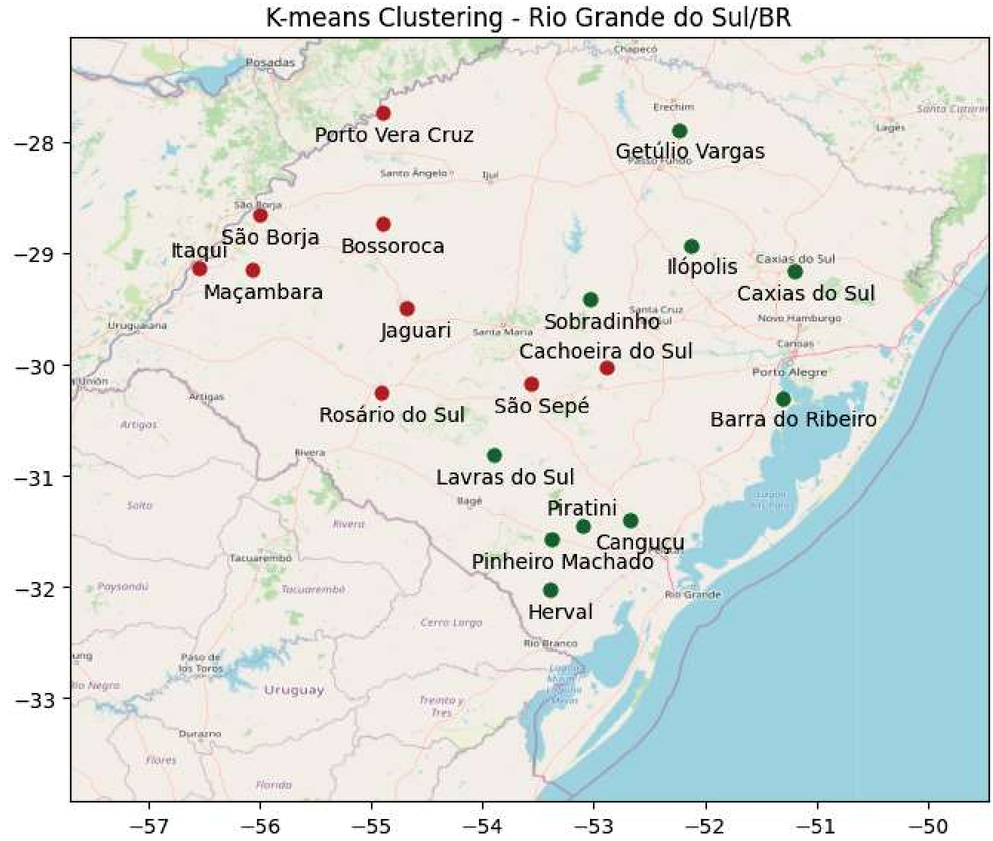


Figure 26 – SIMAGRO-RS K-means Clustering with $k = 2$ (red and green clusters).

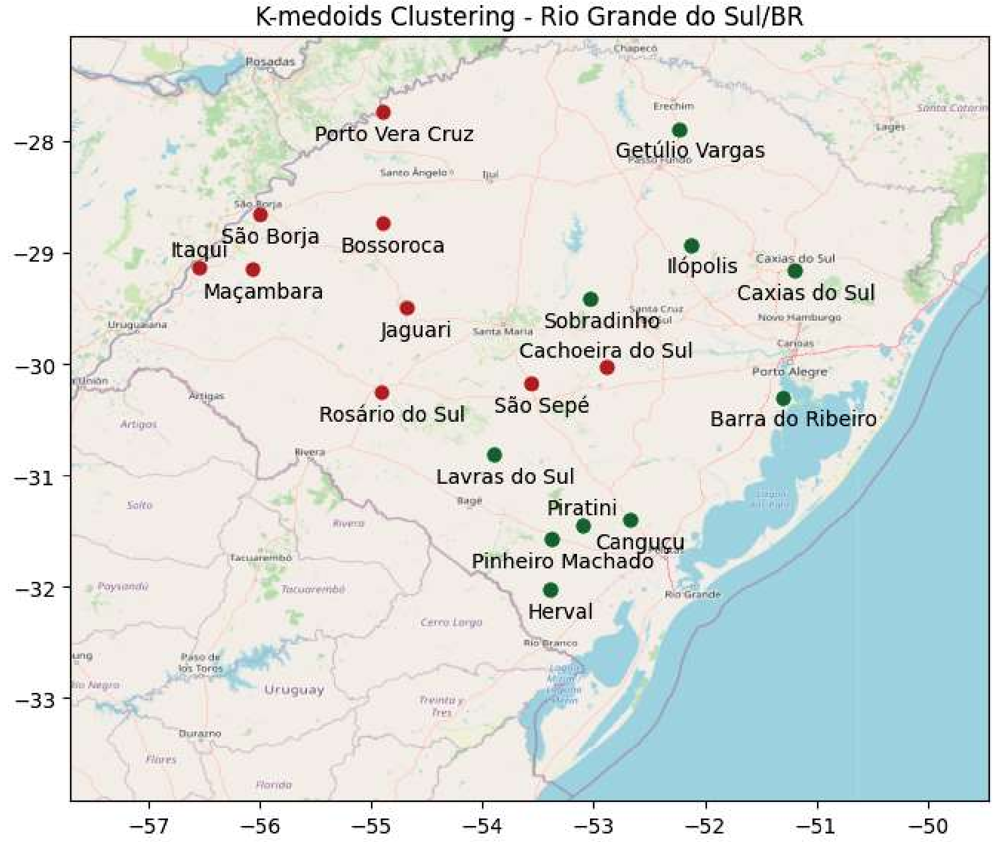


Figure 27 – SIMAGRO-RS K-medoids Clustering with $k = 2$ (red and green clusters).

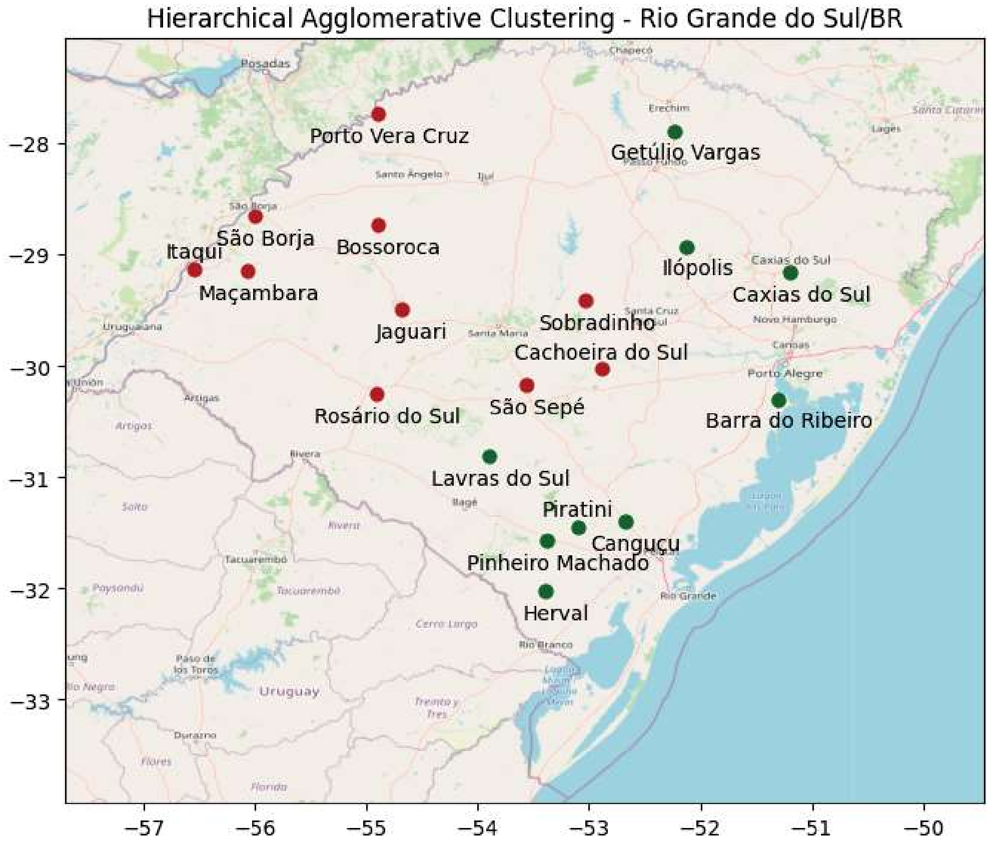


Figure 28 – SIMAGRO-RS Hiearchical Agglomerative Clustering with $k = 2$ (red and green clusters).

Regarding the clustering itself, the results were quite similar, grouping the $19$ stations into $2$ clusters, colored in the images in green and red colors. All three results are quite similar, differing slightly in the definition of the border between the clusters, in the central region of the state. For the red cluster, in the central and western regions of the state, we have a well-defined part of this cluster, with the meteorological stations of `Itaqui`, `Maçambara`, `São Borja`, `Porto Vera Cruz`, `Bossoroca`, `Rosário do Sul`, `Jaguari`, `São Sepé`, and `Cachoeira do Sul`. On the other hand, the well-defined part of the green cluster, in the east and south regions, has the `Getúlio Vargas`, `Ilópolis`, `Caxias do Sul`, `Barra do Ribeiro`, `Lavras do Sul`, `Piratini`, `Canguçu`, `Pinheiro Machado`, and `Herval` meteorological stations. Meanwhile, the most difficult stations to cluster were `Sobradinho`, near to the center of the state. While K-means and K-medoids algorithms placed this station in the east cluster (green), hierarchical clustering placed this station in the west cluster (red).

Furthermore, a relevant contribution of Hierarchical Clustering is that the resulting dendrogram shows the relationships between weather stations, as shown in Figure 29, highlighting the strongest similarities in climate patterns. Despite the station being placed in a different cluster (*Sobradinho*), which would deserve a more in-depth anal-

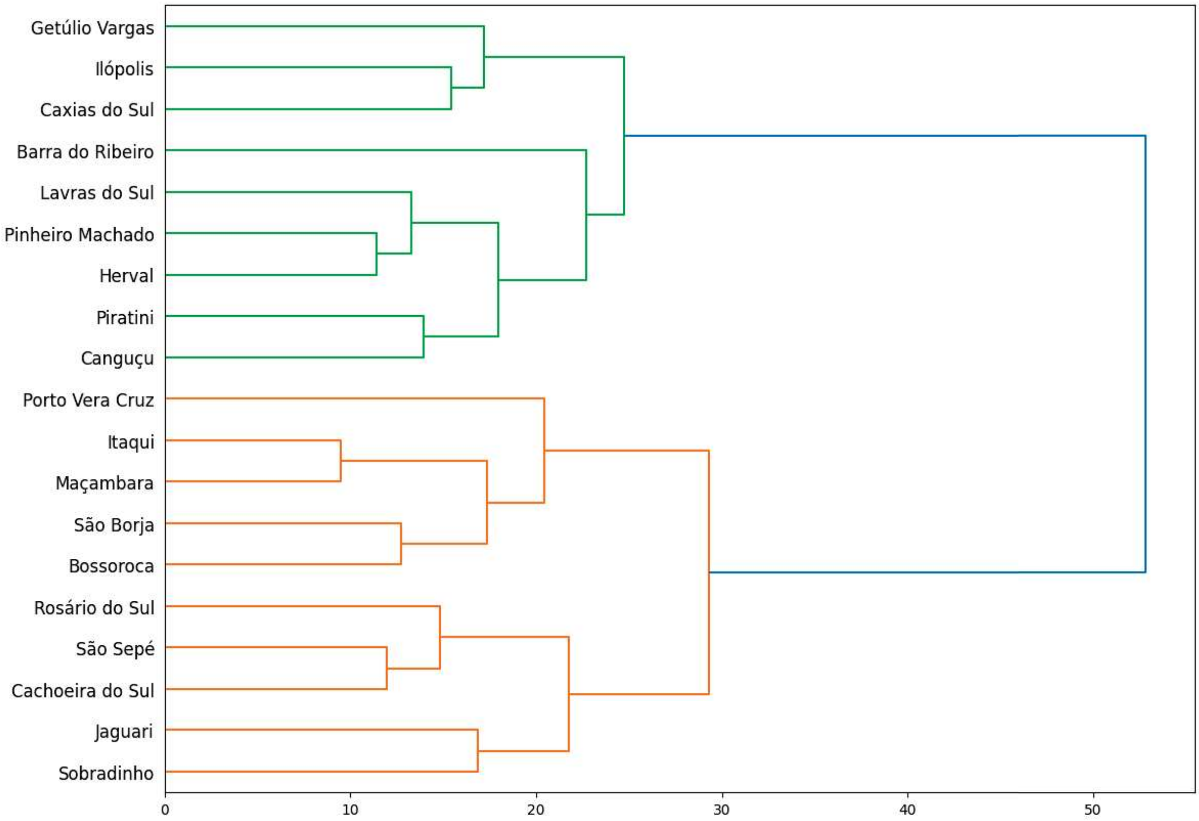


Figure 29 – HAC Dendogram in the SIMAGRO-RS dataset.

ysis, for the other stations it is possible to observe the strongest relationships between locations, which can provide very interesting information for agriculture. For example, for three practically equidistant stations, `Itaqui`, `Macambara`, and `São Borja`, the dendrogram establishes an order of priority in verifying climatic similarities, where `Itaqui` and `Macambara` have a stronger relationship, while `São Borja` has a more direct relationship with the `Bossoroca` station, further away than the others two. In an situation where there is no clear definition of the climate for a location or where there are no meteorological stations, this dendrogram can serve as a guide to where to look for similar conditions, helping to form the basis for making a decision. More general discussions on ways to interpret these results, as well as limitations, are presented at the end of this chapter.

### 6.1.5 RS Clusters Validation

To validate and evaluate the results obtained from clustering in the SIMAGRO-RS dataset, the metric specifically developed for this framework was used, as detailed in the previous chapter. Using the "*leave-one-out*" strategy, one station was removed at a time and this station variables were predicted based on the other stations in its cluster. The average RMSE of the prediction was then calculated for each station, then for each cluster, and, finally, the average RMSE of the algorithm. Table 8 presents the result of this metric for each station, cluster and algorithm, as well as the Total RMSE.

Table 8 – RMSE of cluster-based prediction in the SIMAGRO-RS dataset.

| | **Predicition RMSE** | **K-means** | **K-medoids** | **Hierarc. Clust.** |
|---|---|---|---|---|
| Cluster 1 | Station 1 | 0.385176 | 0.385176 | 0.392722 |
| | Station 2 | 0.458196 | 0.458196 | 0.327842 |
| | Station 3 | 0.533168 | 0.533168 | 0.346821 |
| | Station 4 | 0.515136 | 0.515136 | 0.465508 |
| | Station 5 | 0.788411 | 0.788411 | 0.366784 |
| | Station 6 | 0.474751 | 0.474751 | 0.442053 |
| | Station 7 | 0.441391 | 0.441391 | 0.515610 |
| | Station 8 | 0.443245 | 0.443245 | 0.553216 |
| | Station 9 | 0.602174 | 0.602174 | 0.634107 |
| | Station 10 | 0.650358 | 0.650358 | 0.556093 |
| | *C1 RMSE* | *0.529200* | *0.529200* | *0.460076* |
| Cluster 2 | Station 1 | 0.376194 | 0.376194 | 0.382269 |
| | Station 2 | 0.316325 | 0.316325 | 0.460803 |
| | Station 3 | 0.334933 | 0.334933 | 0.514188 |
| | Station 4 | 0.459615 | 0.459615 | 0.525793 |
| | Station 5 | 0.376683 | 0.376683 | 0.763422 |
| | Station 6 | 0.433474 | 0.433474 | 0.486197 |
| | Station 7 | 0.499726 | 0.499726 | 0.431048 |
| | Station 8 | 0.573455 | 0.573455 | 0.428163 |
| | Station 9 | 0.608885 | 0.608885 | 0.610335 |
| | *C2 RMSE* | *0.442143* | *0.442143* | *0.511357* |
| | **Total RMSE** | **0.485672** | **0.485672** | **0.485717** |

From these very similar performances, it is observed that K-means and K-medoids had a slight advantage in relation to this metric. Given that the clustering result of these two algorithms was the same, it was expected that the performance would also be the same.

It is possible to observe that both the three clustering follows a spatial pattern, grouping adjacent stations, and establishing the boundaries between one cluster and another. This is a behavior that makes sense when it comes to weather patterns. One cluster encompasses stations in the center and west of the state (red), and another encompasses the eastern and southern regions (green). This division was valited comparing to the traditional division of microregions in the state of Rio Grande do Sul (GOVERNO DO ESTADO DO RIO GRANDE DO SUL, 2021), as shown in Figure 30. This clustering respects traditional climatic regions, mainly observed in the center and west of the state, while merges stations in other regions, such as the north, east and south. This merge on right side of the map happens possibly due to the low number of variables and the few years observed. What appears to be the main contribution of this clustering, compared to the traditional division of microregions in RS, is a grouping of

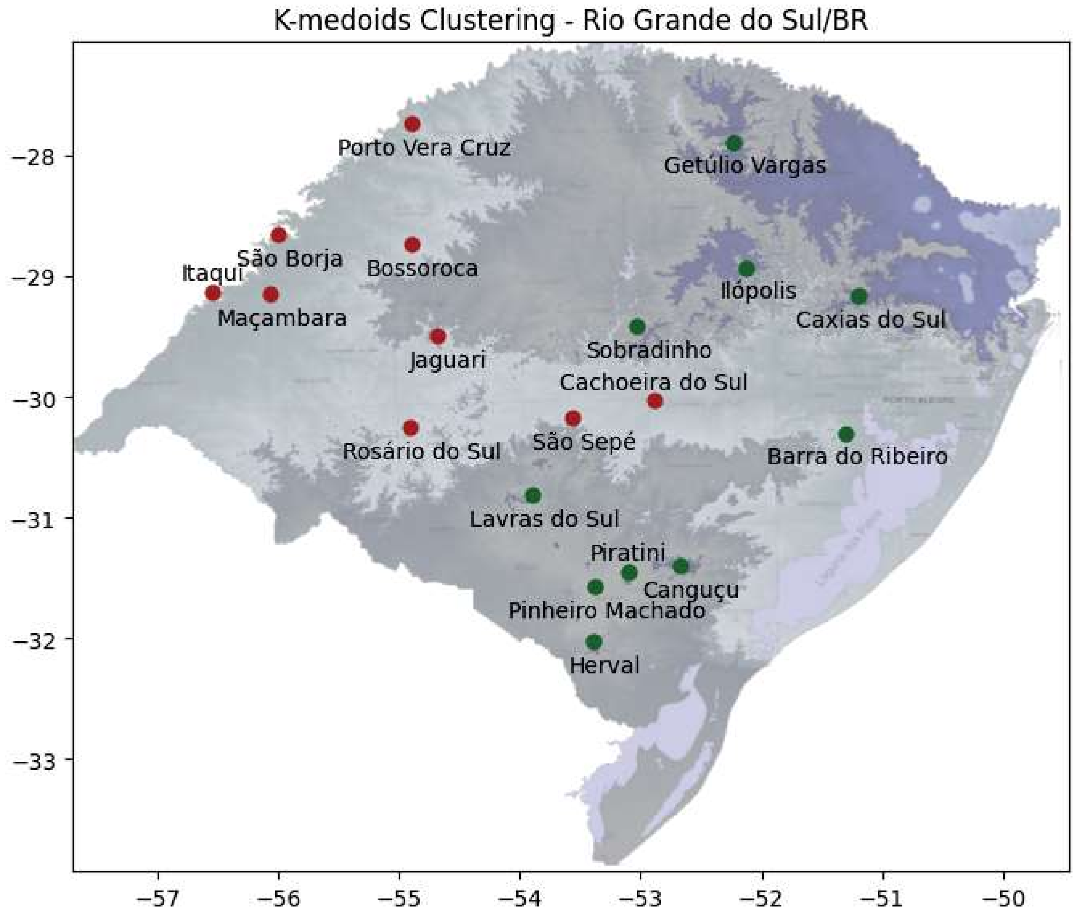


Figure 30 – Result of clustering on the map with background containing regional variations in annual temperature in Rio Grande do Sul.

microregions into larger groups and the movement of the limits of these regions, adjusting the climatic zoning according to the meteorological data of the last three years. The caption with temperature variations is not specifically included in this image, it is just an example to illustrate the climate variations in the state. There is no official climate microregionalization of the state, there are maps that compose the climate behavior, as presented in the Socioeconomic Atlas, from which the analyzes are derived.

## 6.2 Case Study II - AgroClimate/FAWN

This second case study was developed from a research collaboration with the AgroClimate laboratory, at the University of Florida, which deals with climate indicators for agriculture, such as disease alert systems, as reported below.

### 6.2.1 AgroClimate Disease Alert Systems

A Disease Alert System (DAS) in agriculture is a technology or network that provides timely and relevant information to farmers and agricultural stakeholders about the occurrence and spread of plant diseases, pests, or other threats to crops (GENT et al., 2013). These systems are designed to help farmers make informed decisions and take proactive measures to protect their crops and minimize losses. Data is be collected from different sources (weather stations, sensors, satellite imagery) and analyzed to identify patterns and trends. Statistics and ML techniques may be employed to recognize disease-friendly conditions and provide early warning of potential disease outbreaks based on historical data and environmental conditions. Then, the DAS generates alerts and notifications to be sent to farmers via communication channels, such as SMS, email, mobile apps, or web platforms.

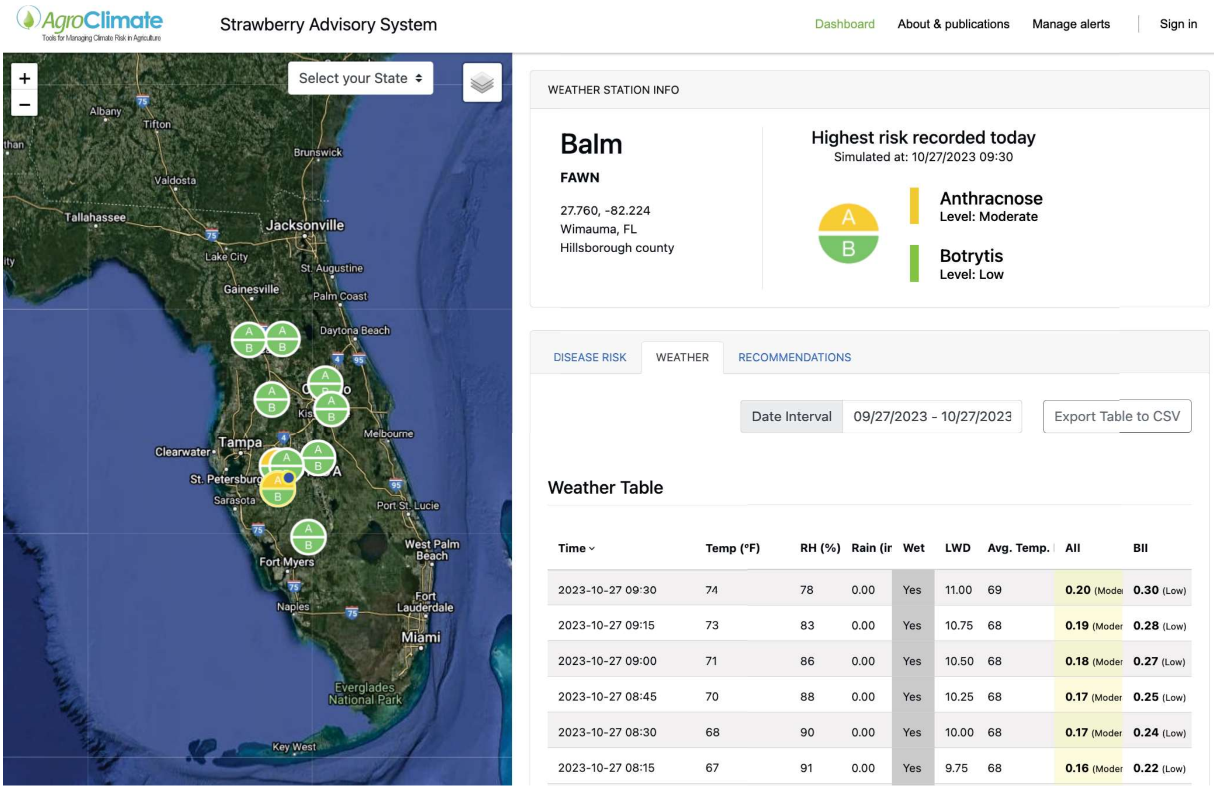
Also, the DAS can provide farmers with recommendations and best practices for managing and controlling the identified threats. This may include guidance on pesticide or fungicide application, or other preventive measures. Weather conditions can significantly influence the spread of diseases and pests, which is why many disease alert systems integrate weather data to provide more accurate predictions and recommendations. These systems also facilitate collaboration and data sharing among farmers, agricultural extension services, research institutions, and government agencies to enhance disease monitoring and control efforts. Overall, since they have real-time or near-real-time information, it enables quick response to emerging threats and contribute to more sustainable and efficient agricultural practices, improving crop management and reducing the economic and environmental impact of diseases and pests.

Disease alert systems find application at different scales, ranging from local and regional to national and even international levels. In the United States, a relevant example of such systems is *AgroClimate*[2], which offers valuable support to farmers, with a particular focus on the state of Florida. AgroClimate encompasses various tools designed to help farmers make informed decisions and manage disease risks effectively. Two key components of the AgroClimate system are the Strawberry Advisory System (StAS) and the Blueberry Advisory System (BAS), both focused on monitoring disease-friendly conditions and helping to manage crops in Florida. These systems address specific disease threats, helping farmers mitigate its impact on strawberries and blueberries crops.

The StAS issues timely warnings related to anthracnose and botrytis fruit rot in strawberry fields. These diseases can cause considerable damage to strawberry crops, reducing yield and quality. Similarly, the BAS is primarily focused on alerting farmers to the presence of anthracnose fruit rot, a common and potentially devastating disease in blueberries. By continuously monitoring environmental conditions, the risk of disease is classified as low, moderate or high for each location, generating alerts and recommendations that empower growers, enabling them to implement preventive measures, such as fungicide applications in response to the disease risks.

Figure 31 shows the StAS interface, where each pin on the map represents a

[2] AgroClimate Project - `https://agroclimate.com/`



WEATHER STATION INFO

**Balm**

**FAWN**

27.760, -82.224<br>
Wimauma, FL<br>
Hillsborough county

Highest risk recorded today
Simulated at: 10/27/2023 09:30

**Anthracnose**
Level: Moderate

**Botrytis**
Level: Low

Date Interval 09/27/2023 - 10/27/2023

Weather Table

| Time | Temp (°F) | RH (%) | Rain (ir | Wet | LWD | Avg. Temp. | AII | BII |
|---|---|---|---|---|---|---|---|---|
| 2023-10-27 09:30 | 74 | 78 | 0.00 | Yes | 11.00 | 69 | **0.20** (Mode | **0.30** (Low) |
| 2023-10-27 09:15 | 73 | 83 | 0.00 | Yes | 10.75 | 68 | **0.19** (Moder | **0.28** (Low) |
| 2023-10-27 09:00 | 71 | 86 | 0.00 | Yes | 10.50 | 68 | **0.18** (Moder | **0.27** (Low) |
| 2023-10-27 08:45 | 70 | 88 | 0.00 | Yes | 10.25 | 68 | **0.17** (Moder | **0.25** (Low) |
| 2023-10-27 08:30 | 68 | 90 | 0.00 | Yes | 10.00 | 68 | **0.17** (Moder | **0.24** (Low) |
| 2023-10-27 08:15 | 67 | 91 | 0.00 | Yes | 9.75 | 68 | **0.16** (Moder | **0.22** (Low) |

Figure 31 – Example of the Strawberry Advisory System (StAS) interface, one of AgroClimate's tools.

weather station in Florida. When selecting a station on the map (in this example, Balm, in west-central Florida), data about disease risk are displayed on the right side of the interface. At the top of the right side information about the station is displayed and the circle with the letters $A$ and $B$ indicate the risk of Anthracnose and Botrytis, respectively. The colors indicate the risk level (green for low, yellow for moderate and red for high). Also, previous measurements (`Weather` tab, selected in this image) are displayed, as well as the current and the estimated risk for diseases in the next 48 hours (`Disease Risk` tab), and recomentations for fungicide application (`Recomendations` tab). Currently, 10 weather stations are available for StAS, those shown in the figure (`Apopka`, `Arcadia`, `Balm`, `Bronson`, `Citra`, `Dover`, `Floral City`, `Lake Alfred`, `Plant City`, and `Umatilla`) and 15 for BAS (`Alachua`, `Apopka`, `Arcadia`, `Balm`, `Bronson`, `Citra`, `Dade City`, `Dover`, `Floral City`, `Jay`, `Lake Alfred`, `Live Oak`, `Ona`, `Putnam Hall`, and `Sebring`).

As these diseases are caused by fungi, risk assessment depends on climatic variables such as temperature, relative humidity, precipitation and solar radiation. In this context, challenges arise when neighboring stations present different levels of risk. These situations can happen due to several factors, including abrupt climate fluctuations, localized rainfall or problems related to observations, mainly arising from sensor hardware malfunction or communication errors between the station and the server.

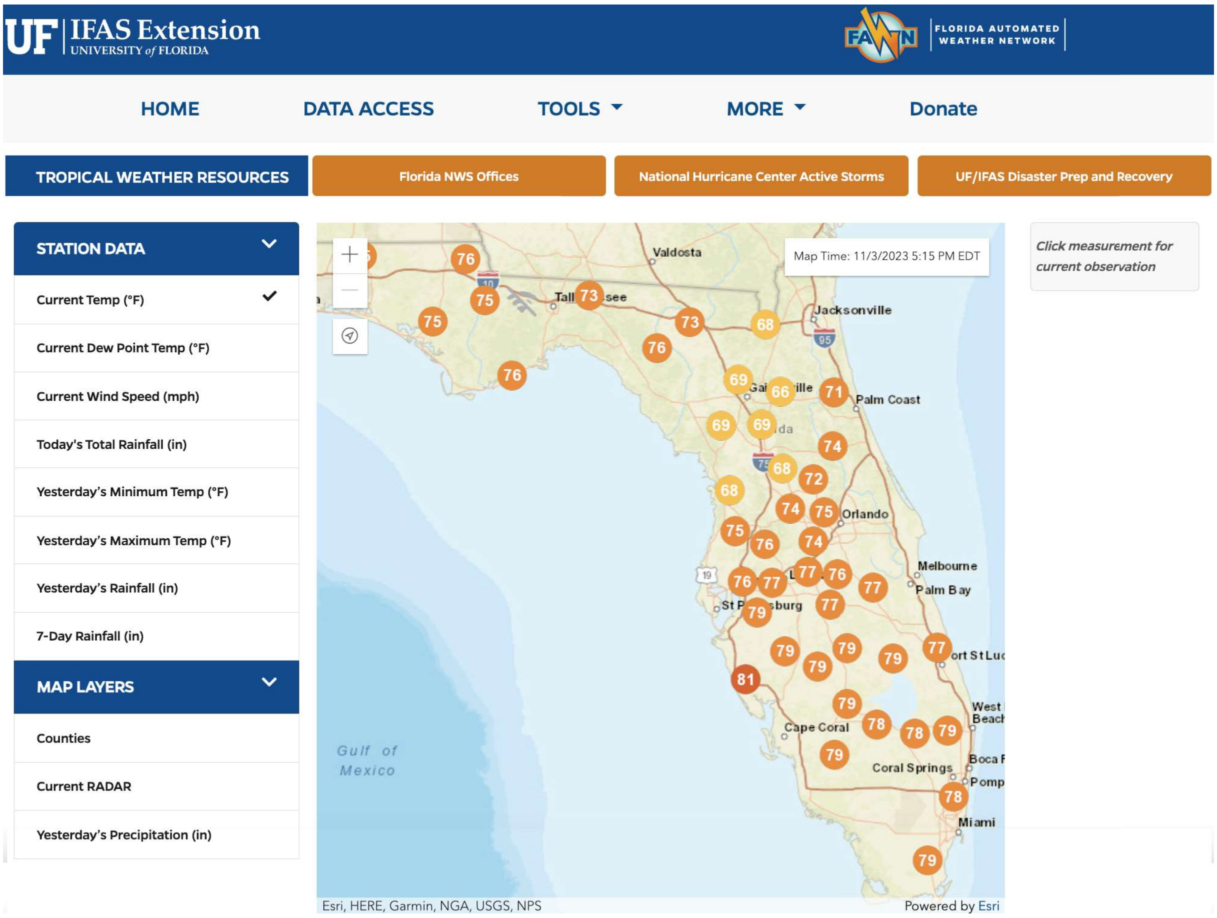

UF | IFAS Extension
UNIVERSITY of FLORIDA
FAWN FLORIDA AUTOMATED WEATHER NETWORK
HOME
DATA ACCESS
TOOLS
MORE
Donate
TROPICAL WEATHER RESOURCES
Florida NWS Offices
National Hurricane Center Active Storms
UF/IFAS Disaster Prep and Recovery
STATION DATA
Current Temp (°F) ✓
Current Dew Point Temp (°F)
Current Wind Speed (mph)
Today's Total Rainfall (in)
Yesterday's Minimum Temp (°F)
Yesterday's Maximum Temp (°F)
Yesterday's Rainfall (in)
7-Day Rainfall (in)
MAP LAYERS
Counties
Current RADAR
Yesterday's Precipitation (in)
Map Time: 11/3/2023 5:15 PM EDT
Click measurement for current observation
Valdosta
Jacksonville
Palm Coast
Orlando
Melbourne
Palm Bay
Cape Coral
Coral Springs
Miami
Gulf of Mexico
Esri, HERE, Garmin, NGA, USGS, NPS
Powered by Esri

Figure 32 – Example of the FAWN interface with weather stations in Florida/US.

### 6.2.2 FAWN Dataset

Since the conditions for the occurrence of agricultural diseases, such as those monitored by AgroClimate, are based on data from meteorological observations, it is important to have a reliable data source for this task. Therefore, the data used in AgroClimate's StAS and BAS uses data from the *Florida Automated Weather Network (FAWN)*[3], an initiative of the University of Florida Institute of Food and Agricultural Sciences (UF/IFAS) which maintains a network of more than 45 weather stations throughout the state of Florida. Figure 32 shows the FAWN online platform interface, with a map of weather stations spread across the state of Florida.

The primary goal of FAWN is to provide reliable climate data to guide agricultural decision-making and the management of natural, human and financial resources. FAWN stations utilize standardized equipment to ensure consistency in measurements across various locations. This standardization is critical for maintaining data quality, as it assures that data from different sources is generated in a consistent manner. Nevertheless, deviations or discrepancies in measurements may still occur, as the equipment's performance can differ in each station due to exposure to unforeseeable events.

[3] Florida Automated Weather Network - `https://fawn.ifas.ufl.edu`

The weather stations collect multiple meteorological variables every 15 minutes, and all data produced is stored in the FAWN database, which is public and accessible online. AgroClimate systems consume these data via the FAWN applet and update risk indicators at the same frequency as measurements. Additionally, data retrieval is feasible by downloading .csv files, allowing data extraction by year and weather station. As this case study was developed from the perspective of AgroClimate's DAS (StAS and BAS) it is interesting to note that not all weather stations were of interest, since strawberry and blueberry production only occurs in some regions and not throughout the state of Florida. Currently, data from 10 weather stations are used for StAS and 15 weather stations for BAS. However, as climate data is available for the entire state, in this case, it is understood that the use of data from the largest number of stations possible would contribute to greater reliability and precision regarding the state's climate zoning, therefore, data from the $29$ stations listed above were used.

At the time of this research, FAWN has $47$ weather stations in operation and all existing data in the FAWN database was exported, for all existing stations, from December 1997 (first records) until August 2023. The export was through the FAWN web platform, downloading files in .csv format, with one file for each year. The extraction resulted in a $51.1MB$ dataset. When extracting data, the existence of data was verified for each location and each year, from 1997 to 2023. Since the weather stations were not all implemented at the same time, it was expected that some stations would have more data than others. It was observed that during the first years of FAWN's operation, there were very few stations in use, which is natural at the beginning of the implementation of these systems. Besides, some stations operated for a short time, then were deactivated. Others were implemented recently, with little data volume, which is insufficient for climate analysis. These stations were then excluded from the data set, as they did not have representative data for some locations.

Finally, it was observed that $29$ stations, well distributed across the state, as shown in Figure 33, have been operating regularly since 2003, which represented a good volume of data and a reasonable time span for the climatic research. For better visualization purposes and not to overload the graph with textual information, in this image the meteorological stations are referenced by the identifier code (ID), instead of the name. In addition to the ID, other basic information about the meteorological stations used in this research, such as name, county and geolocation, are presented in Table 9. Thus, data from these stations were selected, resulting in a data set with meteorological observations of the variables of interest (T, RH, RF and SR) sensed during the last twenty years (2003-2023).

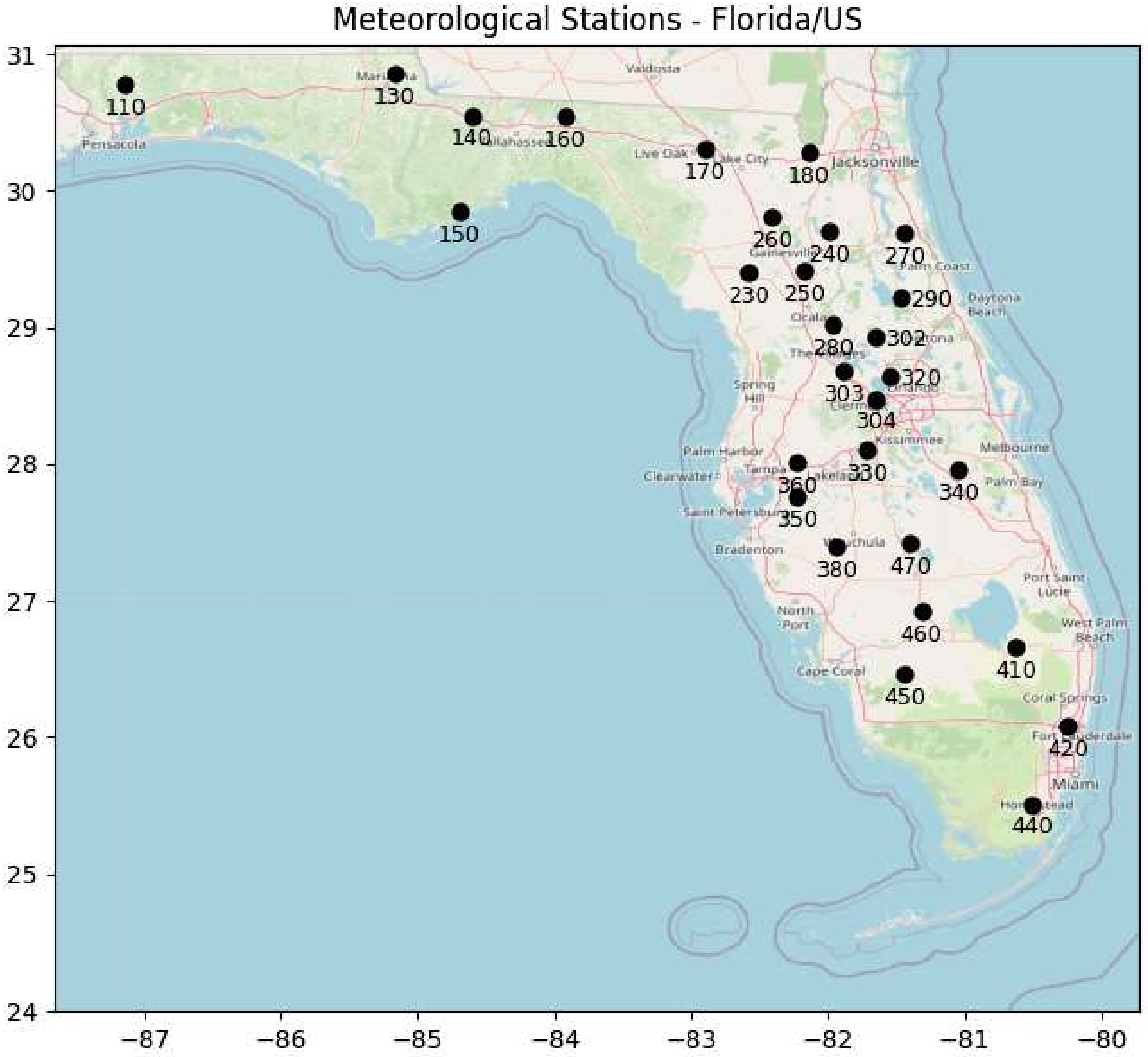


Figure 33 – Spatial distribution of FAWN meteorological stations used in this case study.

### 6.2.3 Data Preprocessing

From the extracted dataset, the next step is its preprocessing, to make it more suitable for data analysis. In this research, two preprocessing steps were carried out, filtering and quality checking, in order to select variables and time periods of interest, as well as ensuring data quality. Both steps are detailed below.

#### *6.2.3.1 Filtering*

The filtering stage aims to select the variables of interest and the time period to be analyzed, from the research perspective. Given the agricultural context of DAS, this phase is generally related to some agricultural activity, such as planting, harvesting or spraying inputs, and the period of time in which this activity occurs. Both StAS and BAS, as previously detailed, focus on monitoring climatic conditions favorable to diseases caused by fungi, estimating the disease risk, in order to alert producers to

| ID | Station Name | County | Lat. (Deg) | Long. (Deg) |
|---|---|---|---|---|
| 110 | Jay | Santa Rosa | 30.77516 | -87.14015 |
| 130 | Marianna | Gadsden | 30.85000 | -85.16516 |
| 140 | Quincy | Jackson | 30.54581 | -84.59898 |
| 150 | Carrabelle | Franklin | 29.84240 | -84.69511 |
| 160 | Monticello | Jefferson | 30.53570 | -83.91760 |
| 170 | Live Oak | Suwanee | 30.30500 | -82.89876 |
| 180 | Macclenny | Baker | 30.28148 | -82.13798 |
| 230 | Bronson | Levy | 29.40038 | -82.58611 |
| 240 | Putnam Hall | Putnam | 29.69700 | -81.98600 |
| 250 | Citra | Marion | 29.41010 | -82.17320 |
| 260 | Alachua | Alachua | 29.80266 | -82.41081 |
| 270 | Hastings | St. John's | 29.69332 | -81.44485 |
| 280 | Ocklawaha | Marion | 29.02033 | -81.96896 |
| 290 | Pierson | Volusia | 29.21717 | -81.46065 |
| 302 | Umatilla | Lake | 28.92655 | -81.65297 |
| 303 | Okahumpka | Lake | 28.68165 | -81.88565 |
| 304 | Avalon | Orange | 28.47485 | -81.65300 |
| 320 | Apopka | Orange | 28.63771 | -81.54675 |
| 330 | Lake Alfred | Polk | 28.10185 | -81.71128 |
| 340 | Kenansville | Osceola | 27.96221 | -81.05123 |
| 350 | Balm | Hillsborough | 27.75998 | -82.22410 |
| 360 | Dover | Hillsborough | 28.01510 | -82.23254 |
| 380 | Ona | Hardee | 27.39750 | -81.93973 |
| 410 | Belle Glade | Palm Beach | 26.65678 | -80.63001 |
| 420 | Ft. Lauderdale | Broward | 26.08530 | -80.24050 |
| 440 | Homestead | Dade | 25.51260 | -80.50310 |
| 450 | Immokalee | Collier | 26.46225 | -81.44033 |
| 460 | Palmdale | Glades | 26.92480 | -81.31455 |
| 470 | Sebring | Highlands | 27.42108 | -81.40095 |

Table 9 – FAWN Meteorological Stations used in this case study.

spray fungicide. Therefore, filtering in this case starts from the variables that are related to these climatic conditions favorable to diseases, which are: *Temperature (T)*, *Relative Humidity (RH)*, *Rainfall (RF)* and *Solar Radiation (SR)*.

For this second case study, this four climatic variables were chosen again, as in the first case study, but now for different reasons. These four variables are those most directly linked to climatic conditions favorable to the occurrence of fungal diseases, which develop in environments with excessive humidity. Temperature affects the growth and activity of plant pathogens and their vectors, while humidity levels influence the germination of fungal spores and the spread of diseases like powdery mildew and downy mildew. Also, excessive or prolonged rainfall can create waterlogged conditions in the soil, increasing the risk of root diseases, and solar radiation influences plant photosynthesis and plant stress levels.

Thus, after selecting the variables, filtering was done by the agricultural period of interest, which in this case was the entire annual production season of these fruits, the period in which diseases occur and, therefore, the systems are used. The strawberry production in Florida is between November and February and the blueberry production season is between December and May. Given the similarity and intersection between the production periods of both fruits, in this research the two seasons were merged, and the data was extracted between November 1st and May 31st, which covers both seasons, totaling $212$ days, disregarding the extra day of leap years.

#### *6.2.3.2 Quality Checking*

The first step of quality checking performed in this research was a scan for missing values. Data gaps are caused by various reasons, such as sensor malfunctions, hardware degradation, configuration errors, erroneous human inputs, or communication failures between the sensor devices and the central server. Depending on the specific circumstances surrounding each missing data point, corrective measures could be considered, or alternatively, data points may be removed from the dataset if their prevalence does not pose a substantial threat to the dataset's overall integrity. For the $29$ meteorological stations, only $367$ missing values were identified, for all meteorological variables together, T, RH, RF and SR, which represents $0.2\%$ of the dataset. Regarding null values, $1,342$ null measurements were identified between the four variables, representing $1\%$ of the dataset. Therefore, given the high-frequency data collection and the possibility of using aggregated representative values for these four variables, it was decided to overlook missing values due to their low proportion in relation to the whole dataset.

Furthermore, an outlier analysis was carried out in order to identify discrepant values with the distribution of each variable. These outliers possibly represent measurement errors, such as sensor failures or failures in sending data to the server. After checking for outliers, the following were identified and replaced, adding upper and lower outliers: $1,190$ outliers for T; $1,288$ for RH; $23,150$ for RF; and, $32$ for SR.

### 6.2.4 Feature Extraction

Feature extraction was performed on the FAWN dataset, in two stages: Aggregation and Standardization. When monitoring disease risk, as is the case with AgroClimate's DAS, the aggregation of meteorological data by day of the year helps to identify trends and patterns that are relevant for recommending fungicide application. In the same context, standardization helps to represent data on the same scale. The details of the aggregation and standardization in the FAWN dataset, used by the AgroClimate DAS, are presented below.

#### *6.2.4.1 Data Aggregation*

Considering the four variables under analysis in this research (T, RH, RF and SR), the initial dataset contains records of their daily values, over the last twenty years. From the perspective of the production season of both strawberries and blueberries, it is feasible and interesting to aggregate these data into DOY average values, as described previously in the framework methodology. For example, November 1st, the first day of the season under analysis, is day 305 of the year. For aggregation, the values on day 305 of all years under analysis (2003-2023) were considered, for T, RH, RF and SR. Then, this process resulted in an aggregated dataset containing $6,148$ values for each variable ($212$ days of the season multiplied by the $29$ weather stations). Figure 34 shows the result of the aggregation for the dataset of this case study, a graph for each variable, as in the previous case study.

#### *6.2.4.2 Standardization*

In the context of the climate data used in this research, the range of values for each variable is quite different. Thus, the standardization proposed in the framework was carried out, in order to transform the data and represent it on the same scale. Figure 35 shows, at the top, all the original variables plotted under the same vertical axis, in order to highlight the distance between them. At the bottom, its presented a new representation of the dataset, in a standardized version. The data was resized to a representation in which all four variables are plotted closely, under the same vertical axe. This resizing of the variable scale impacts the performance of the clustering algorithms, since the distance between the time series is normalized.

### 6.2.5 Clustering Results

This section presents the clustering results for the case study with the FAWN dataset. First, the search for the optimal number of clusters is presented, then the clustering itself and the evaluation of results.

#### *6.2.5.1 Number of Clusters*

As detailed before, in this case study were analyzed data from 29 meteorological stations. The optimal number of clusters was investigated between 2 and 10, in order to explore a reasonable range of possibilities. Then, the Elbow Plot and the Silhouette Score were used to define the $k$ value.

Figure 36 presents the result of the Elbow Plot for the K-means and K-medoids algorithms, in a similar way to the previous case study. Again, with this metric alone it was not possible to ensure the optimal number of clusters, because, even though K-means has a slight elbow in $k = 2$, for K-medoids there is no clear elbow. In fact, several variations are observed in the plotted curve, without it being possible to define

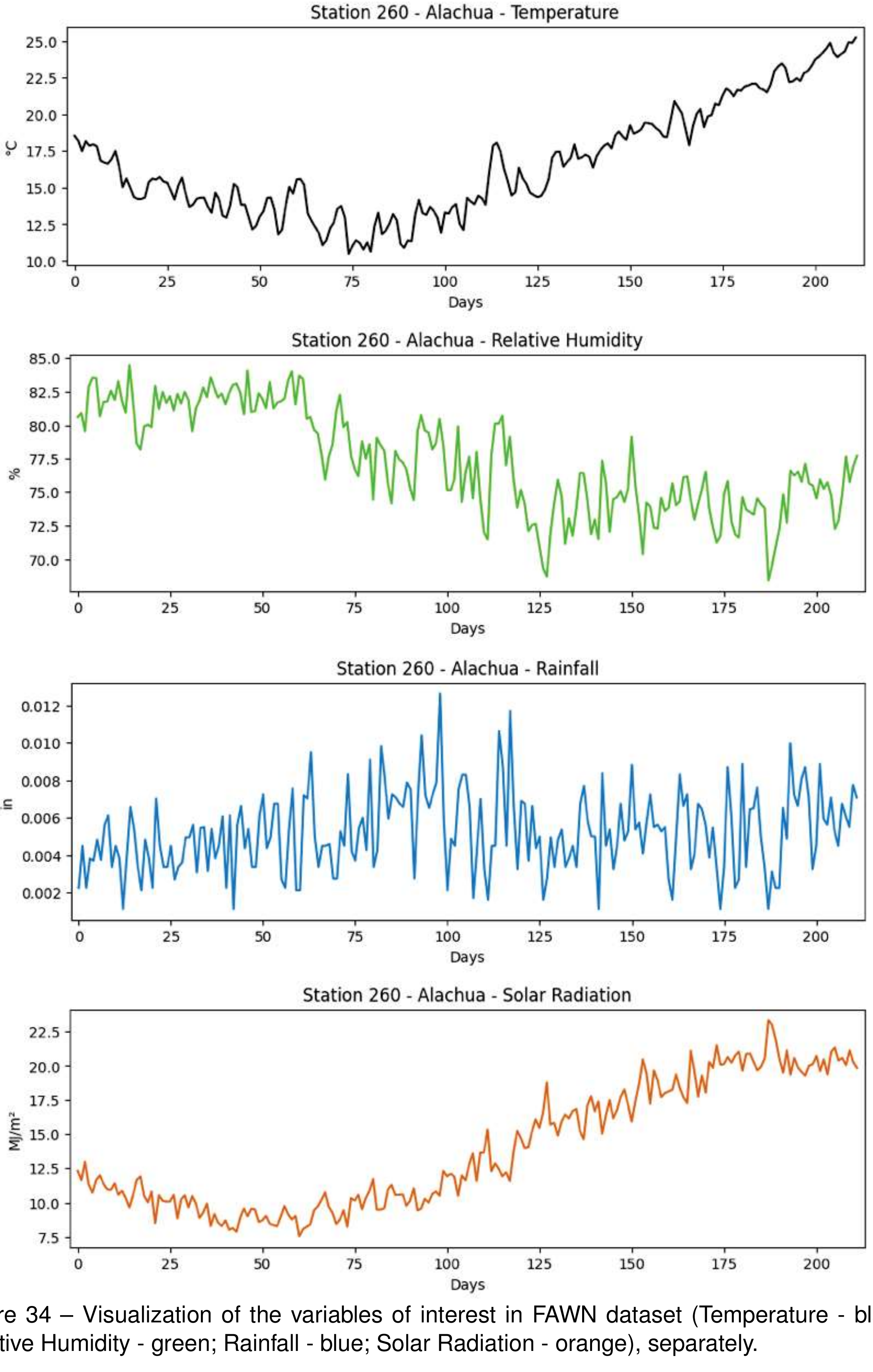


Figure 34 – Visualization of the variables of interest in FAWN dataset (Temperature - black; Relative Humidity - green; Rainfall - blue; Solar Radiation - orange), separately.

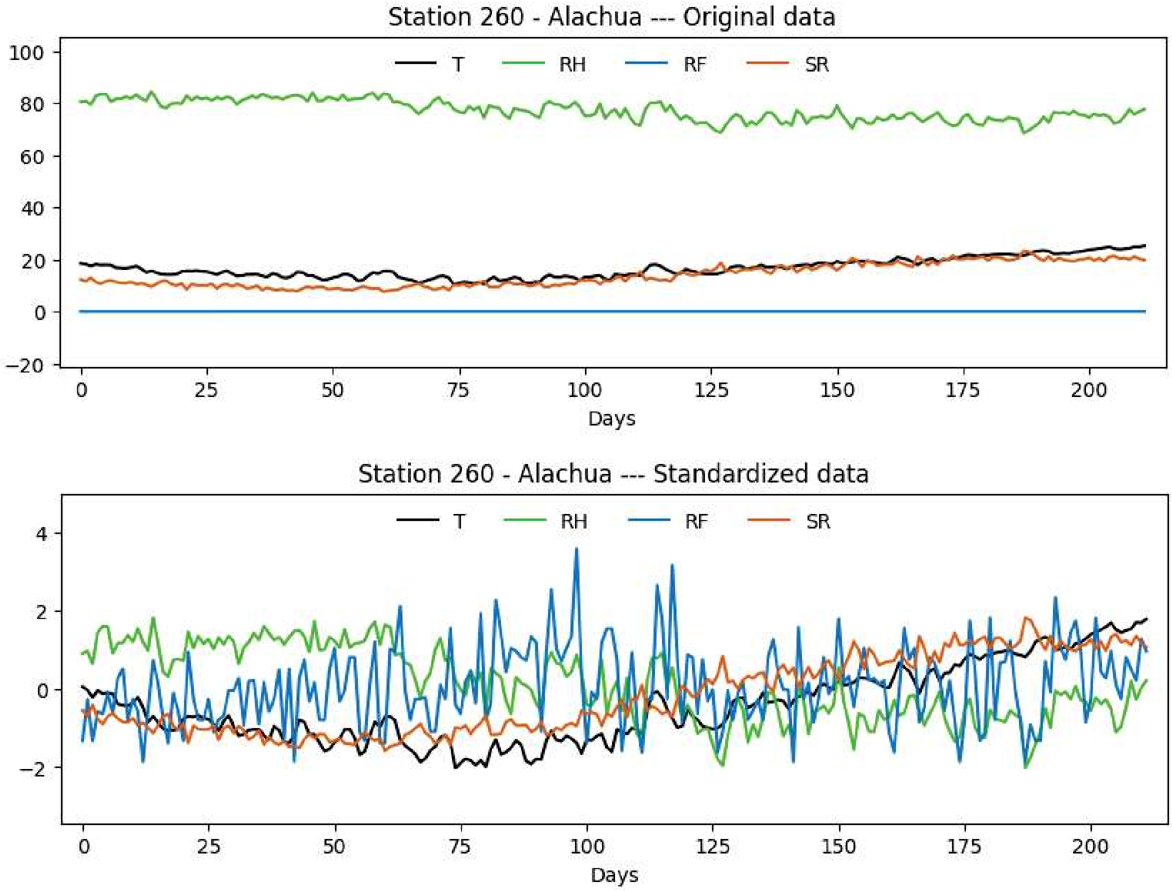


Figure 35 – Original (above) and standardized (below) time series of FAWN weather variables.

an optimal number of clusters.

The Silhouette Score was also calculated for all algorithms, as shown in Table 10. This coefficient, which varies between -1 and 1, assigns a higher score to better defined clusters. Based on these results, again analyzing the *Elbow Plot* and *Silhouette Score* together, we arrived at the ideal number of clusters $k = 2$. As in the other case study, it is worth noting that although the silhouette score points to an ideal number of clusters, higher scores (around $0.25$) indicate that the quality of the cluster is closer to the average $(0.0)$ than the maximum value $(1.0)$. This indicates that the data is challenging to cluster and corroborates the importance of the framework.

Given the characteristics of this case study and the unclear result of the two already used techniques, in order to improve the choice of an optimal number of clusters, it was decided to also use the RMSE prediction metric in this stage of choosing the number of clusters. Thus, the RMSE of the clusterizations was calculated for the three algorithms, varying the number of clusters from 2 to 7, as Florida is historically divided into seven climatic regions.

Table 11 shows the results of this metric for the FAWN dataset, summarizing the results of several runs with different clustering setups. The first double column (`Clusters (k)`) indicates the number of clusters for that run, along with a cluster identifier

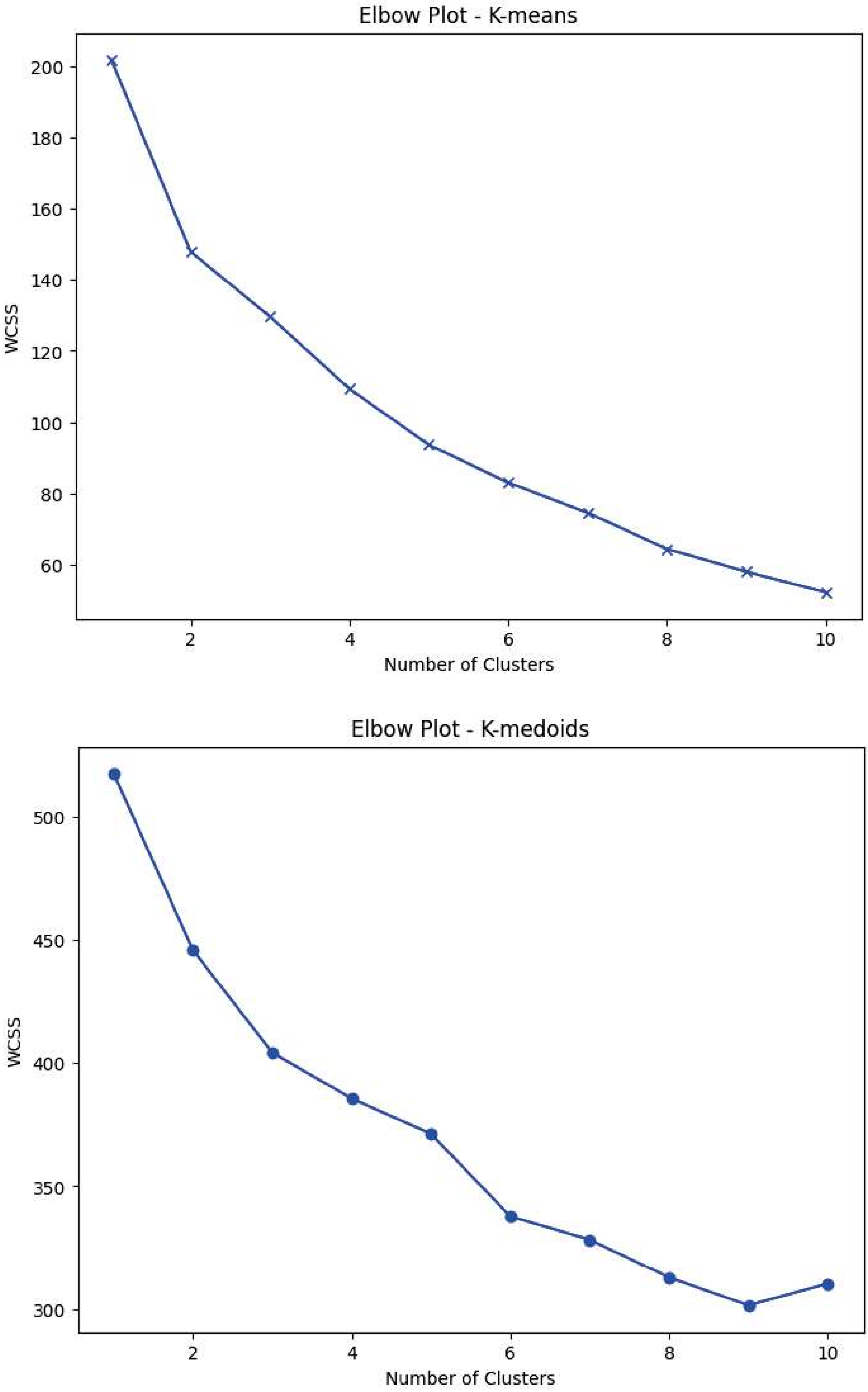


Figure 36 – Elbow Plots for the K-means and K-medoids algorithms on the FAWN dataset.

Table 10 – Silhoutte Score for FAWN dataset.

| N of clusters | K-means | K-medoids | Hierarc. Clust. |
|---|---|---|---|
| **2** | **0.219** | **0.247** | **0.221** |
| 3 | 0.197 | 0.153 | 0.182 |
| 4 | 0.149 | 0.072 | 0.195 |
| 5 | 0.165 | 0.039 | 0.195 |
| 6 | 0.140 | 0.056 | 0.175 |
| 7 | 0.166 | 0.049 | 0.170 |
| 8 | 0.147 | 0.027 | 0.158 |
| 9 | 0.124 | 0.026 | 0.137 |
| 10 | 0.112 | -0.043 | 0.131 |

$(C_1, ..., C_n)$. The other three double columns (`K-means`, `K-medoids` and `Hierarchical`) present the results of each algorithm for each run. Also, for each configuration, it is presented the number of stations in that cluster (`Station` subcolumn), the prediction RMSE by cluster (`RMSE` subcolumn), and the Total RMSE (lines in italics) by algorithm. The missing values in the table (marked with a dash) refer to clusters with only one station, where it is not possible to calculate the metric, disregarded from the calculation. The best value found is highlighted in bold, which means the cluster configuration with the lowest prediction error.

From this table it is possible to observe some interesting aspects of the clustering for this dataset. Initially, it is noted that for $k = 2$ the three algorithms perform in the same way, with the same arrangement of clusters and equal RMSE values. This implies that if we only used the two initial metrics to define k (Elbow and Silhouette), it would not make sense to explore different algorithms, which could hide important results. As the number of clusters increases, performance varies. As the number of clusters increases, performance varies. As the search here is focused on the lowest value of the prediction RMSE, the K-means clustering algorithm performed best with $k = 3$ ($RMSE = 0.363948$), while K-medoids with $k = 6$ (best overall performance, $RMSE = 0.338810$) and Hierarchical clustering with $k = 7$ ($RMSE = 0.347625$). Therefore, from here on, the ideal configuration for clustering was defined with $k = 6$ and K-medoids algorithm.

### *6.2.5.2 Clustering Algorithms Results*

After running the clustering algorithms, the results were consolidated as follows.The stations were grouped using algorithm K-medoids into six clusters, being C1 with 4 stations (Northwest), C2 with 8 stations (North), C3 with 4 stations (North Center), C4 with 3 stations (South Center), C5 with 2 stations (West Center) and C6 with 8 meteorological stations (South). Figure 37 presents the distribution of the clusters

Table 11 – RMSE of cluster-based prediction in the FAWN dataset.

| **Clusters (*k*)** | | **K-means** | | **K-medoids** | | **Hierarchical** | |
|---|---|---|---|---|---|---|---|
| | | **Stations** | **RMSE** | **Stations** | **RMSE** | **Stations** | **RMSE** |
| k=2 | C1 | 13 | 0.346243 | 13 | 0.346243 | 13 | 0.346243 |
| | C2 | 16 | 0.383892 | 16 | 0.383892 | 16 | 0.383892 |
| | *Total* | | *0.365067* | | *0.365067* | | *0.365067* |
| k=3 | C1 | 8 | 0.420681 | 8 | 0.307635 | 6 | 0.403291 |
| | C2 | 11 | 0.331649 | 13 | 0.346243 | 13 | 0.346243 |
| | C3 | 10 | 0.339514 | 8 | 0.395442 | 10 | 0.339514 |
| | *Total* | | *0.363948* | | *0.349773* | | *0.363016* |
| k=4 | C1 | 10 | 0.323647 | 12 | 0.404089 | 10 | 0.323647 |
| | C2 | 6 | 0.403291 | 5 | 0.325341 | 6 | 0.403291 |
| | C3 | 10 | 0.339514 | 7 | 0.413878 | 10 | 0.339514 |
| | C4 | 3 | 0.415971 | 5 | 0.317309 | 3 | 0.415971 |
| | *Total* | | *0.370606* | | *0.352055* | | *0.370606* |
| k=5 | C1 | 7 | 0.413878 | 12 | 0.404089 | 10 | 0.323647 |
| | C2 | 3 | 0.383048 | 4 | 0.331113 | 3 | 0.415971 |
| | C3 | 9 | 0.321412 | 3 | 0.271684 | 9 | 0.321412 |
| | C4 | 3 | 0.415971 | 2 | 0.360668 | 1 | - |
| | C5 | 7 | 0.319850 | 8 | 0.395442 | 6 | 0.403291 |
| | *Total* | | *0.370832* | | *0.342119* | | *0.363469* |
| k=6 | C1 | 7 | 0.310207 | 4 | 0.331113 | 9 | 0.321412 |
| | C2 | 4 | 0.359697 | 8 | 0.326411 | 4 | 0.347587 |
| | C3 | 6 | 0.403291 | 4 | 0.347587 | 6 | 0.305315 |
| | C4 | 2 | 0.502882 | 3 | 0.271684 | 1 | - |
| | C5 | 4 | 0.347587 | 2 | 0.360668 | 3 | 0.415971 |
| | C6 | 6 | 0.323648 | 8 | 0.395442 | 6 | 0.403291 |
| | *Total* | | *0.374552* | | ***0.338810*** | | *0.358715* |
| k=7 | C1 | 5 | 0.327884 | 5 | 0.376115 | 1 | - |
| | C2 | 6 | 0.305315 | 6 | 0.305315 | 6 | 0.305315 |
| | C3 | 3 | 0.415971 | 3 | 0.312527 | 1 | - |
| | C4 | 4 | 0.401165 | 3 | 0.271684 | 4 | 0.347587 |
| | C5 | 2 | 0.488984 | 2 | 0.360668 | 2 | 0.360520 |
| | C6 | 4 | 0.347587 | 2 | 0.400007 | 6 | 0.403291 |
| | C7 | 5 | 0.348975 | 8 | 0.395442 | 9 | 0.321412 |
| | *Total* | | *0.376554* | | *0.345965* | | *0.347625* |

graphically on the map of Florida, coloring the clusters with different colors: orange, dark blue, light green, yellow, violet, and gray, respectively from C1 to C6.

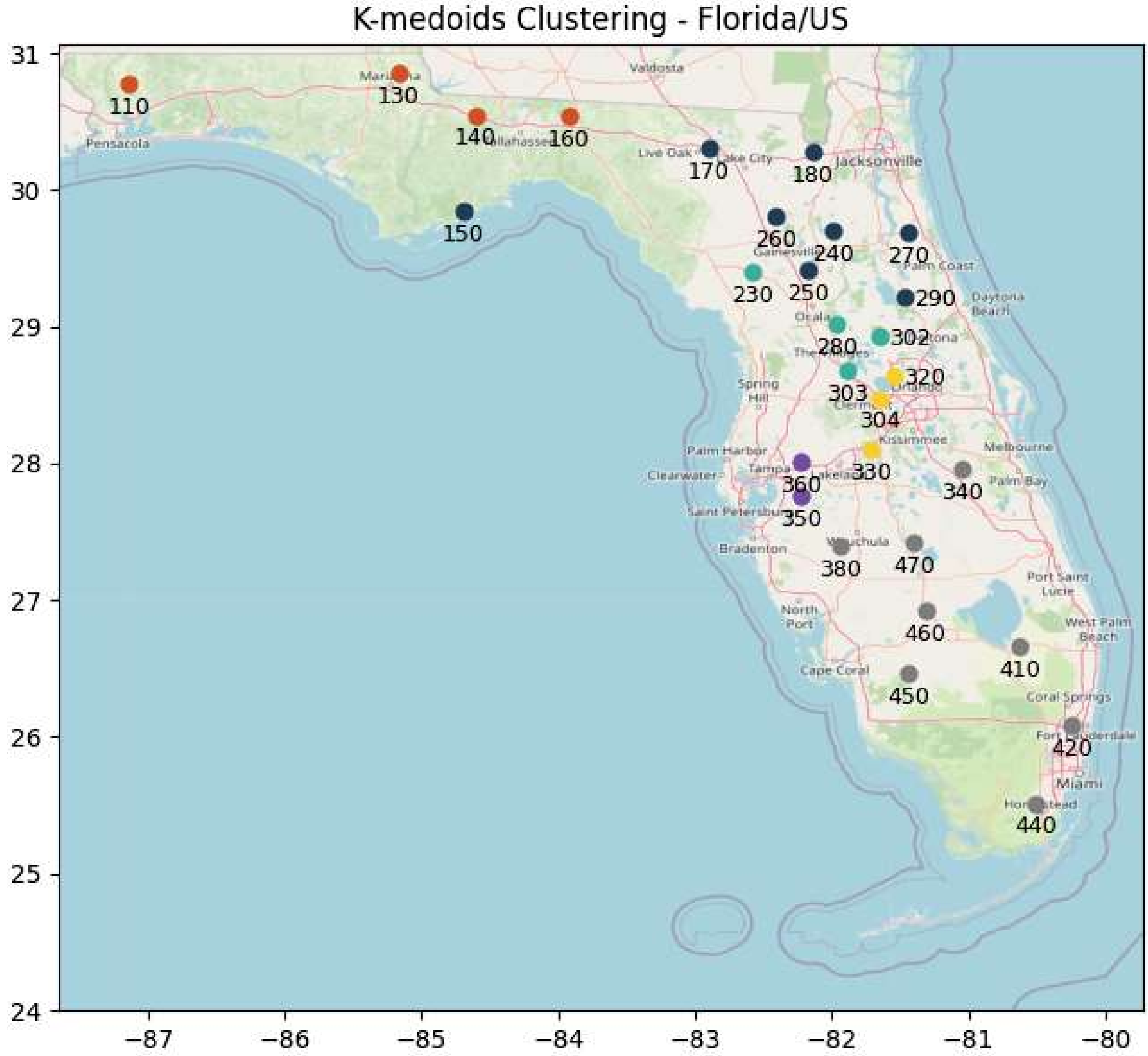


Figure 37 – FAWN K-medoids Clustering with $k = 6$ (orange, dark blue, light green, gray, and yellow clusters).

Also, Table 12 details the region of the state and the locations of the stations of each cluster. It is possible to observe that the regions were well defined, with stations within the same county always being in the same cluster. The density of stations per cluster was related to the availability of data for each region.

Then, within the context of AgroClimate's agricultural DAS, after clustering, the result is applied to the stations in use in StAS and BAS. For StAS, 10 meteorological stations are currently in use, as previously detailed: `Apopka (320)`, `Arcadia (490)`, `Balm (350)`, `Bronson (230)`, `Citra (250)`, `Dover (360)`, `Floral City (210)`, `Lake Alfred (330)`, `Plant City (300)`, and `Umatilla (302)`. Note that three of these ten stations were not part of the initial clustering, due to lack of representative data from the last twenty years: `Arcadia`, `Floral City` and `Plant City`. Even so,

| Cluster | FL Region | ID | Station Name | County |
|---|---|---|---|---|
| C1 | Northwest | 110 | Jay | Santa Rosa |
| | | 130 | Marianna | Gadsden |
| | | 140 | Quincy | Jackson |
| | | 160 | Monticello | Jefferson |
| C2 | North | 150 | Carrabelle | Franklin |
| | | 170 | Live Oak | Suwanee |
| | | 180 | Macclenny | Baker |
| | | 240 | Putnam Hall | Putnam |
| | | 250 | Citra | Marion |
| | | 260 | Alachua | Alachua |
| | | 270 | Hastings | St. John's |
| | | 290 | Pierson | Volusia |
| C3 | North Center | 230 | Bronson | Levy |
| | | 280 | Ocklawaha | Marion |
| | | 302 | Umatilla | Lake |
| | | 303 | Okahumpka | Lake |
| C4 | South Center | 304 | Avalon | Orange |
| | | 320 | Apopka | Orange |
| | | 330 | Lake Alfred | Polk |
| C5 | West Center | 350 | Balm | Hillsborough |
| | | 360 | Dover | Hillsborough |
| C6 | South | 340 | Kenansville | Osceola |
| | | 380 | Ona | Hardee |
| | | 410 | Belle Glade | Palm Beach |
| | | 420 | Ft. Lauderdale | Broward |
| | | 440 | Homestead | Dade |
| | | 450 | Immokalee | Collier |
| | | 460 | Palmdale | Glades |
| | | 470 | Sebring | Highlands |

Table 12 – FAWN meteorological stations clustering results.

clustering was applied to these stations according to the county or spatial distribution within one of the resulting clusters. Then, the clustering result for StAS is shown in Figure 38.

Similarly, for BAS, 15 meteorological stations are currently in use: `Alachua (260)`, `Apopka (320)`, `Arcadia (490)`, `Balm (350)`, `Bronson (230)`, `Citra (250)`, `Dade City (311)`, `Dover (360)`, `Floral City (300)`, `Jay (110)`, `Lake Alfred (330)`, `Live Oak (170)`, `Ona (380)`, `Putnam Hall (240)`, and `Sebring (470)`. Of these fifteen stations, again three stations were not part of the initial clustering because they did not have data for twenty years: `Arcadia`, `Floral City` and `Dade City`. Repeating the process, assigning a cluster to each of them according to the location, the clustering result for the BAS is presented in Figure 39.

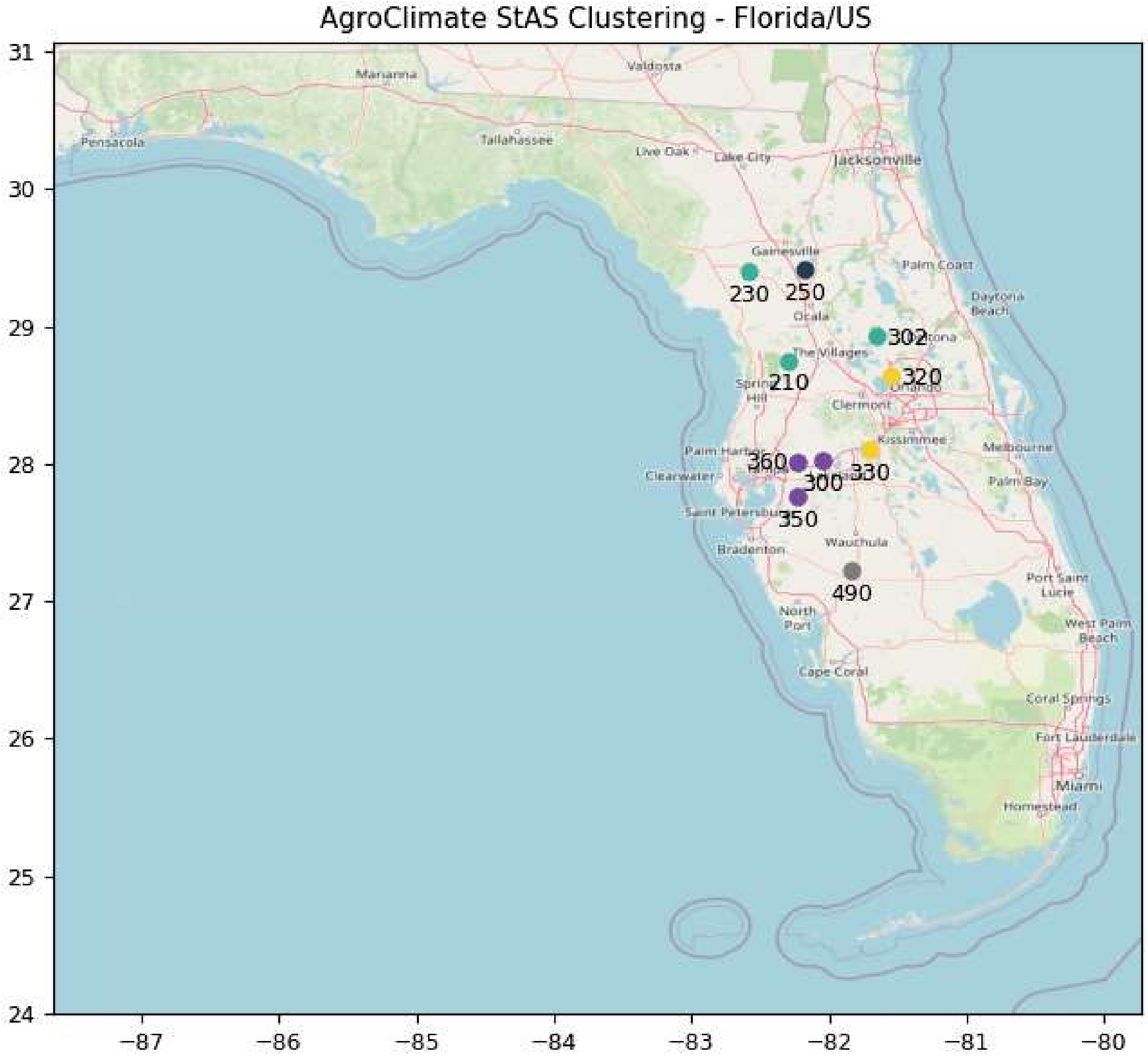


Figure 38 – AgroClimate Strawberry Advisory System final clustering.

### 6.2.6 FL Clusters Validation

The evaluation metric proposed by the framework was already calculated, based on the “leave-one-out” strategy, in order to evaluate the predictive capacity of each cluster, as shown in the Table 11. Therefore, for this case study, we proceeded with clustering only the algorithm that already presented the best performance.

Furthermore, based on the historical climate pattern of the state of Florida (BLACK, 1993), it is possible to evaluate that the clustering result makes sense, since the stations are grouped with other stations that share microclimate characteristics of the different regions of the state (ASSENG, 2013), as shown in Figure 40. Florida traditionally has seven climate divisions (NOAA, 2023), namely: Northwest, North, North Central, South Central, Everglades and Southwest Coast, Lower East Coast, and Keys. The arrangement of stations and clusters obtained by this research respects this division based on the historical behavior of the climate. Therefore, the main contribution of this

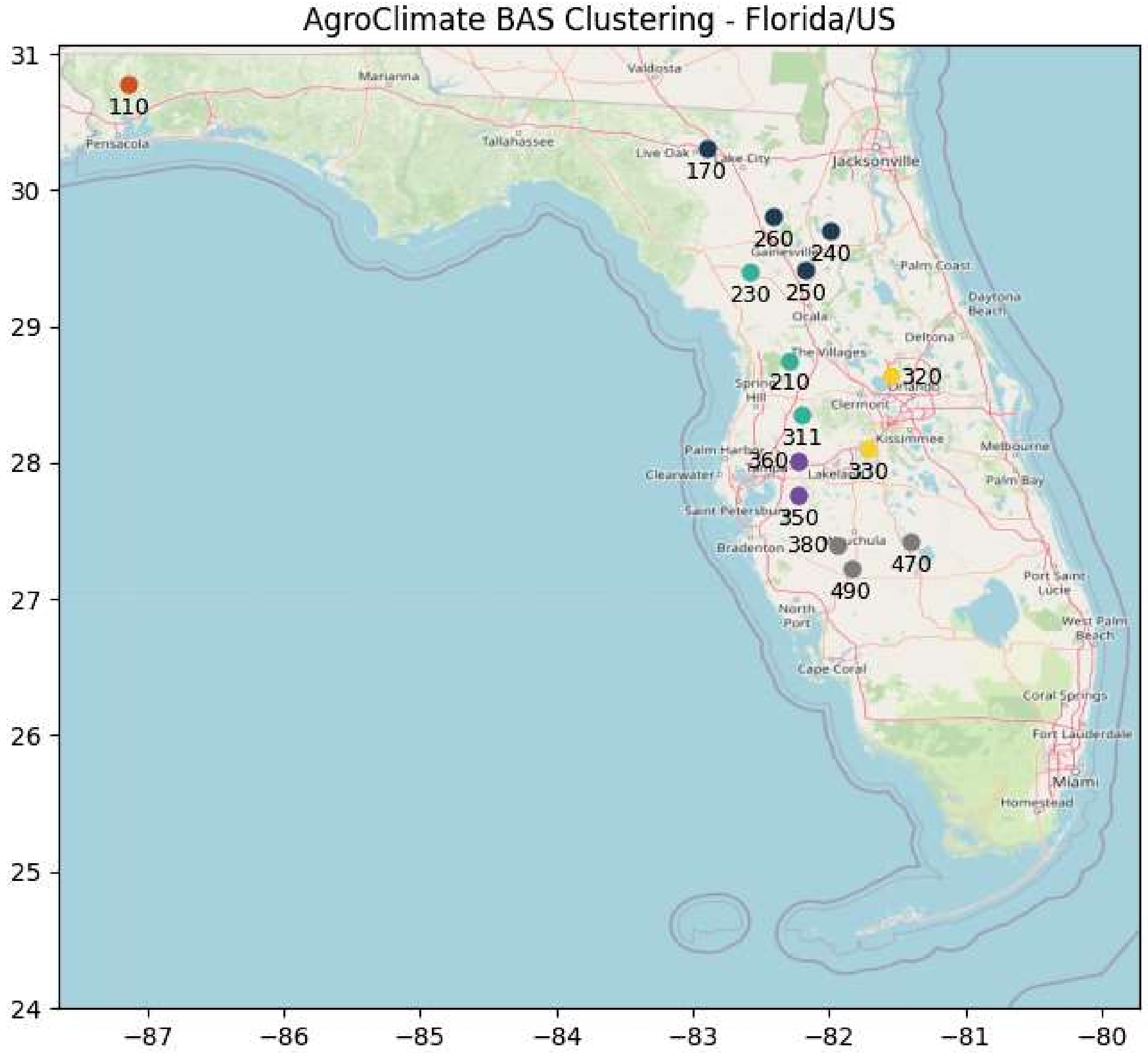


Figure 39 – AgroClimate Blueberry Advisory System final clustering.

work is the adjustment of the region's borders based on data from the last twenty years. Instead of straight horizontal cuts, the definition of cluster borders follows the climatic pattern of the border stations.

Finally, it is understood that the difference between the number of clusters (6) and the number of climate divisions (7) is due to the climate division of the Keys, in the extreme south of the state, not having FAWN stations, therefore, there is no representation of this region in the dataset of this case study. Since Florida climate differences impact on weather variables, such as temperature, relative humidity, rainfall and solar radiation, this clustering can be taken into account for the monitoring of agricultural diseases caused by fungi (BREUER; FRAISSE, 2020).

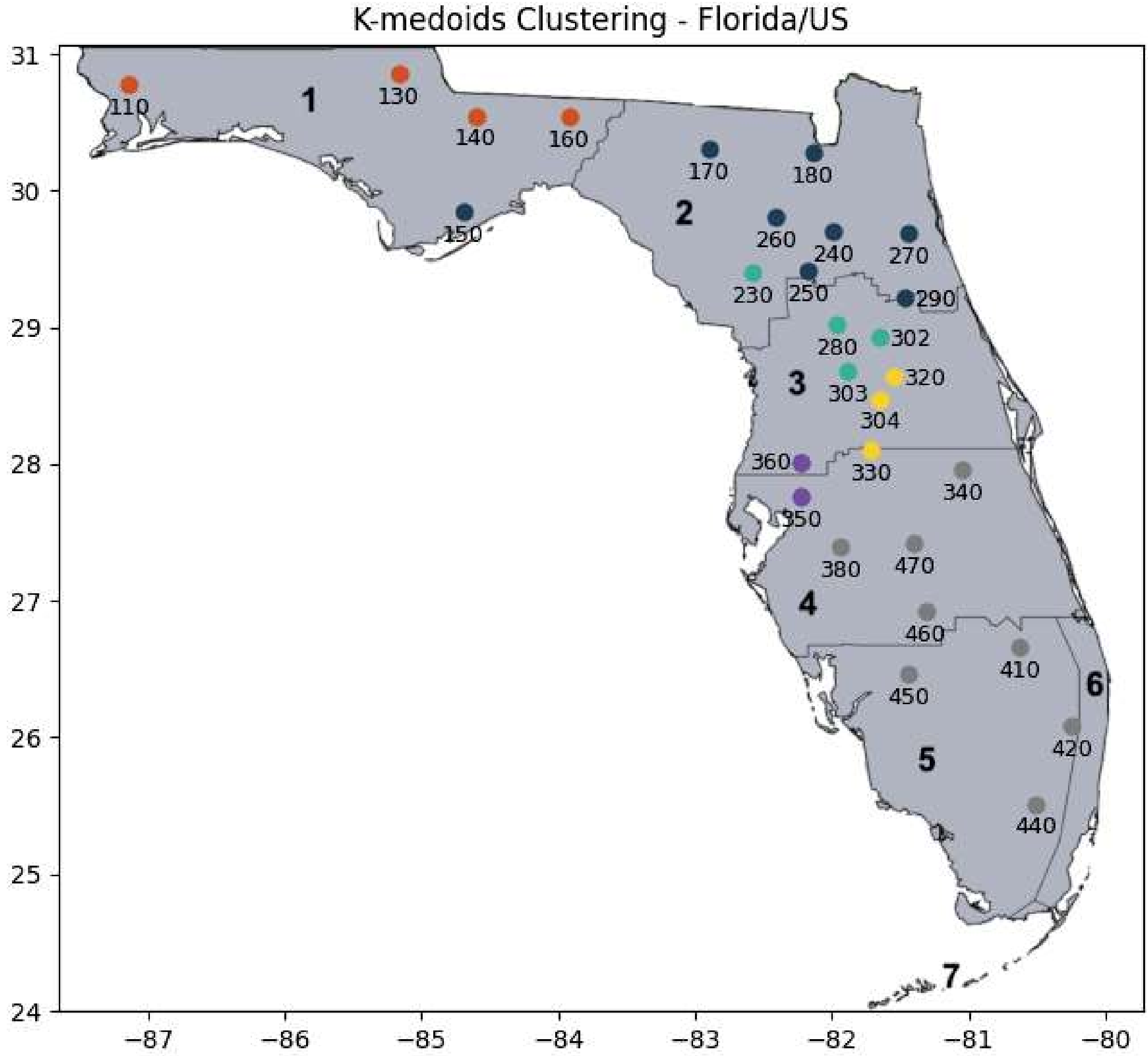


Figure 40 – Result of clustering on the map with the seven climate divisions in Florida (NOAA, 2023)

## 6.3 Discussion

Based on the presented case studies, the following discussions provide insights into the results, both individually and collectively. Both case studies were carried out this year (2023), and the order in which they were presented in this thesis does not specifically reflect the chronological order of the experiments. In fact, the second case study was started before the first, during a sandwich doctorate period of almost a year carried out at the University of Florida, together with researchers from the AgroClimate project. This project has been working for over twenty years to develop agroclimatic indicators for Florida and the southeastern region of the USA. Thus, during this period, the research group's experience in researching climate data for agriculture helped guide the development of the proposed framework. Then, after finishing the exchange period, we tried to reproduce the work in a local context, with climate data from the state of RS, focused on agricultural crops in the state. The scientific collaboration with SIMAGRO-RS made this other part of the research viable.

However, it is clear that it was not possible to advance in the first case study (SIMAGRO-RS), as much as in the second (AgroClimate-UF), and this can be understood taking into account the difference in the agricultural and technological maturity of the countries. While in the USA technology is more advanced, connectivity in rural areas is greater, and agricultural recommendation systems are more mature, in Brazil we are a few steps behind. Both in technological devices and in implementation, there is a lag in the use of technology in agriculture, which was initially also pointed out in the systematic literature review presented in Chapter 3. This is even more evident, for example, in the availability of climate data for the regions of the two study cases: while for Florida there is data for more than twenty years, for RS data is available from 2009 onwards. Thus, we reinforce the importance of research like this, so that we can advance the state of the art and intensify research with agricultural technologies also in Brazil, which has a huge productive potential. Some insights obtained from the results of the case studies are further expanded.

### 6.3.1 Main Insights

After conducting the two case studies reported here, some discussions arise from the results obtained in both datasets. First, it is interesting to point out how the results can be useful and interpreted within the agricultural context. As main outcomes, the following possibilities and research directions stand out.

- Agroclimatic Zoning: use the proposed cluster to specialize agricultural strategies and to segment the state into micro agroclimatic zones, each characterized by coherent weather patterns. This zoning can provide a better understanding of microregional climate variations, enabling the tailoring of crop choices and cultivation practices to specific zones.

- Crop-Specific Insights: use this cluster to uncover weather conditions that are most conducive to the growth and development of specific crops. By linking these clusters to the stages of crop growth, farmers can make informed decisions regarding planting, irrigation, and pest management.

- Extreme Weather Event Detection: based on this cluster, it can be possible to identify anomalous weather events, enabling early warning systems to mitigate the impact of extreme conditions on crops. This can help improve agricultural resilience and minimize yield losses.

- Decision Support for Agriculture: extract insights gained from clustering to support agricultural decision making, including crop selection, resource allocation,

> and risk management. By translating complex weather data into actionable recommendations, this study can contribute to more sustainable and productive agricultural practices.

In both case studies, the results can holds significant importance for the region's agricultural and environmental management. With this clustering of weather stations, it is possible to develop a new level of verification of weather data quality, checking whether new data collected by a station is in agreement with its cluster. From a hardware perspective, it helps to identify sensor problems in advance and perform early maintenance, reducing data loss or incorrect measurements. From a software perspective, it enables complementary checks on new measurements, comparing them with historical patterns from other stations in the same cluster.

A common situation when analyzing data from stations in a given region is when adjacent stations present very different data. For disease monitoring systems this implies different levels of risk for nearby regions, which often makes it difficult for the model to be accurate. In addition, although we now have good availability of weather stations, they are still relatively few given the total area of the state. Thus, clustering helps regions that do not have their weather station to monitor weather events more accurately, especially those located between two or more seasons.

#### *6.3.1.1 SIMAGRO-RS Outcomes*

The use of SIMAGRO-RS data as in this work, the first carried out on this dataset, and the publication of its results is important so that more people, especially researchers and producers, can explore the information it makes available and benefit from it. Therefore, computer science researchers can help farmers and agrometeorologists with the necessary technologies to read and interpret data, transforming it into useful information.

In general, the main limitation of this work is the still low availability of data. Of course, given the limited amount of data from just three seasons, the analysis is insufficient to determine whether this clustering pattern is the result of recent years only or has been occurring for longer. However, knowing the current situation of climate change occurring in different places around the world, monitoring data like these can help to identify changes more quickly, adjusting production patterns and contributing to accuracy in agricultural practices. Below are listed some possible uses of this proposed clustering.

#### *6.3.1.2 AgroClimate DAS Improvements*

For the AgroClimate DAS, the focus of the second case study, due to the maturity of FAWN's meteorological station network, with more stations and operating for longer, it was possible to further advance the analysis of improvements to the systems. Based on the resulting clusters, improvements in disease modeling are proposed.

Upon receiving new data, every 15 minutes, both DAS simulate the disease risk in real-time and predict the risk for the next 48 hours. Based on computed clusters, a conceptual approach was elaborated to check a new observation at the moment of simulation, using flags to signal potential problems, to improve the information presented in the system interface. Note that the cluster does not necessarily need to be visible. If desired, this information can be graphically added as a layer to the interface, but the key is for the system to be aware of clustering when computing disease risk levels for each station.

Initially, it is proposed to check every new value according to the historical average for the day of the season and the measured time, as calculated for clustering (light flag). Afterwards, it can be verified with the other stations within the cluster, checking if the new measurement is as plausible according to the last hours of observations in your cluster (medium flag). Furthermore, it is possible to check specifically with the nearest station(s), inside the cluster, if the new value is a potential problem (serious flag). For each level of suspected problem (light, medium and serious), a color scheme can be used, such as yellow, orange and red. Finally, it is also possible to insert an number corresponding to the sum of observations with suspicion, meaning that the last $N$ observations are under suspicion. Given that a new computation is performed every 15 minutes, at each time interval in which the situation remains suspicious, the flag can be incremented, establishing a threshold for sending warnings to stakeholders.

An interface idea, to be tested and validated, is presented in Fig. 41. In it, some stations have a colored flag indicating the level of the suspected problem at that station, considering recent observations. The number inside the flag indicates the number of measurements accumulated since the problem was observed. This idea was designed to enable the approval of the same with stakeholders, for later implementation. These flags, warnings and information generated based on cluster behavior can be sent both to producers via *SMS (Short Message Service)*, for example, to closely monitor the risk of the disease in the next few hours/days. Furthermore, it is possible to implement direct communication with the data provider, in this case FAWN, enabling the triggering of equipment maintenance when applicable.

The definition of being in agreement or not with the cluster values is broad and can be performed in several ways, mainly through statistical methods. This aspect will not be deepened here due to the understanding that depending on the variable and the perspective, this can be specialized. At different times of the season and for different diseases, the resistance of a fruit to that disease can be higher or lower. Therefore, the natural next step is to implement different forms of checking, with configurable parameters according to the case. Once the form of verification is defined, the conceptual

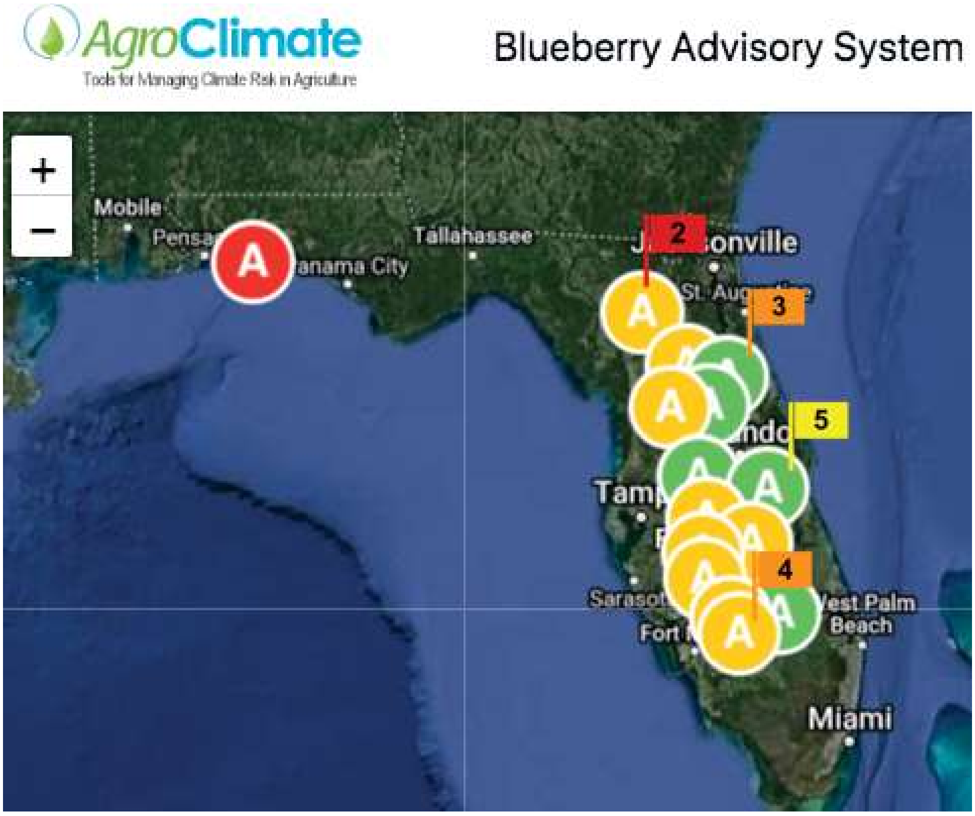


Figure 41 – BAS interface idea with station cluster quality cumulative flags.

approach can be followed. These are initial, conceptual ideas, but they appear to be feasible and reasonable advances for DAS. With these improvements, it is understood that the systems will be more accurate in predicting disease risk, increasing their efficiency and contribution to farmers.

## 6.4 Chapter Remarks

In this chapter, the two case studies developed were presented in order to demonstrate and validate the proposed framework. Experiments were carried out with two datasets from real agricultural applications, from SIMAGRO-RS and AgroClimate/-FAWN. The flow of use of the proposed framework was demonstrated and the clustering results obtained were also discussed, as well as ways of interpreting them in the agricultural context. The case studies not only validated, but presented practical applications that can be explored, showing the importance of the framework in the agrometeorological context.

# 7 CONCLUSION

This final chapter aims to summarize and bring together the main points presented in this thesis, discussing and answering the research questions previously established. The contributions and results of the proposed framework are also highlighted. Finally, the scientific contributions resulting from the different phases of this research are reported, as well as future work proposed in this research direction.

## 7.1 Research Summary

This research proposes a API-based framework for clustering of meteorological time series for agricultural applications. The framework was implemented in two versions, the first in the format of a python library (`FMTSClust.py`), and the second, representing an evolution of the first, in the format of an API, called `FMTSClustAPI`. The main objective of the framework is that the result of clustering, climate zoning, serves as input for agricultural decision systems. The framework proposes four major steps, in order to guide the clustering procedure: data extraction, data preprocessing, feature engineering, and clustering. Each step is composed of substeps supported by one or more implemented functions (library) or endpoints (API).

After design and implementation, the framework was tested and validated through two case studies, one with meteorological data from SIMAGRO-RS, on the climate of Rio Grande do Sul, in southern Brazil, and the other with meteorological data from FAWN, with data collected in the state of Florida, southeastern United States. In the initial stage, data pertaining to the Temperature, Relative Humidity, Rainfall, and Solar Radiation variables were extracted from meteorological stations situated in diverse locations within each state. The datasets contained climate data from recent years, with different periods in each study case. Afterwards, in the second stage, the data was pre-processed, with quality checking and filtering according to different agricultural interests, depending on each case, for different agricultural seasons, different crops and different agricultural activities. In the third stage, feature engineering, the data was aggregated into daily average and standardized values, in order to represent all variables on the same scale, improving their suitability for training ML models. Finally, the data was clustered, with different algorithms and evaluated using similarity metrics, in order to determine an optimal number of clusters and the algorithm with the best performance in each case study. As a result, climate zones were established for each state, with meteorological stations grouped into clusters, in order to support agricultural decision systems with a new layer of valuable knowledge about meteorology.

After analyzing the results and checking with traditional reference climate documents, for each case, it was observed that the results of the proposed clustering make sense, in general, as it preserves spatial characteristics of the different regions of the state. Furthermore, the framework makes its main contribution in adjusting the borders of climatic regions, as well as the number of stations in each region. The analysis of this climate data specifically for each region, and also focused on specific seasons and agricultural activities, presents specialized insights in comparison to the traditional and historical division of climate regions for both states.

Regarding the volume of data, in places that have been collecting meteorological data for a longer time, as FAWN has been doing for Florida for more than twenty years, there have already been many advances in data processing and, consequently, in the use of this data in agricultural practices. However, in places where meteorological data collection is recent, this framework may have even more contributions to make, instigating and speeding up the analysis of this data, in an innovative way that is unprecedented in many regions. In the case of the SIMAGRO-RS dataset, for example, this research was the first to use it, being the first scientific publication on this dataset. As it is a recent sensor network, in operation for only three years, the database is not yet large enough for deeper analysis. However, as measurements are collected at high frequency, every thirty minutes, the database will soon have a very large volume of data for more complex analyses. Research with this data set must continue, year after year, and this framework can greatly facilitate this, as it already implements the import of data from this specific database. Thus, it will also be possible to observe the impact of increased data volume on the quality of clustering results, and, given the importance of this economic sector for the state of RS, it is vital that this analysis is continuous, to at least mitigate the delay in data collection, compared to other regions.

In general, the framework proposed in this thesis is intended to pave the way towards research on clustering MTS, bringing together stat-of-the-art statistical and ML techniques, in order to compile and automate them, building a solid ground to enable the advancement of research in this direction. The open source implementation, in a widely known programming language, with modern libraries and frameworks, and publicly available, leaves room for collaboration in many aspects, both conceptually and practically. The discussion remains open about clustering theories and algorithms, MTS analysis, feature engineering and extracting new insights from data. At any time that researchers or agricultural stakeholders want to collaborate, they can access published textual materials and implemented codes, extending functionalities, customizing for other data sets, other performance evaluation metrics, or even refactoring the existing implementation. Specifically, below are some possibilities already envisioned for future work.

## 7.2 Research Questions and Hypothesis Discussion

To conclude this research, the answers to the research questions initially defined in the introductory chapter are reported below.

To answer the central Research Question ***(cRQ: How to effectively cluster meteorological time series data in order to help agricultural decision support systems?)***, this research proposed an innovative and automated API-based framework to clustering meteorological time series for agricultural applications, to address this complex challenge, following a well-defined sequence of steps, using statistical and ML techniques to cluster MTS data and producing climate zoning as the main output, to be used as input to agricultural decision systems and improve recommendations accuracy.

Based on a systematic review of the literature and the exploration of advanced clustering techniques and methodologies, it was possible to answer the first minor research question **(*mRQ1: What are the main solutions for clustering meteorological time series in the agricultural context?)*** From the RSL, the two most used algorithms with this type of time series were identified: K-means and Hierarchical Clustering. This served as the basis for beginning the implementation of the framework, which was based on existing studies with these two techniques. Both were then implemented and extended in the proposed framework, to deal with multivariate time series. Subsequently, other algorithms were also implemented to increase the range of possibilities and, based on the performance presented in preliminary studies, the K-medoids algorithm was also chosen to compose the framework. These three algorithms, then, compose the answer to the *mRQ1*.

For the second minor Research Question ***(mRQ2: How are these algorithms evaluated and compared?)***, the answer is also based on the RSL outocomes, however, it advances and permeates the development of the framework, addressing the challenge of evaluating clustering solutions. It was identified that comparing clustering solutions is quite challenging, mainly because the solutions differ greatly when they are designed. There is an extensive list of intra- and extra-cluster similarity metrics, as well as a diverse list of algorithm performance metrics, as presented in previous chapters. The applicability of the metrics also varies greatly, making this task difficult. Thus, in general, the main insight extracted from the investigation into performance metrics was that specialized metrics are more appropriate than generalist metrics. For this reason, in the proposed framework, metrics of both natures were used. While well-known metrics such as Silhouette Score and Elbow Plot were used to search for an optimal number of clusters, a new metric (RMSE prediction), specific to evaluating the quality of climate zoning, was designed. These metrics made it possible to evaluate and compare the three algorithms implemented in the framework, ensuring the best performance result for each case study.

Finally, to answer the third minor research question (***mRQ3: What factors motivate the clustering of time series in agriculture applications?***), the outcomes of the reviewed papers, the implementation of the framework and the case studies were also used. The RSL made it possible to survey the reasons why clustering is applied in different for different purposes (monitoring crops, soil, water resources, weather), with practically all of them focused on zoning agricultural areas, according to some meteorological aspect. Conducting case studies made it possible to get closer to stakeholders to identify and understand how clustering can help improve agricultural practices. In summary, there is a wide use of agricultural decision systems to inform stakeholders with recommendations, however, in many cases, climate unpredictability impacts and distorts these results. Therefore, the updated climate zoning of agricultural regions helps to correct and improve recommendations, which is the main factor that motivates the use of clustering of agricultural time series, also answering this research question.

In a macro and general view, after all stages of this research, with literature review, basic concepts, implementation of the framework and ending with the two case studies, the analysis and discussion of the results makes it possible to confirm the research hypothesis established at the beginning of this research. (***RHyp: The existence of a framework, supported by a software implementation to automate processes, can assist stakeholders and make the use of meteorological time series clustering accessible to a greater number of agricultural applications.***) By bringing together and implementing clustering algorithms, taking advantage of their adaptability to climatic variables, the proposed framework contributes to the extraction of a new layer of agrometeorological knowledge, relating it to agricultural activities. The results of this investigation offer a valuable toolkit for stakeholders to make informed decisions and adapt strategies in real time. As we face an ever-evolving landscape of climate variability, this work lays a foundation for future research efforts and highlights the potential for dynamic clusters in the evolution of agricultural decision support systems.

## 7.3 Future Works

As future work, many research possibilities are identified, both in depth and breadth. Initially, from the perspective of the data set, it is possible to expand the range of climatic variables, in different ways, such as increasing the number of variables. Currently, both case study databases (SIMAGRO-RS and FAWN) and other existing meteorological databases collect more variables than the four used here. The choice of variables for the case studies (Temperature, Relative Humidity, Rainfall and Solar Radiation), until then, was based on discussion with agricultural stakeholders about which variables would have the most impact on what was desired to be analyzed (crop, season, agricultural activity). However, many other variables are available, such as Wind Speed, Wind Direction, Dew Point, Soil Temperature and Air Pressure, in addition to variables that can be derived from these collected such as Evapotranspiration, Chill Hours and Leaf Weatness, which can influence clustering and improve the accuracy of climate zoning. In addition to the number of variables, the granularity of the data can also be varied, deepening the analysis, for example, using data at a higher measurement frequency, such as hourly data. Depending on the crop and agricultural season, data analysis with a greater zoom can reveal new layers of information about agrometeorological patterns, improving clustering.

From the technical perspective of computer science, future work is also planned regarding the implementation of other clustering algorithms. The state of the art in this field of research is constantly changing, with the subject of ML experiencing enormous popularity and growth in recent years. This means that new algorithms are developed frequently, creating the need to keep up to date and follow the most relevant techniques for clustering tasks. Likewise, the evaluation metrics related to these techniques are also frequently evolving. The implementation of similarity and performance metrics, complementary to those already used, is seen as essential for greater adaptation of the framework in different scenarios. Furthermore, to implement FMTSClust, auxiliary libraries were used, such as pandas and scikit-learn, and the web framework, FastAPI. These software packages are also undergoing constant evolution, awakening the need for the framework proposed here to follow this evolution, updating and complementing the implementation of functionalities.

From a data availability perspective, it is also intended to explore the use of data from remote sensing, such as gridded satellite images, from well-known platforms and repositories such as Geographic Information System (GIS) and Google Earth Engine. The use of these data would increase the range of analysis possibilities, enabling, in addition to the clustering of meteorological stations, the spatial definition of climatic zones. In addition to increasing the amount of data for analysis, satellite data will make it possible to increase the granularity of the data. This can enhance the investigation of geopositional aspects, including new variables, such as relief, and making it possible to perform other ML-based tasks, such as prediction.

Another important future activity that is already on the research radar is the analysis of the impact of global meteorological phenomena on clustering, such as ENSO (El-

Niño Southern Oscillation). This type of analysis is seen as having greater potential for future scientific contributions from this work. ENSO is an atmospheric and oceanic phenomenon characterized by an abnormal warming of surface waters in the Tropical Pacific Ocean, altering global wind patterns, modifying the rainfall regime in low and medium latitudes. This phenomenon occurs at irregular intervals and has intensely impacted many regions around the world, as well as the states of Florida and Rio Grande do Sul. Therefore, the intention is to analyze the climate data in order to split the dataset, according to the ENSO classification (in El-Niño, La-Niña and Neutral years), to identify and analyze the impact of ENSO on clustering and climate zoning. The result of this is expected to be different clustering for each type of ENSO year, and this would enable different agricultural insights for different situations. This would make it possible to improve agricultural management in years that are characterized by unpredictability and productive and financial losses for farmers.

Finally, researchers, students, professors and agricultural stakeholders are strongly invited to collaborate with this research in developing future research together. The aim of developing this framework is to popularize and speed up access to ML models that can contribute to agricultural decision systems. Therefore, contributions are welcome in the direction of maturing the framework, increasing the accuracy of agricultural recommendations and, consequently, improving productivity and reducing costs for producers, contributing to the global food supply.

## 7.4 Scientific Contributions

The main contribution of this thesis is the design and implementation of a framework for clustering meteorological time series for agricultural applications. However, to achieve this contribution, many intermediate phases of research were necessary, to test and validate possibilities in different research directions and to better understand the research context. These different phases resulted in many contributions, especially in the form of research papers, published in different conferences and journals, as well as different implementation versions that accompanied each paper.

Initially, research was focused on the time series data format, in order to better understand its characteristics, types of time series used in agricultural applications and time series compression for IoT. Afterwards, the research turned to ML applications in these time series from sensors deployed in rural environments, up to MTS. In the final phase, the research was specialized for clustering MTS, related to the agricultural applications previously investigated. All these phases contributed, in different ways and degrees, to the maturation of the work and helped to design the framework proposed here in this thesis. The publications relating to these different phases of research are listed below, in chronological order.

- **Title:** Quality Checking of Meteorological Observations used for Disease Alert Systems in Florida. (OLIVEIRA JR.; CAVALHEIRO; FRAISSE, 2023)
  ***Authors:*** *Marcos A. de Oliveira Jr, Gerson Geraldo H. Cavalheiro, Clyde Fraisse*
  *Conference Poster - Artificial Intelligence in Agriculture 2023 - Orlando/FL, USA.*
  Link: `https://abe.ufl.edu/2023-ai-conference/posters/`

- **Title:** Clustering Weather Time Series used for Agricultural Disease Alert Systems in Florida. (DE OLIVEIRA et al., 2023)
  ***Authors:*** *Marcos A. De Oliveira, Gerson Geraldo H. Cavalheiro, Vinícius Andrei Cerbaro, Clyde Fraisse*
  *Conference Paper - 2023 IEEE 26th International Symposium on Real-Time Distributed Computing (ISORC) - Nashville/TN, USA.*
  DOI: `https://doi.org/10.1109/ISORC58943.2023.00029`

Also, other two papers were finished and submitted at the time of this thesis, and are currently awaiting review results, as follows:

- **Title:** Clustering of meteorological data to improve agricultural decisions: a case study with SIMAGRO-RS.
  ***Authors:*** *Marcos A. de Oliveira Jr, Flavio Varone, Clyde William Fraisse, Ricardo Matsumura Araujo, Gerson Geraldo H. Cavalheiro*
  *Conference Paper - XX Brazilian Symposium on Information Systems - Juiz de Fora/MG, Brazil.*

- **Title:** Meteorological Time Series Clustering in Agricultural Applications: A Systematic Literature Review.
  ***Authors:*** *Marcos A. de Oliveira Jr., Monalisa F. Oliveira, Gerson Geraldo H. Cavalheiro*
  *Conference Paper - XX Brazilian Symposium on Information Systems - Juiz de Fora/MG, Brazil.*

- **Title:** Weather Data Clustering Approach for Improving Agricultural Disease Alert Systems.
  ***Authors:*** *Marcos A. de Oliveira Jr., Gerson Geraldo H. Cavalheiro, Vinícius Andrei Cerbaro, Clyde Fraisse*
  *Full Paper - Computers and Electronics in Agriculture - Special Issue: AI in Agriculture: Innovation and Discovery to Equitably meet Producer Needs and Perceptions.*

In addition to the papers directly related to the central theme of this thesis, other papers were also published, related to collateral research and academic cooperation, as follows:

- **Title:** A study on Strategies for Time Series Compression for IoT applications (In Portuguese). (OLIVEIRA JR.; CAVALHEIRO, 2021)

***Authors:*** *Marcos A. de Oliveira Jr., Gerson Geraldo H. Cavalheiro*
*Short Paper - XII Escola Regional de Alto Desempenho da Região Sul 2021, Brazil.*
DOI: `https://doi.org/10.5753/eradrs.2021.14800`

- **Title:** Effects of agro-sensor time series approximation on plant stress detection: an experimental study. (OLIVEIRA JR. et al., 2021)
  ***Authors:*** *Marcos A. de Oliveira Junior, Gregory Sedrez, Anderson Monteiro, Fernando Emilio Puntel, Gerson Geraldo H. Cavalheiro*
  *Conference Paper - XIII Congresso Brasileiro de Agroinformática - Porto Alegre/RS, Brazil.*
  DOI: `https://doi.org/10.5753/sbiagro.2021.18380`
- **Title:** An Application with Jetson Nano for Plant Stress Detection and On-field Spray Decision. (DE et al., 2022)
  ***Authors:*** *Marcos A. de Oliveira Jr., Gregory Sedrez, Guilherme de Souza, Gerson Geraldo H. Cavalheiro*
  *Conference Paper - International Conference on Sensor Networks 2022, Online.*
  DOI: `https://doi.org/10.5220/0010983900003118`
- **Title:** ABP vs. OTAA activation of LoRa devices: an Experimental Study in a Rural Context. (ROCHA et al., 2023)
  ***Authors:*** *Anderson M. Da Rocha; Marcos A. De Oliveira; Pauletti José F. M.; Gerson Geraldo H. Cavalheiro*
  *Conference Paper - 2023 International Conference on Computing, Networking and Communications - Honolulu/HW, USA.*
  DOI: `https://doi.org/10.1109/icnc57223.2023.10074553`
- **Title:** ML-based Plant Stress Detection from IoT-sensed Reduced Electromes. (DE OLIVEIRA JR; SEDREZ; H. CAVALHEIRO, 2023)
  ***Authors:*** *Marcos De Oliveira Jr, Gregory Sedrez, Gerson Geraldo H. Cavalheiro*
  *Conference Paper - 36th Florida Artificial Intelligence Research Society Conference 2023 - Clearwater/FL, USA.*
  DOI: `https://doi.org/10.32473/flairs.36.133180`
- **Title:** Time Series Compression for IoT: A Systematic Literature Review. (OLIVEIRA et al., 2023)
  ***Authors:*** *Marcos A. de Oliveira Jr., Anderson Monteiro da Rocha, Fernando Emilio Puntel, Gerson Geraldo H. Cavalheiro*
  *Full Paper - Wireless Communications and Mobile Computing (Hindawi).*
  DOI: `https://doi.org/10.1155/2023/5025255`

Finally, another important and significant contribution during the development period of this research was co-advising a undergraduate final paper, as follows:

- **Title:** Plant Electrome Time Series Approximation and its Application in Classification Algorithms (In Portuguese).
  ***Author:*** *Gregory Sedrez*
  ***Advisor:*** *Gerson Geraldo H. Cavalheiro*
  ***Co-Advisor:*** *Marcos A. de Oliveira Jr*
  *Undergraduate Final Paper - Universidade Federal de Pelotas, Pelotas/RS, Brazil*

# REFERENCES

ABANDA, A.; MORI, U.; LOZANO, J. A. A review on distance based time series classification. **Data Mining and Knowledge Discovery**, [S.l.], v.33, p.378–412, mar 2019.

AHER, M. C.; YADAV, S. M. Assessment of rainfall trend and variability of semi-arid regions of Upper and Middle Godavari basin, India. **Journal of Water and Climate Change**, [S.l.], v.12, n.8, p.3992–4006, 10 2021.

AHMED, M.; SERAJ, R.; ISLAM, S. M. S. The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. **Electronics**, [S.l.], v.9, n.8, 2020.

AKAY, . Examination of the 21 European countries and Turkey in terms of water resources along with the effect of climate change by time series clustering. **Environmental Earth Sciences**, [S.l.], v.80, 2021.

ANDERSON, R.; BAYER, P. E.; EDWARDS, D. Climate change and the need for agricultural adaptation. **Current Opinion in Plant Biology**, [S.l.], v.56, p.197–202, 2020. Biotic interactions – AGRI 2019.

ASSENG, S. **Agriculture and Climate Change in the Southeast USA**. Washington, DC: Island Press/Center for Resource Economics, 2013.

AZIMI, R.; GHOFRANI, M.; GHAYEKHLOO, M. A hybrid wind power forecasting model based on data mining and wavelets analysis. **Energy Conversion and Management**, [S.l.], v.127, 2016.

BEN AYED, R.; HANANA, M. Artificial Intelligence to Improve the Food and Agriculture Sector. **Journal of Food Quality**, [S.l.], 2021.

BIEHL, M. **API Architecture**. [S.l.]: API-University Press, 2015. v.2.

BLACK, R. J. **Florida climate data**. [S.l.]: University of Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, 1993.

BLáZQUEZ-GARCÍA, A.; CONDE, A.; MORI, U.; LOZANO, J. A. A Review on Outlier/Anomaly Detection in Time Series Data. **ACM Comput. Surv.**, New York, NY, USA, v.54, n.3, apr 2021.

BREGAGLIO, S. et al. Improving crop yield prediction accuracy by embedding phenological heterogeneity into model parameter sets. **Agricultural Systems**, [S.l.], v.209, p.103666, 2023.

BREUER, N.; FRAISSE, C. W. Climate Services for Agricultural and Livestock Producers: What have we learned? **Agrometeoros**, [S.l.], v.28, 2020.

CASSISI, C. et al. Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining. In: KARAHOCA, A. (Ed.). **Advances in Data Mining Knowledge Discovery and Applications**. Rijeka: IntechOpen, 2012.

CERLINI, P. B.; SILVESTRI, L.; SARACENI, M. Quality control and gap-filling methods applied to hourly temperature observations over central Italy. **Meteorological Applications**, [S.l.], v.27, n.3, p.e1913, 2020.

CHEN, S.; ZWART, J. A.; JIA, X. Physics-Guided Graph Meta Learning for Predicting Water Temperature and Streamflow in Stream Networks. In: ACM SIGKDD CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 28., 2022. **Proceedings...** [S.l.: s.n.], 2022. p.2752–2761.

CHEN, Z. et al. Scalable nearest neighbor based hierarchical change detection framework for crop monitoring. In: IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.1309–1314.

CLIFTON, A.; LUNDQUIST, J. K. Data Clustering Reveals Climate Impacts on Local Wind Phenomena. **Journal of Applied Meteorology and Climatology**, Boston MA, USA, v.51, n.8, p.1547 – 1557, 2012.

CONRADT, T.; GORNOTT, C.; WECHSUNG, F. Extending and improving regionalized winter wheat and silage maize yield regression models for Germany: Enhancing the predictive skill by panel definition through cluster analysis. **Agricultural and Forest Meteorology**, [S.l.], v.216, p.68–81, 2016.

COOPER, H. M. Organizing knowledge syntheses: A taxonomy of literature reviews. **Knowledge in Society**, [S.l.], v.1, n.104, 1988.

CUI, M. et al. Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. **Accounting, Auditing and Finance**, [S.l.], v.1, n.1, p.5–8, 2020.

CUTURI, M.; BLONDEL, M. Soft-DTW: a Differentiable Loss Function for Time-Series. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 34., 2017. **Proceedings. . .** PMLR, 2017. p.894–903. (Proceedings of Machine Learning Research, v.70).

DAS, M.; GHOSH, S. K. Data-driven approaches for meteorological time series prediction: A comparative study of the state-of-the-art computational intelligence techniques. **Pattern Recognition Letters**, [S.l.], v.105, p.155–164, 2018. Machine Learning and Applications in Artificial Intelligence.

DE, O. J. M. A.; SEDREZ, G.; SOUZA, G. de; CAVALHEIRO, G. G. H. An Application with Jetson Nano for Plant Stress Detection and On-field Spray Decision. In: INTERNATIONAL CONFERENCE ON SENSOR NETWORKS - SENSORNETS„ 11., 2022. **Proceedings. . .** SciTePress, 2022. p.215–222.

DE OLIVEIRA JR, M.; SEDREZ, G.; H. CAVALHEIRO, G. G. ML-based Plant Stress Detection from IoT-sensed Reduced Electromes. **The International FLAIRS Conference Proceedings**, [S.l.], v.36, n.1, May 2023.

DE OLIVEIRA, M. A.; CAVALHEIRO, G. G. H.; CERBARO, V. A.; FRAISSE, C. Clustering Weather Time Series used for Agricultural Disease Alert Systems in Florida. In: IEEE 26TH INTERNATIONAL SYMPOSIUM ON REAL-TIME DISTRIBUTED COMPUTING (ISORC), 2023., 2023. **Anais. . .** [S.l.: s.n.], 2023. p.158–163.

DEBAUCHE, O.; MAHMOUDI, S.; MANNEBACK, P.; LEBEAU, F. Cloud and distributed architectures for data management in agriculture 4.0 : Review and future trends. **Journal of King Saud University - Computer and Information Sciences**, [S.l.], v.34, n.9, p.7494–7514, 2022.

DEFAYS, D. An efficient algorithm for a complete link method. **The Computer Journal**, [S.l.], v.20, n.4, p.364–366, 01 1977.

DENG, J. D. et al. Analyzing Wind Speed Data through Markov Chain Based Profiling and Clustering. In: MLSDA 2014 2ND WORKSHOP ON MACHINE LEARNING FOR SENSORY DATA ANALYSIS, 2014. **Proceedings. . .** [S.l.: s.n.], 2014.

DUBES, R.; JAIN, A. Clustering Methodologies in Exploratory Data Analysis. **Advances in Computers**, [S.l.], v.19, p.113–228, 1980.

ESLING, P.; AGON, C. Time-Series Data Mining. **ACM Comput. Surv.**, New York, NY, USA, v.45, n.1, dec 2012.

ETIENNE, E.; DEVINENI, N.; KHANBILVARDI, R.; LALL, U. Development of a Demand Sensitive Drought Index and its application for agriculture over the conterminous United States. **Journal of Hydrology**, [S.l.], v.534, p.219–229, 2016.

FERRELLI, F. et al. Climate regionalization and trends based on daily temperature and precipitation extremes in the south of the Pampas (Argentina). **Cuadernos de Investigación Geográfica**, [S.l.], v.45, n.1, p.393–416, Jun. 2019.

FRERY, A. C. **Interquartile Range**. [S.l.]: Springer International Publishing, 2021.

GANDIN, L. S. Complex Quality Control of Meteorological Observations. **Monthly Weather Review**, Boston MA, USA, v.116, n.5, p.1137 – 1156, 1988.

GANGULY, A. R.; STEINHAEUSER, K. Data Mining for Climate Change and Impacts. In: IEEE INTERNATIONAL CONFERENCE ON DATA MINING WORKSHOPS, 2008., 2008. **Anais. . .** [S.l.: s.n.], 2008. p.385–394.

GENT, D. H.; MAHAFFEE, W. F.; MCROBERTS, N.; PFENDER, W. F. The Use and Role of Predictive Systems in Disease Management. **Annual Review of Phytopathology**, [S.l.], v.51, n.1, p.267–289, 2013.

GOVENDER, P.; SIVAKUMAR, V. Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). **Atmospheric Pollution Research**, [S.l.], v.11, n.1, p.40–56, 2020.

Governo do Estado do Rio Grande do Sul. **Atlas Socioeconômico do Rio Grande do Sul**. 6ª.ed. Porto Alegre/RS: [s.n.], 2021.

Governo do Estado do Rio Grande do Sul. **Rio Grande do Sul - South Brazilian State presents itself to the world as a model of sustainable diversified agriculture**. Porto Alegre, RS: Secretaria da Agricultura, Pecuária e Desenvolvimento Rural do RS, 2022.

GUILLéN, A. J.; CRESPO, A.; GóMEZ, J. F.; SANZ, M. D. A framework for effective management of condition based maintenance programs in the context of industrial development of E-Maintenance strategies. **Computers in Industry**, [S.l.], v.82, p.170–185, 2016.

GUO, A.; JIANG, A.; LIN, J.; LI, X. Data mining algorithms for bridge health monitoring: Kohonen clustering and LSTM prediction approaches. **The Journal of Supercomputing**, [S.l.], v.76, p.932–947, 2020.

HAJLAOUI, R.; ALSOLAMI, E.; MOULAHI, T.; GUYENNET, H. An adjusted K-medoids clustering algorithm for effective stability in vehicular ad hoc networks. **International Journal of Communication Systems**, [S.l.], v.32, n.12, p.e3995, 2019.

HAMILTON, J. D. **Time series analysis**. [S.l.]: Princeton university press, 2020.

IRANI, J.; PISE, N. N.; PHATAK, M. V. Clustering Techniques and the Similarity Measures used in Clustering: A Survey. **International Journal of Computer Applications**, [S.l.], v.134, p.9–14, 2016.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data Clustering: A Review. **ACM Comput. Surv.**, New York, NY, USA, v.31, n.3, p.264–323, sep 1999.

JAVAID, M.; HALEEM, A.; KHAN, I. H.; SUMAN, R. Understanding the potential applications of Artificial Intelligence in Agriculture Sector. **Advanced Agrochem**, [S.l.], v.2, n.1, p.15–30, 2023.

KARUNATHILAKE, E. M. B. M. et al. The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture. **Agriculture**, [S.l.], v.13, n.8, 2023.

KAUFMAN, L.; ROUSSEEUW, P. **Finding Groups in Data: An Introduction To Cluster Analysis**. [S.l.: s.n.], 1990.

KHOSLA, M. et al. Machine learning in resting-state fMRI analysis. **Magnetic Resonance Imaging**, [S.l.], v.64, p.101–121, 2019. Artificial Intelligence in MRI.

KITCHENHAM, B. A.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. [S.l.]: Keele University and Durham University Joint Report, 2007. (EBSE 2007-001).

KLUYVER, T. et al. **Jupyter Notebooks – a publishing format for reproducible computational workflows**. [S.l.: s.n.], 2016. p.87–90.

KOBRYN, C. Modeling components and frameworks with UML. **Communications of the ACM**, [S.l.], v.43, n.10, p.31–38, 2000.

KOMALASARI, K. E.; PAWITAN, H.; FAQIH, A. Descriptive Statistics and Cluster Analysis for Extreme Rainfall in Java Island. **IOP Conference Series: Earth and Environmental Science**, [S.l.], v.58, n.1, p.012039, mar 2017.

LATHKAR, M. Introduction to FastAPI. **High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python**, Berkeley, CA, p.1–28, 2023.

LEAL FILHO, W. et al. Handling the impacts of climate change on soil biodiversity. **Science of The Total Environment**, [S.l.], v.869, p.161671, 2023.

LI, S.; CHEN, L.; CHEN, S. An SNN Ontology Based Environment Monitoring Method for Intelligent Irrigation System. **Journal of Shanghai Jiaotong University (Science)**, [S.l.], v.23, 06 2018.

LUKASOVá, A. Hierarchical agglomerative clustering procedure. **Pattern Recognition**, [S.l.], v.11, n.5, p.365–381, 1979.

MA, Y.; XIONG, Q.; ZHU, J.; JIANG, S. Early Warning Indexes Determination of the Crop Injuries Caused by Waterlogging Based on DHSVM Model. **The Journal of Supercomputing**, USA, v.76, n.4, p.2435–2448, apr 2020.

MANIKAS, K. Revisiting software ecosystems Research: A longitudinal literature study. **Journal of Systems and Software**, [S.l.], v.117, p.84–103, 2016.

MASSE, M. **REST API design rulebook**: designing consistent RESTful web service interfaces. [S.l.]: " O'Reilly Media, Inc.", 2011.

MCMEEKIN, N.; WU, O.; GERMENI, E.; BRIGGS, A. How methodological frameworks are being developed: evidence from a scoping review. **BMC Medical Research Methodology**, [S.l.], v.20, p.173, 2020.

MUDELSEE, M. **Climate Time Series Analysis**. [S.l.]: Springer Cham, 2014. v.51.

MUDELSEE, M. Trend analysis of climate time series: A review of methods. **Earth-Science Reviews**, [S.l.], v.190, p.310–322, 2019.

MUELLER, J. **Special edition using SOAP**. [S.l.]: Que Publishing, 2002.

MULLAPUDI, A.; VIBHUTE, A. D.; MALI, S.; PATIL, C. H. Spatial and Seasonal Change Detection in Vegetation Cover Using Time-Series Landsat Satellite Images and Machine Learning Methods. **SN Comput. Sci.**, Berlin, Heidelberg, v.4, n.3, mar 2023.

MURTAGH, F.; LEGENDRE, P. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? **Journal of classification**, [S.l.], v.31, p.274–295, 2014.

MUSTARIK, A.; SULTAN, M. T.; ISLAM, M. M. A Microclimate based Crop Recommender System for Precision Agriculture. **Int J Comput Appl**, [S.l.], v.183, n.3, p.41–46, 2021.

NOAA. **CONUS Climate Divisions**. Accessed on November, 2023. Disponível em: <https://www.ncei.noaa.gov/access/monitoring/reference-maps/conus-climate-divisions>.

OKOLI, C. A Guide to Conducting a Standalone Systematic Literature Review. **Communications of the Association for Information Systems**, [S.l.], v.37, 11 2015.

OLIVEIRA JR., M. A. de; CAVALHEIRO, G. G. H.; FRAISSE, C. Poster: Quality Checking of Meteorological Observations used for Disease Alert Systems in Florida. In: AI IN

AGRICULTURE: INNOVATION AND DISCOVERY TO EQUITABLY MEET PRODUCER NEEDS AND PERCEPTIONS, 2023. **Anais. . .** [S.l.: s.n.], 2023.

OLIVEIRA JR., M. de; CAVALHEIRO, G. A study on Strategies for Time Series Compression for IoT applications (In Portuguese). In: XXI ESCOLA REGIONAL DE ALTO DESEMPENHO DA REGIãO SUL, 2021, Porto Alegre, RS, Brasil. **Anais. . .** SBC, 2021. p.123–124.

OLIVEIRA JR., M. de et al. Effects of agro-sensor time series approximation on plant stress detection: an experimental study. In: XIII CONGRESSO BRASILEIRO DE AGROINFORMáTICA, 2021, Porto Alegre, RS, Brasil. **Anais. . .** SBC, 2021. p.99–107.

OLIVEIRA-JúNIOR, J. F. de et al. Fire foci in South America: Impact and causes, fire hazard and future scenarios. **Journal of South American Earth Sciences**, [S.l.], v.112, p.103623, 2021.

OLIVEIRA-JúNIOR, J. F. de et al. Wet and dry periods in the state of Alagoas (Northeast Brazil) via Standardized Precipitation Index. **Journal of Atmospheric and Solar-Terrestrial Physics**, [S.l.], v.224, p.105746, 2021.

OLIVEIRA, M. A. d.; ROCHA, A. M. da; PUNTEL, F. E.; CAVALHEIRO, G. G. H. Time Series Compression for IoT: A Systematic Literature Review. **Wireless Communications and Mobile Computing**, [S.l.], v.2023, p.5025255, 2023.

PARÉ, G.; TRUDEL, M.-C.; JAANA, M.; KITSIOU, S. Synthesizing information systems knowledge: A typology of literature reviews. **Information & Management**, [S.l.], v.52, n.2, p.183–199, 2015.

PERALTA, J. H. **Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More**. [S.l.]: Simon and Schuster, 2023.

POTT, L. P. et al. Satellite-based data fusion crop type classification and mapping in Rio Grande do Sul, Brazil. **ISPRS Journal of Photogrammetry and Remote Sensing**, [S.l.], v.176, p.196–210, 2021.

PRAKASH, C.; SINGH, L. P.; GUPTA, A.; LOHAN, S. K. Advancements in smart farming: A comprehensive review of IoT, wireless communication, sensors, and hardware for agricultural automation. **Sensors and Actuators A: Physical**, [S.l.], v.362, p.114605, 2023.

QIAN, Z.; PEI, Y.; ZAREIPOUR, H.; CHEN, N. A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. **Applied Energy**, [S.l.], v.235, p.939–953, 2019.

RASCHKA, S. **Python Machine Learning**. [S.l.]: Packt Publishing, 2015.

RASCHKA, S.; LIU, Y.; MIRJALILI, V.; DZHULGAKOV, D. **Machine Learning with PyTorch and Scikit-Learn**: Develop Machine Learning and Deep Learning Models with Python. [S.l.]: Packt Publishing, 2022. (Expert insight).

RDUSSEEUN, L.; KAUFMAN, P. Clustering by means of medoids. In: L1 NORM CONFERENCE, NEUCHATEL, SWITZERLAND, 1987. **Proceedings...** [S.l.: s.n.], 1987. v.31.

REYES, F. et al. Soil properties zoning of agricultural fields based on a climate-driven spatial clustering of remote sensing time series data. **European Journal of Agronomy**, [S.l.], v.150, p.126930, 2023.

ROBINSON, C. et al. Temporal Cluster Matching for Change Detection of Structures from Satellite Imagery. **CoRR**, [S.l.], v.abs/2103.09787, 2021.

ROCHA, A. M. D.; OLIVEIRA, M. A. D.; JOSé F. M., P.; CAVALHEIRO, G. G. H. ABP vs. OTAA activation of LoRa devices: an Experimental Study in a Rural Context. In: INTERNATIONAL CONFERENCE ON COMPUTING, NETWORKING AND COMMUNICATIONS (ICNC), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.630–634.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**, [S.l.], v.20, p.53–65, 1987.

ROY, T.; GEORGE K, J. Precision Farming: A Step Towards Sustainable, Climate-Smart Agriculture. **Global Climate Change: Resilient and Smart Agriculture**, Singapore, p.199–220, 2020.

RUDENKO, R. et al. A Brief Review on 4D Weather Visualization. **Sustainability**, [S.l.], v.14, n.9, 2022.

SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, [S.l.], v.26, n.1, p.43–49, 1978.

SALEEM, M. H.; POTGIETER, J.; ARIF, K. M. Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments. **Precision Agriculture**, [S.l.], v.22, 2021.

SATHIARAJ, D.; HUANG, X.; CHEN, J. Predicting climate types for the Continental United States using unsupervised clustering techniques. **Environmetrics**, [S.l.], v.30, n.4, p.e2524, 2019.

SAXENA, A. et al. A review of clustering techniques and developments. **Neurocomputing**, [S.l.], v.267, p.664–681, 2017.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval**. [S.l.]: Cambridge University Press Cambridge, 2008. v.39.

SERRÀ, J.; ARCOS, J. L. An empirical evaluation of similarity measures for time series classification. **Knowledge-Based Systems**, [S.l.], v.67, p.305–314, sep 2014.

SERVICE, F. A. **World Agricultural Production**. [S.l.]: United States Department of Agriculture (USDA), 2023. (Circular Series - WAP 10-23).

SHU, L.; HSIAO, B.; LIOU, Y.-H.; TSAI, Y.-L. On the Selection of Features for the Prediction Model of Cultivation of Sweet Potatoes at Early Growth Stage. In: IEEE SYMPOSIUM SERIES ON COMPUTATIONAL INTELLIGENCE (SSCI), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.2981–2988.

SIBSON, R. SLINK: An optimally efficient algorithm for the single-link cluster method. **The Computer Journal**, [S.l.], v.16, n.1, p.30–34, 01 1973.

SOHOULANDE, C. D.; STONE, K.; SZOGI, A.; BAUER, P. An investigation of seasonal precipitation patterns for rainfed agriculture in the Southeastern region of the United States. **Agricultural Water Management**, [S.l.], v.223, p.105728, 2019.

SOKAL, R. R.; MICHENER, C. D. A statistical method for evaluating systematic relationships. **University of Kansas science bulletin**, [S.l.], v.38, p.1409–1438, 1958.

STANčIN, I.; JOVIć, A. An overview and comparison of free Python libraries for data mining and big data analysis. In: INTERNATIONAL CONVENTION ON INFORMATION AND COMMUNICATION TECHNOLOGY, ELECTRONICS AND MICROELECTRONICS (MIPRO), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.977–982.

STEINLEY, D. K-means clustering: A half-century synthesis. **British Journal of Mathematical and Statistical Psychology**, [S.l.], v.59, n.1, p.1–34, 2006.

TADIć, L.; BONACCI, O.; BRLEKOVIć, T. An example of principal component analysis application on climate change assessment. **Theoretical and Applied Climatology**, [S.l.], 2019.

TAVENARD, R. et al. Tslearn, a Machine Learning Toolkit for Time Series Data. **J. Mach. Learn. Res.**, [S.l.], v.21, n.1, jun 2022.

THORNDIKE, R. L. Who belongs in the family? **Psychometrika**, [S.l.], v.18, p.267–276, 1953.

TIAN, W. et al. A survey on clustering based meteorological data mining. **Int. J. Grid Distrib. Comput**, [S.l.], v.7, n.6, p.229–240, 2014.

TIWARI, M.; MISRA, D. B. Article: Application of Cluster Analysis in Agriculture-A Review Article. **International Journal of Computer Applications**, [S.l.], v.36, n.4, p.43–47, December 2011. Full text available.

TORRESAN, S. et al. DESYCO: A decision support system for the regional risk assessment of climate change impacts in coastal zones. **Ocean Coastal Management**, [S.l.], v.120, p.49–63, 2016.

WANG, G.; JIA, L.; XIAO, Q. A Hybrid Approach Based on Unequal Span Segmentation-Clustering for Short-Term Wind Power Forecasting. **IEEE Tran. on Power Systems**, [S.l.], 2023.

WANG, S. et al. Exploring the optimal crop planting structure to balance water saving, food security and incomes under the spatiotemporal heterogeneity of the agricultural climate. **Journal of Environmental Management**, [S.l.], v.295, p.113130, 2021.

WARD, J. H. Hierarchical Grouping to Optimize an Objective Function. **Journal of the American Statistical Association**, [S.l.], v.58, n.301, p.236–244, 1963.

WEISSTEINER, C. J. et al. A Crop Group-Specific Pure Pixel Time Series for Europe. **Remote Sensing**, [S.l.], v.11, n.22, 2019.

WONG, T.-T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. **Pattern Recognition**, [S.l.], v.48, n.9, p.2839–2846, 2015.

YANG, J. et al. A Robust Method for Generating High-Spatiotemporal-Resolution Surface Reflectance by Fusing MODIS and Landsat Data. **Remote Sensing**, [S.l.], v.12, n.14, 2020.

ZHANG, Y.; LI, Y.; ZHANG, G. Short-term wind power forecasting approach based on Seq2Seq model using NWP data. **Energy**, [S.l.], v.213, p.118371, 2020.

ZHAO, J.; LIU, D.; HUANG, R. A Review of Climate-Smart Agriculture: Recent Advancements, Challenges, and Future Directions. **Sustainability**, [S.l.], v.15, n.4, 2023.