information

MDPI

Article

A Context-Based Perspective on Frost Analysis in Reuse-Oriented Big Data-System Developments

Agustina Buccella [1,*,†], Alejandra Cechich [1,†], Federico Saurin [1,†], Ayelén Montenegro [2,†], Andrea Rodríguez [2,†] and Angel Muñoz [2]

[1] GIISCO Research Group, Departamento de Ingeniería de Sistemas, Facultad de Informática, Universidad Nacional del Comahue, Neuquen 8300, Argentina; alejandra.cechich@fi.uncoma.edu.ar (A.C.); federico.saurin@fi.uncoma.edu.ar (F.S.)

[2] Instituto Nacional de Tecnología Agropecuaria (INTA), Alto Valle de Río Negro y Neuquén, Allen 8328, Argentina; montenegro.ayelen@inta.gob.ar (A.M.); rodriguez.andrea@inta.gob.ar (A.R.); munoz.angel@inta.gob.ar (A.M.)

[*] Correspondence: agustina.buccella@fi.uncoma.edu.ar

[†] These authors contributed equally to this work.

Abstract: The large amount of available data, generated every second via sensors, social networks, organizations, and so on, has generated new lines of research that involve novel methods, techniques, resources, and/or technologies. The development of big data systems (BDSs) can be approached from different perspectives, all of them useful, depending on the objectives pursued. In particular, in this work, we address BDSs in the area of software engineering, contributing to the generation of novel methodologies and techniques for software reuse. In this article, we propose a methodology to develop reusable BDSs by mirroring activities from software product line engineering. This means that the process of building BDSs is approached by analyzing the *variety* of domain features and modeling them as a family of related assets. The contextual perspective of the proposal, along with its supporting tool, is introduced through a case study in the agrometeorology domain. The characterization of variables for frost analysis exemplifies the importance of identifying variety, as well as the possibility of reusing previous analyses adjusted to the profile of each case. In addition to showing interesting findings from the case, we also exemplify our concept of *context variety*, which is a core element in modeling reusable BDSs.

Keywords: reusability; big data systems; variety identification; agrometeorology domain

check for updates

Citation: Buccella, A.; Cechich, A.; Saurin, F.; Montenegro, A.; Rodríguez, A.; Muñoz, A. A Context-Based Perspective on Frost Analysis in Reuse-Oriented Big Data-System Developments. *Information* **2024**, *15*, 661. https://doi.org/10.3390/info15110661

Academic Editor: Vincenzo Moscato

Received: 28 August 2024
Revised: 2 October 2024
Accepted: 17 October 2024
Published: 22 October 2024

1. Introduction

Success in global markets for software-intensive products depends, among other things, on shortening the time to market, as well as improving quality and evolution or reducing the resources required. This aspect can be addressed using external or internal strategies. External strategies include acquiring off-the-shelf components and having development, maintenance, and support performed by third parties. In-house strategies include global software development, for which resources are geographically distributed according to specific needs, profiles, conditions, or costs [1], and software product line (SPL) engineering and ecosystems, i.e., *the strategic acquisition, creation, and reuse of software assets* [2–4]. Our research focuses on the latter context—that of engineering product lines and ecosystems. This engineering differs from traditional systems' development in two fundamental respects [5]:

- It requires two distinct development processes: domain engineering and application engineering. The first defines and models the commonality and variability, establishing a platform for rapidly developing quality applications within the line. Application engineering derives specific applications by strategically reusing that platform and exploiting its variability.

- It needs to explicitly define and manage variability. During domain engineering, variability is introduced into all software assets, such as domain requirements, architectural models, components, or test cases. During application engineering, it is exploited to derive large-scale customized applications according to the needs of customers and markets.

In this context, we can ask ourselves whether it would be possible to think about developing reusable big data systems so that common and variable aspects are treated as in an SPL. This would afford us the advantages of separating the complexity of BDS design into domain and application analysis while, at the same time, allowing us to detect the influences of these domains on the results obtained via the systems. Investigating the literature, we see that there are several proposals that consider reuse during the development of big data systems (BDSs). For example, the work [6] analyzes reusability concepts in the context of the use and reuse of data, and the work described [7] addresses privacy aspects when data are reused. It proposes a reuse taxonomy to delimit when data can be used/shared and how to provide privacy. Other proposals have contributed to maximizing data reuse by promoting the use of shared technologies such as cloud computing or shared storage [8,9]. On the other hand, although there are several works in the literature on reusability, almost all of them only focus on promoting data exchange, sharing, and reusing, but not on the development of reusable software artifacts. In spite of in other areas of computing, reusability is a key factor to guarantee systems with a higher quality, a better time-to-market, lower development costs, etc.; in the development of big data systems, this quality is still emerging.

Only one work relates to ours closely. In [10], the detection of common and variable aspects within the development of BDSs is incorporated as a way of building *families of systems*. The work presents a reference architecture that allows system designers to achieve the following: (1) define requirements—the reference architecture identifies significant requirements and shows variations according to the type of requirement; (2) develop and evaluate solutions—the architecture identifies modules that must be developed in order to allow a certain required capacity; and (3) integrate systems—existing systems can be mapped to the modules of the reference architecture, which results in the easy identification of conflict points where the interoperability between systems must be worked on.

According to this previous scenario, and differently from [10], in this work, we propose to take advantage of one of the most important engineering approaches promoting reusability: software product line (SPL) engineering [3]. It has emerged as a paradigm for domain-oriented software construction to obtain faster-developed and high-quality software applications. Considering that the SPL development focuses on domain-oriented reuse, its success depends on the identification, use, and administration of artifacts inside those domains; therefore, the application of specific techniques for systematizing reuse becomes crucial. In this work, we propose adaptations to/extensions of resources created using the SPL paradigm to support the particularities of BDSs. The first and more important adaptation is to approach BDSs as domain-oriented systems, that is, thinking of BDSs as applications within a domain while developing software artifacts that can be reused across them.

As a domain-oriented proposal, in this article, we present each of its steps through case studies in a specific domain. We focus on the agrometeorology domain (the agrometeorology domain was extracted from one of the most popular ontologies in agriculture, named AGROVOC—developed by the Food and Agriculture Organization (FAO) https://agrovoc.fao.org/browse/agrovoc/en/ accessed on 16 October 2024),which "deals with all the weather-sensitive elements of agriculture production", such as pollination, animal migration, weather risk assessments, etc.

Our cases address the problem of frost, which is a climatic phenomenon that poses a constant threat to crops around the world. It is critical to understand the differences between two types of frost that affect crops: (1) radiation frost and (2) advection frost. The first occurs on clear and calm nights, leading to the formation of a thermal inversion where temperatures near the surface drop below freezing. There are several methods for protecting a plant from this type of frost. On the other hand, the second occurs when a mass of cold air suddenly invades a warmer area, which can also be accompanied by winds and clouds. In this case, there is limited protection; to mitigate frost damage, it is crucial to prevent and predict this weather phenomenon.

Frost prevention and prediction involve the use of early warning information about the weather conditions that lead to frost, allowing farmers to take protective measures in a timely manner. By receiving early warnings, farmers can make informed decisions that can make the difference between catastrophic losses and a successful harvest. Therefore, understanding the characteristics of frost and taking proactive measures are essential to ensure food security and economic well-being in regions affected by this climate phenomenon [11].

Currently, we are working on the problem of late frosts damaging fruit production by developing a BDS for detection and/or prevention in a particular area of north Patagonia (the valley of Rio Negro and the Neuquen rivers is an area that produces apples and pears for export, so the quality of production is essential for its economy). This is a known problem that several works in the literature have faced [12–15]; however, as far as we know, there are no proposals that include reuse issues in this domain.

Thus, the main contributions of this paper are summarized as follows: (1) a reuse-oriented big data methodology based on principles of software product line development, (2) the development of domain and reusable case assets within the agrometeorology domain, and (3) a case of reuse considering previous developments in the domain.

This article is organized as follows. The next section describes our reuse-oriented big data methodology, together with the software artifacts to be developed and the components to be used. Section 3 describes the development of two domain and reusable cases within the agrometeorology domain by following a contextual (top-down) approach. In Section 5, we highlight lessons learned about the development of the reusable cases and methodology. Finally, we address conclusions and future work.

2. Materials and Methods

Traditional software and decision-making support (DMS) systems, including BDSs, must collaborate with each other to reach the goal of having efficient AI-engineered software systems, which should be able to process a huge amount of data, as well as perform traditional organizational operating processes [16]. Developing both perspectives is like thinking of a two-sided conceptualization. On the one hand, the software system must satisfy operational stakeholders' needs; on the other hand, the DMS system (using data analytics) must achieve accuracy and precision for decision making. Quality properties, such as reusability, are also addressed by the two systems; however, the different conceptualizations make researchers rethink the way these properties should be modeled.

By mirroring variability management and domain-oriented development in SPLs, we might conceptualize data variety for BDSs as a set of cases that should be abstracted according to their different varieties. Our methodology for reuse-oriented big data systems takes advantage of SPL engineering to define common artifacts with variabilities that will be instantiated during product development (Figure 1). Thus, as in SPL, we define a two-phase development: (1) the *domain Engineering*, for developing common assets of a domain, and (2) *application engineering*, for developing case applications from the above common assets.

Both phases include the basic activities of the *big data process* based on a *domain problem*. For instance, we could wonder whether *climate influences frosts differently, depending on the environment, in order to discover similar (or different) climatic factors that may be relevant.* This problem is divided into domain and case development, and it is guided by our definition of *variety*. Varieties are like alternatives to driving each activity within domain engineering, for example, by detecting different environmental conditions, features, sources, etc. Then, during application engineering, these alternatives (varieties) are selected to build the domain or reusable products.

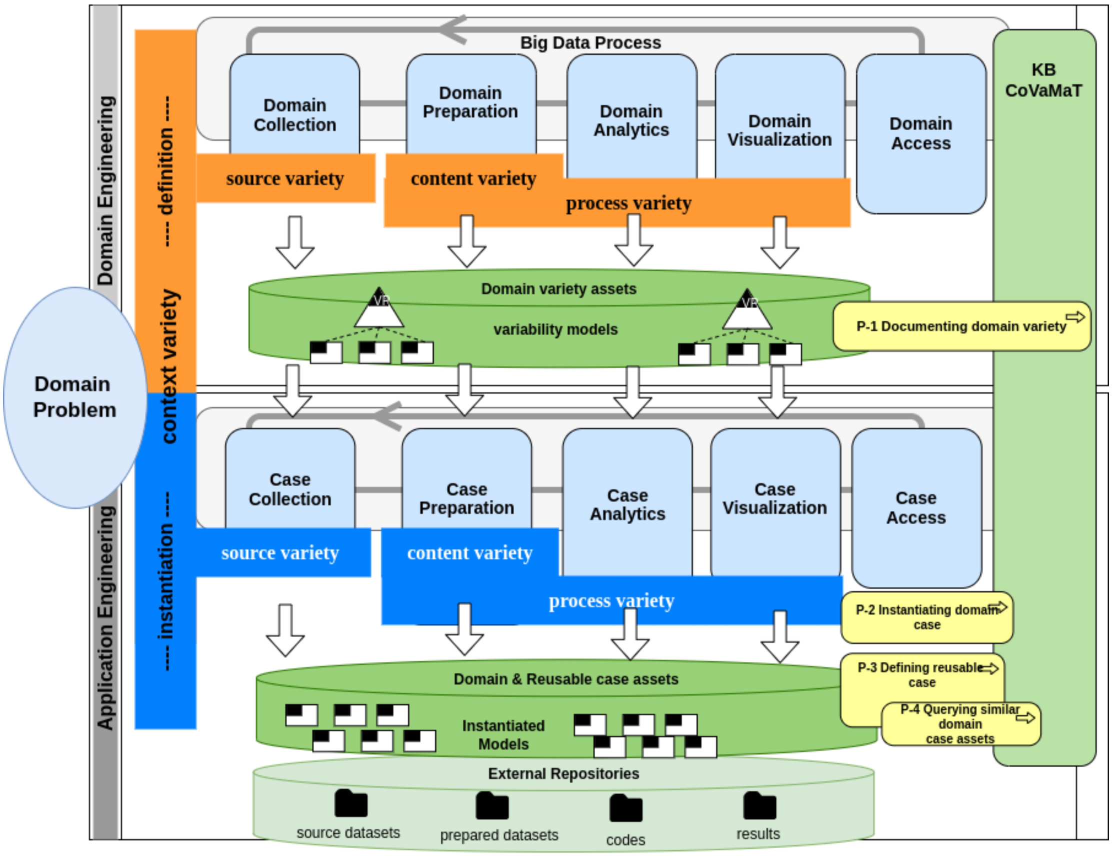


Figure 1. Reuse-oriented big data methodology.

Our concepts of variety. In previous works, we have defined four types of varieties [17,18]: (1) *source variety*, denoting the different types of data sources that can be useful for the domain problem, such as structured, semi-structured, and without structure, and the collection methods, such as publish–subscribe, event processing, etc.; (2) *content variety*, identifying variables or factors extracted from sources that are useful when considered as an analysis feature; (3) *process variety*, gathering information on techniques for data cleaning and data analytics; and (4) *context variety*, denoting the situation or conditions under which the problem domain can be applied.

For instance, we could set a case with specific weather conditions and soil (context variety), collect data from structured datasets (source variety), select relevant features for the case—temperature, rainfall, etc. (content variety), and analyze data using different correlation techniques (process variety). For both types of engineering, *context variety* is the main artifact providing the basis for reuse. It defines domain variations that will constrain and affect the results of the process.

Let us introduce an example. The different types of contexts depend on each domain; for instance, for the hydrology domain, water bodies can be classified as rivers, lakes, seas, groundwater, etc., and water flows can be influenced by climatology, geology, and so on. Particularly, the state of practice for predicting groundwater level fluctuations as a function of meteorological variables is extensive. For example, in the study in [19], general trends in local climate variation are examined in the Winnipeg, Canada, area, analyzing the relationship between these trends and groundwater level fluctuations. From the data analysis, it is concluded that these levels show a strong correlation with both *annual precipitation* and *average annual temperature*.

At this point, we could ask ourselves whether these variables are equally relevant in any similar analysis of groundwater level fluctuations. Then, from another analysis carried out in the Mediterranean basin, where the Institute of Atmospheric Sciences and Climate of the Italian National Research Center (ISAC-CNR) has taken measurements of fluctuations of an aquifer in the karst zone [20], it was concluded that *precipitation* and *air humidity* are the significant variables. Clearly, the relevant variables are different for both cases. This might indicate that variables affecting groundwater fluctuations (content variety) might be influenced by specific contextual features (context variety). The Mediterranean area is forested, with an average annual rainfall of 1.67 mm (2017–2022) and fA (sandy loam) soils, very fast permeability, and wells located in the vicinity of urban areas. We consulted the opinion of experts, and it was deemed reasonable to expect those relations in this context; i.e., the most significant meteorological variables will be those indicated for the Mediterranean area, differently from the area in Canada, which is a low-lying flood plain with an extremely flat topography (the level of rainfall added to the vegetation makes soil moisture very relevant in the Mediterranean area).

Variety influencing the process. During the *domain engineering phase*, the activities of a big data process are influenced by variety in different ways (top of Figure 1):

- *Domain collection* refers to ways of collecting data and datasets that can be used. *Source variety* here helps detect different data structures, acquisition techniques, etc.
- *Domain preparation* refers to the cleaning and preparation tasks necessary to transform datasets into more suitable ones. The *content variety* helps determine the variables to be considered. *Process variety* detects and defines preparation techniques needed for the domain.
- *Domain analytics* involves the data analysis techniques and the processing types (batch, real-time, and interactive). *Process variety* helps detect variations in data analysis techniques.
- *Domain visualization* refers to visualization alternatives to prepare results that will be shown to users. *Process variety* determines visualization techniques.
- *Domain access* defines the ways in which users can access the results.

Representing variety. All the previous varieties defined are modeled using variability models as a simplification of the Orthogonal Variability Model (OVM) notation [3], which allows for documenting *variation points* and their *variants* as datasheets, including the following:

- *Variability subject*: A variable element of the real world or a variable property of such an element. Example: the color of a car.
- *Variability object*: A particular instance of a subject of variability. Example: red, green, pink, etc.
- *Variation point*: a representation of a subject of variability within domain artifacts enriched with context information.
- *Variant*: a representation of an object of variability.

Datasheets are used to model the design functionalities of SPLs; however, in the case of variety modeling in BDSs, we applied a simplified view. In particular, *variation points (VPs)* and *variations* were limited to the case of exposing a structure where a VP will have a tag and a list of variations. For simplicity's sake, the different relationships that may exist between these elements were not used, nor were the optional points, alternative points, etc. As an example, Figure 2 shows a content variety datasheet, where the *weather variables'* variation point is the root of the hierarchy, and there is one variant for each possible variable type (wind chill, wind speed, high temperature, and so on).

The supporting tool. Models are stored in a knowledge base named the *Context-Based Variety Management Tool (CoVaMaT)* [17]. The domain varieties are stored as *domain-variety assets* through the P-1 process (*documenting domain variety*) of CoVaMaT.

During the *application engineering phase* (bottom of Figure 1), the activities include the same tasks as before, but for the development of a specific case. In this phase, firstly, *domain-case assets* are generated by instantiating variation points of the domain (defined during domain engineering) and storing these cases so that they are ready for reuse. They are created via the P-2 process (*instantiating domain case*) of CoVaMaT. Once domain cases exist, *reusable case assets* are generated by searching for similarity and reusing those stored *domain-case assets*. These activities are performed via the P-3 (*defining reusable case*) and P-4 (*querying similar domain or case assets*) processes.

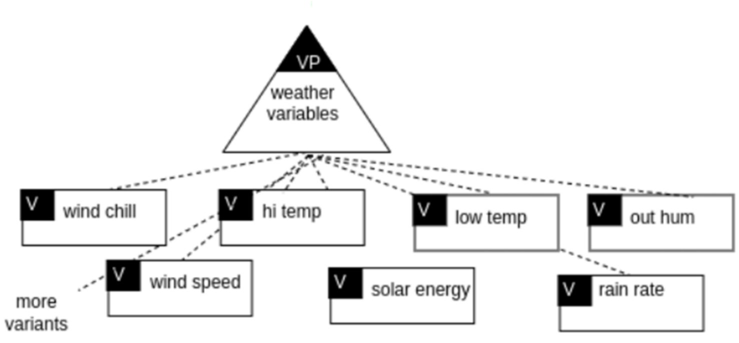


Figure 2. Datasheet for recording content variety: an example.

At the same time, during the application engineering phase, we also associate instantiated variants to the files used/generated during the execution of the previous activities of the big data process (datasets, files, codes, etc.). These files are temporarily stored in an external repository. At the moment, as CoVaMaT is an ongoing project, it does not have the functionality to manage files. However, we structure an external repository with specific folders to manage these files manually. At the bottom of Figure 1, we can see the four folders created: (1) *source datasets* for storing datasets and files during the *case collection* activity, (2) *prepared datasets* for storing the transformed datasets using the techniques defined, (3) *codes* for uploading the codes (in general with Python) used for preparation, analytics, and visualization tasks, and (4) for storing the results as graphics, papers, reports, etc.

Our variety identification process. In previous works [17,18], we have proposed a variety identification process based on bottom-up and top-down approaches. These approaches allow data analytics and domain experts to identify the four types of varieties while developing big data activities. The approaches are not divided into a domain and an application because both phases of engineering are carried out iteratively; that is, domain assets might also be created during the development of domain cases. Thus, each execution of the five activities of the big data process creates *domain-variety assets* while developing *domain* or *reusable case assets*.

After the problem definition (1) (i.e., identifying influences on frosts to improve predictions), the top-down approach (Figure 3) begins with (2) the formulation of hypotheses by an expert in the domain (*what hypothesis do I have?*), based on his or her prior knowledge, which guides the subsequent analysis of data to validate those assumptions. Then, data analysts begin to perform the big data process (3) to analyze data according to the defined hypotheses (do the data corroborate?). Later, these results are returned to validate these hypotheses (4), possibly visualizing the results in different ways, and to allow for its reformulation or the completion of the activities of the process, alternatively (5). As a result of this approach, specific domain cases are generated that store the identified variations, which can be reused in future instances, thus promoting efficiency and consistency in future analyses within the same domain.

On the other hand, after (1) defining the problem, the B-VIP approach begins with (2) the decision to conduct an exploratory study on the domain problem (what do the data say?). Then, (3) data analysts begin data analysis without predefined hypotheses, allowing patterns and trends to emerge naturally through a process of big data. The results obtained during this study are returned and (4) subjected to a process of validation and discussion with experts in the field. Then, (5) a decision can be made to end the process or reformulate the search. The documentation in this approach focuses on the variety that emerges during the exploratory study. The domain cases instantiated during this process are stored to be reused in future queries and analyses, thus contributing to the accumulation of knowledge and efficiency in decision making.

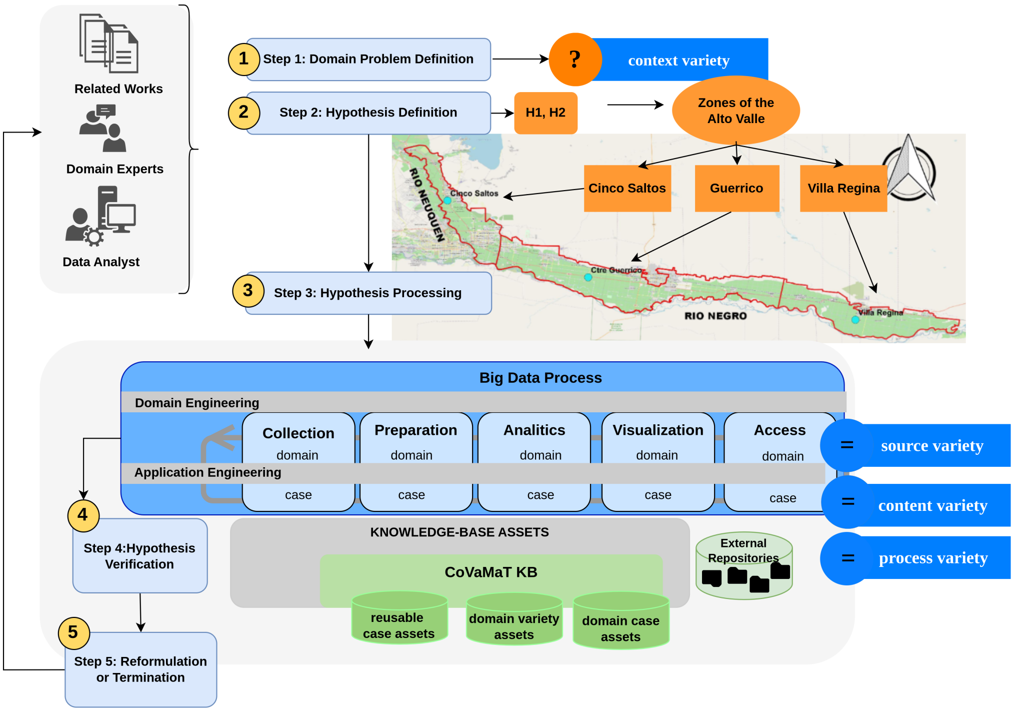


Figure 3. Top-downapproach to the development of domain and reusable cases in the agrometeorology domain.

3. Development of Two Application Cases in the Agrometeorology Domain

In this section, we describe the application of the top-down approach to the development of two domain cases in the agrometeorology domain (Figure 3). The approach consists of five main steps. In general terms, it starts from the definition of a domain problem (step 1), where the context variety is determined. The domain problem here should be researched as deeply as possible. To do so, the stakeholders involved in this first step are not only expert users and data analysts but also information available for this domain, such as standards, related work, previous studies, etc. All of this information will help draw the hypotheses for a particular case's development (step 2). During the third step, the activities of the domain and engineering phases for our big data process should be carried out. As we described in the previous section, we do not build domain assets independently. Both domain and application activities are performed iteratively, based on the development of domain cases. This means filtering domain information through the relevance determined by the cases. In addition, at this point, three varieties (source, content, and process) must be identified and documented. Finally, steps 4 and 5 verify the hypotheses and terminate or reformulate the application case, respectively.

The next parts of the article describe the activities performed at each step during the development of the two domain cases.

Step 1: domain problem definition

We currently work with expert users of the National Institute of Agricultural Technology (INTA) at the experimental station in Alto Valle of Río Negro and Neuquén (https://www.argentina.gob.ar/inta/cr-patagonia-norte/eea-altovalle accessed on 16 October 2024). Alto Valle is a region located in the north of Patagonia, which includes the Neuquén, Limay, and Río Negro rivers. It is a productive area of approximately one hundred thousand hectares, with an estimated annual production of seven hundred thousand tons of pears and apples, destined mainly for export and for the concentrated juice industry. In particular, the institute is dedicated to research on plant entomology and therapeutics, plant nutrition, plant pathology, irrigation and drainage, the post-harvest period, crop management, etc.

The Alto Valle of Río Negro and Neuquén is an area that can be sub-delimited by three specific zones according to the location of three main weather stations. In the right side of Figure 3, we can see these stations and the covered zones, named Villa Regina (https://sipan.inta.gob.ar/agrometeorologia/met/40/clima.htm accessed on 16 October 2024), Guerrico (https://sipan.inta.gob.ar/agrometeorologia/met/10/clima.htm accessed on 16 October 2024), and Cinco Saltos (https://sipan.inta.gob.ar/agrometeorologia/met/45/clima.htm accessed on 16 October 2024). The figure shows the delimited zones and the geographic points of the weather stations in light blue (located in the center of each zone). Although the zones are near each other, they have different characteristics with respect to climate, soil, and vegetation.

At the same time, we analyzed information provided in related work from the literature on the detection and prediction of frosts within the agrometeorology domain.

The analysis of these works provides an overview of the main goals, source data, variables, and results obtained in order to think about solutions to early frost detection. Table 1 shows a summary of eight works from the literature described by goals, sources, periods of study, analysis techniques, and results. Although this is not an exhaustive list of all possible works in the literature, it provides a broad overview of the importance of mining data on frost events in order to help determine mechanisms to protect crops from frost.

Table 1. Summary of some related works about frost detection and prediction.

Works	Objectives	Sources/Period	Analysis	Results
Verdes et al. [21] 2000 (Rosario, Argentina)	Develop an empirical prediction system for frost protection of fruits and vegetables	2601 daily records of a weather station in Zavalla (from 1973 to 1989)	Artificial neural networks (ANNs), simple Bayes (SB), k-nearest neighbors (k-NNs)	SB generated better results than ANNs and k-NNs in frost detection, but it obtained a higher probability of false detection
Ding et al. [22] 2019 (Japan)	Construction of predictive models to capture possible causal relations between environmental factors and frost	Sensors in a local area from 2017 to 2019	Correlation Support vector machine (SVM)	Temperature is the most important factor in predictive models; humidity helps an early alarm (2–3 h ahead), and radiation improves the sensitivity of the response in a short period
Ding et al. [23] 2020 (Japan)	Develop a causal model for understanding relationships between past and future frost events	Sensors in a local area in 2017	Support vector regression (SVR) Structural causal model (SCM)	Better results were obtained with labeled data and the task of learning as classification

Table 1. *Cont.*

Works	Objectives	Sources/Period	Analysis	Results
Zhou et al. [12] 2022 (Australia)	Increase the prediction frequency from once per 12–24 h for the next day or night's events to minute-wise predictions for the next hour's events	Sensors from different weather stations (from 2016 to 2017)	Recurrent neural networks (RNNs) Long short-term memory (LSTM) Gated recurrent unit (GRU) Artificial neural network (ANN)	LSTM is more suitable for a dataset of the current year; ANN models were faster than RNN-based models, with better accuracy and performance
Kotikot et al. [24] 2020 (Kenia)	Integrate local characteristics of the land surface with weather variables to map frost hotspots	282 observations during frost seasons (2011 and 2012), NASA-LIS data, MODIS sensor, and radar topography data	Multicollinearity Binary logistic regression Correlation	The model identified areas of high frost susceptibility using a threshold based on the probability of the logistic model; it was useful for managing frost-damage mitigation efforts
Gobbett et al. [13] 2020 (Australia)	Generate future climate scenarios (from 2030 to 2050) to predict night-time minimum temperatures	MODIS with night-time temperatures, elevation, terrain indices, and meteorological station data	Multivariate adaptive regression splines (MARSs)	Elevation and time-of-the-year variables affect frost occurrences; they found a decrease in frost at lower elevations under the tested future climate scenarios
Diedrichs et al. [14] 2018 (Mendoza)	Frost prediction at least 24 h ahead with SMOTE technique	Five meteorological stations during 2001–2006	Bayesian networks (BNs) Random forest (RF) Logistic regression (LR) Binary trees (BTs)	The application of an oversampling technique showed improvements in RF and LR models
Talsma et al. [15] 2023 (Alcalde)	Application of regression models to predict temperature and not just the occurrence of frost	10 years of historical data from the Natural Resource Conservation Service weather station	Random forest (RF) Deep neural network (DNN) Convolution long short-term memory neural network (CNN)	CNN showed better results, and the use of soil temperature is a key parameter in longer-term predictions' application (>24 h)

Highlighting aspects of the table, we can see that source data generally come from weather stations. But in some cases, studies also include satellite imagery for elevation and/or soil characteristics. These source data determine the set of variables used for the studies. In general, for sources from weather stations, variables include temperature, wind, humidity, dew, rain, and solar radiation. In addition, when data are obtained from sensors, variables are related to topographic aspects such as elevation, convexity, etc. For example, in [24], the variables are obtained from NASA's Land Processes Distributed Active Archive Center (LP DAAC), and NASA-LIS (https://lis.gsfc.nasa.gov/ accessed on 16 October 2024) was used to identify relative soil moisture, wind speed, and specific humidity. Another example, presented in [13], uses terrain metrics and vegetation indexes extracted from the MODIS LST sensor (https://modis.gsfc.nasa.gov/data/dataprod/mod11.php accessed on 16 October 2024).

The preparation of datasets before analytics involves its own particularities in several studies [12–15]. In general, datasets are normalized and prepared to contain temperatures of certain time lapses, such as 1 h, 1 day, or sometimes more than one week, in order to train the models on events that occurred some time before. With these datasets and the analysis techniques chosen, works can predict the minimum temperature at multiple prediction lead times, such as 6–48 h [12], 24 h [15], etc.

Then, when addressing the analysis techniques and methods applied in the studies, we can see that applying different types of neural networks is the most frequent approach; for example, recurrent neural networks (RNNs) in [12], artificial neural networks (ANNs) in [12,21], and convolutional neural networks (CNNs) in [15]. Also, some studies apply a random forest (RF) [14,15] and compare their behaviors and results against those of other techniques, such as neural networks or logistic regression (LR) [14].

Finally, regarding the results, in addition to evaluating the analysis techniques with respect to accuracy and performance [12,14,15,24], the works remark that some of the most relevant variables involved in frost prediction are temperature, humidity, radiation, and elevation.

In particular, in our work, user experts were interested in analyzing historical data to predict the behavior of frosts and the influence of weather in the three zones (Regina, Cinco Saltos, and Guerrico). Like in any other productive area, frosts cause great damage to fruit plants, and during the spring season, this damage is really serious because leaves and flowers are emerging. Therefore, we defined the main objective of the study as follows:

> ***Analyze the influence of weather variables for late spring frosts in Alto Valle of Río Negro and Neuquén***

The objective was addressed as a *context variety* problem based on the zones (in orange in Figure 3). Thus, the first activity was to apply the P-4 process *(P-4, querying similar domain-case assets)* of CoVaMaT to find out whether there is a similar domain-variety asset stored. In this case, the inputs of the process are the *agrometeorology domain* (extracted from AGROVOC), the context variety based on *zones*, and the main objective defined. At this point, CoVaMaT returned results indicating that there was no similar domain stored, so we used the P-1 process *(documenting domain variety)* to document the three zones. As we can see in Figure 4, through this process, we created the *agrometeorology-domain asset* with the *associated main objective*, and the variation point (*zones*) with its variants (*Villa Regina zone*, *Guerrico zone*, and *Cinco Saltos zone*).

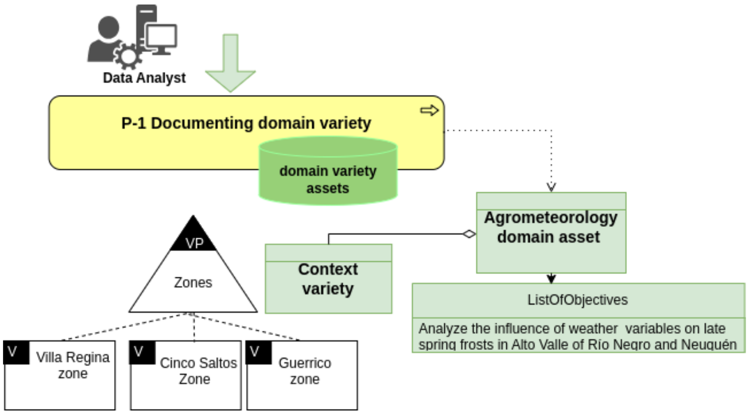

Data Analyst
P-1 Documenting domain variety
domain variety assets
Agrometeorology domain asset
VP
Zones
Context variety
ListOfObjectives
Analyze the influence of weather variables on late spring frosts in Alto Valle of Río Negro and Neuquén
V
Villa Regina zone
V
Cinco Saltos Zone
V
Guerrico zone

Figure 4. P-1 process of CoVaMaT documenting the *agrometeorology-domain asset*, together with the *context variety* for the objective.

It is important to highlight that, at this point, the *agrometeorology-domain asset* had only the associated *context variety* because the other three varieties were defined later, during the development of domain cases. Thus, in the next subsections, we describe the development of two domain cases based on the context variety for Villa Regina and Guerrico. In particular, during *Domain Case 1—Villa Regina*, we developed *domain-case assets* and *domain-variety assets* in the context of the analysis of the Villa Regina zone. And in *Domain Case 2—Guerrico* for the Guerrico zone, we reused the previously developed assets in creating new *reusable case assets*. Cinco Saltos is not described in the article due to space limitations. However, for the development of *Domain Case 3—Cinco Saltos*, we similarly reused the assets of the agrometeorology domain, as well as the assets of Villa Regina and Guerrico domain cases.

3.1. Domain Case 1—Villa Regina

Here, we describe the next four steps (from 2 to 5) of the top-down approach performed during the development of the domain case for the Villa Regina context (Figure 3).

Step 2: hypothesis definition

For the hypothesis definition, we analyzed not only the information provided by expert users at INTA but also information from related works in the literature (Table 1).

From expert users, we obtained relevant information about the specific needs of the zone being analyzed. For example, they started from the assumption that temperatures have a strong influence on frosts, as described in some works in the literature [12,22]. However, expert users were not aware of the specific influences of other weather factors, such as wind and rain. For example, they argued that "when it is raining or there is a strong wind, there is no frost." However, they also knew that this is not always the case, and it depends on the wind speed and rainfall. Even in the related works presented in [22,24], the authors analyzed the influences of these factors without finding strong relations or conclusive results that influence frost.

Considering these previous concerns, we defined the first hypothesis as follows:

Hypothesis 1. *Do wind and rain factors influence late frost events?*

Another important aspect for the expert users was to know in which way weather factors affect frosts before (minutes, hours, or even days) an event occurs. Thus, our second hypothesis was defined as follows:

Hypothesis 2. *How do specific weather factors influence frosts before they occur?*

This concern was also studied in the literature [12,15].

Step 3: hypothesis processing

We performed the five activities involved in the big data process (in blue in Figure 3) interacting with CoVaMaT for the documentation and recovery of the domain and reusable assets (in green in Figure 3).

Big data process: collection activity. The source dataset was obtained from the Villa Regina Weather Station, located in the same city, which stores 33 variables in 10 min intervals from 5 July 2009 to 17 September 2019, with a total of 511.263 records.

During the collection activity for this domain (in the domain engineering phase), we stored (through the P-1 process of CoVaMaT) the *source variety* associated with the *agrometeorology-domain asset* (previously created in Figure 4). The *source-variety asset* was documented with one variant (*weather station of Villa Regina*), as we can see in Figure 5.

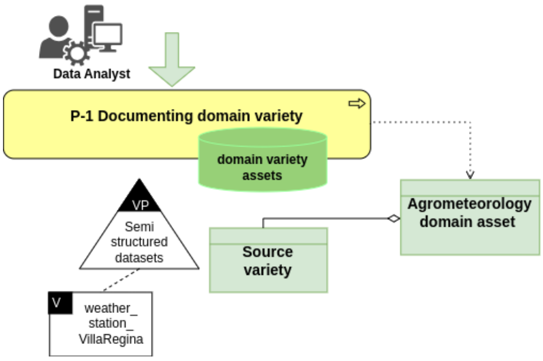

Data Analyst
P-1 Documenting domain variety
domain variety assets
Agrometeorology domain asset
VP
Semi structured datasets
Source variety
V
weather_ station_ VillaRegina

Figure 5. P-1 process of CoVaMaT documenting the *source variety* associated with the *agrometeorology-domain asset*.

Big data process: preparation activity. In this activity, we created two types of datasets. The first one, named *Original Dataset Villa Regina*, was generated by considering information given by expert users according to the variables that could influence frost events. Thus, we considered only 14 variables (Table 2). Also, we cleaned the dataset by removing/replacing some outliers and null values (we used specific Python libraries to do so, such as Pandas (https://pandas.pydata.org/ accessed on 16 October 2024) and KNNImputer (https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html accessed on 16 October 2024).

The second dataset was built by adding a new continuous variable to the original dataset (so, this dataset contains 15 variables.), named the *FrostCont* (in our analyses, we consider "frost" the occurrence of a temperature less than or equal to 0 °C) variable, that contains the value of the low-temperature n records ahead. Therefore, we could generate several datasets of this type by setting the number of records in advance. This means that, as the dataset stores one record each of 10 min, $n = 6$ stores the low temperature corresponding to 1 h later, $n = 12$ corresponds to 2 h later, and so on. In Table 3, we can see part of one dataset generated, in which the *LowTemp* variable is used to complete the *FrostCont* variable, depending on the n value. Then, these datasets were named *Dynamic Dataset Villa Regina(n)*, depending on the n value.

Table 2. Selected variables for the *Original Dataset Villa Regina*.

Variable	Description	Format
HiTemp	High temperature	Numeric
LowTemp	Low temperature	Numeric
OutHum	Out humidity	Numeric
DewPt	Dew point	Numeric
WindSpeed	Wing speed	Numeric
WindRun	Distance of traveled wind	Numeric
WindChill	Cooling effect of the wind	Numeric
Bar	Atmospheric pressure	Numeric
Rain	Rain presence	Categoric
RainRate	Rain average	Numeric
SolarRad	Solar radiation	Numeric
SolarEnergy	Solar energy	Numeric
Hi SolarRad	Max solar radiation	Numeric
ET	Evapotranspiration	Numeric

Table 3. Examples of the *Dynamic Dataset Villa Regina(n)* with (**a**) $n = 6$, (**b**) $n = 12$, and (**c**) $n = 138$.

	LowTemp	FrostCont
1	0.5	1.0
2	0.7	1.6
3	0.6	1.6
4	0.5	1.3
5	0.4	0.8
6	0.4	0.2
7	1.0	−0.6
8	1.6	−0.7
...	...	...
13	−0.6	−1.5
14	−0.7	−1.5
...	...	...
139	5.3	2.8

(**a**) With $n = 6$—1 h

	LowTemp	FrostCont
1	0.5	−0.6
2	0.7	−0.7
3	0.6	−0.7
4	0.5	−0.8
5	0.4	−0.9
6	0.4	−1.1
7	1.0	−1.5
8	1.6	−1.5
...	...	...
13	−0.6	−1.0
14	−0.7	−1.3
...	...	...
139	5.3	10.1

(**b**) With $n = 12$—2 h

	LowTemp	FrostCont
1	0.5	5.3
2	0.7	5.1
3	0.6	4.7
4	0.5	3.8
5	0.4	3.1
6	0.4	2.1
7	1.0	1.9
8	1.6	1.8
...	...	...
13	−0.6	0.8
14	−0.7	0.7
...	...	...
139	5.3	14.8

(**c**) With $n = 138$—23 h

The last step was selecting values with similar occurrences of frost events and non-frost events in the late-frost period (from July to September) to avoid the risk of overfitting.

Finally, with these two types of datasets created, we again documented assets of the domain engineering phase. Here, we added new domain varieties for both *content*- and *process*-variety assets. Thus, in Figure 6, we can see a new variation point (*weather variables*) with 14 variants (Table 2) for the *content-variety asset* and another variation point (*cleaning and preparation*) for the techniques and procedures used to identify and treat outliers, nulls, and overfitting.

Big data process: analytics and visualization activities. Through these activities, we performed several analyses in order to verify the hypotheses defined.

Considering Hypothesis 1 (*Do wind and rain factors influence late frost events?*), we firstly explored the influence of the 13 variables with respect to low temperatures. To do so, we identified correlations by applying two coefficients: (a) Pearson and (b) Spearman (Kendall was discarded because the coefficient is more suitable for small datasets, and in our analysis, the results were always far different.) [25]. In Figure 7, we show the results (the order of the variables is the same for the two coefficients), for which we can see only a few variations between them. For example, for Pearson, the *DewPt* variable returned 45%, and for Spearman, 48%. In general, Spearman produced slightly better results (≈1–2%). Also, in the figure, we can see that *HiTemp* (≈99%), *WindChill* (≈98%), *WindSpeed* (≈47%), *WindRun* (≈47%), *Hi SolarRad* (≈44%), *SolarEnergy* (≈43%), *SolarRad* (≈43%), and *DewPt* (≈40%) obtained the best results. On the other hand, *Bar* (≈−38%) and *OutHum* (≈−61%) showed a considerable negative influence. The rest of the variables did not obtain significant correlations.

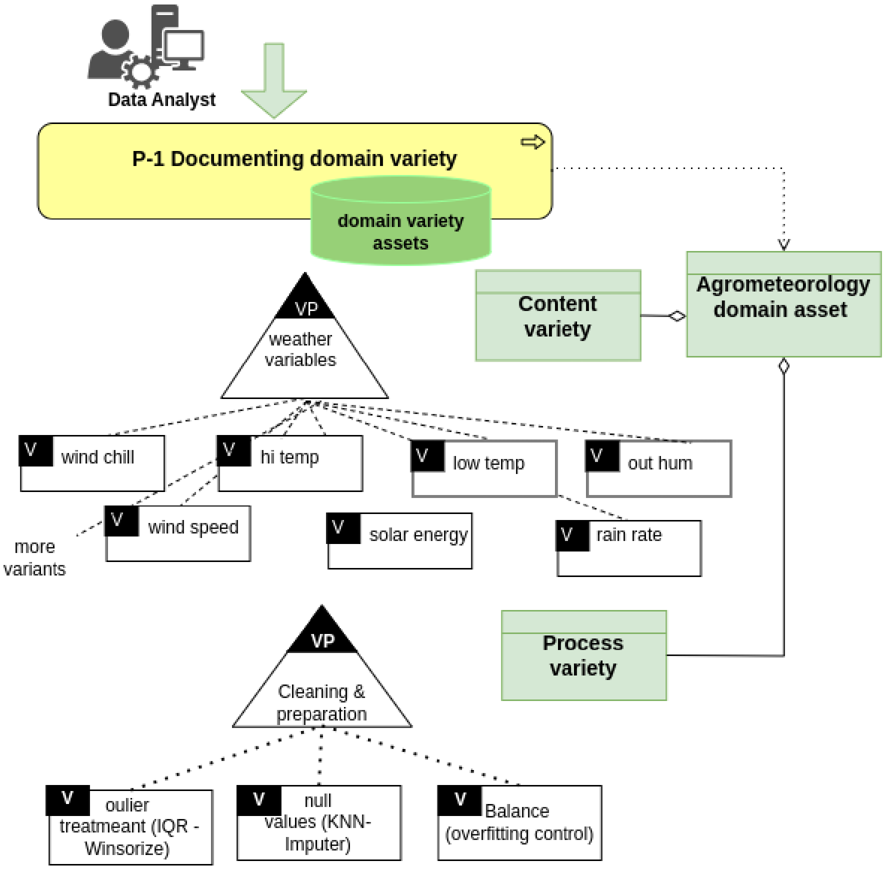


Figure 6. P-1 process of CoVaMaT documenting the *content* and *process variety* assets.

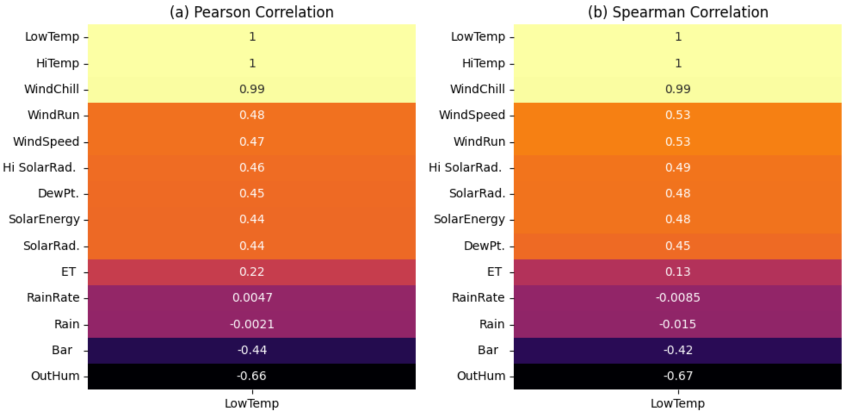
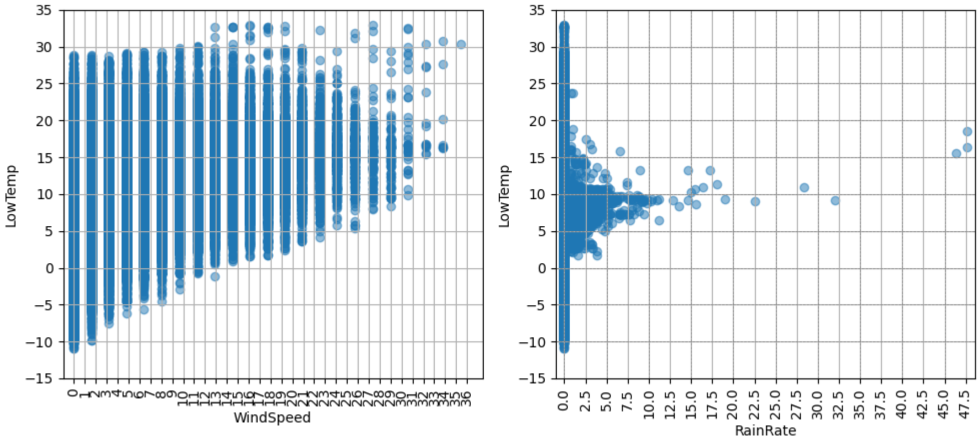


Figure 7. Correlation analysis for the *Original Dataset Villa Regina*.

Considering Hypothesis 1, with respect to the *RainRate* (stored in millimeters), we can see that it does not have any influence on the low temperature, with a result of ≈0% in the correlations. The *WindSpeed* variable obtained a moderate result (≈47%), but this value is not good enough either. As these coefficients did not present us with a clear panorama depicting their influences on frosts, we decided to analyze them separately and only with frost events (temperatures below 0 °C). In this way, we analyzed and reorganized these variables in order to obtain more insight into their behavior. For example, in Figure 8, scatter graphs show the conditions of frosts, depending on the value of the variables. Figure 8a) shows that, when the wind speed is greater than 13 km/h, the temperature is never below 0 °C, and consequently, there is no frost. Also, in Figure 8b), the graph shows that, when rainfall is greater than 2.5 mm, the temperatures remain over 0 °C. In this way, although we obtained moderate and low correlations between the variables using the coefficients, we can see here that wind speed and rainfall have a great influence on frost events in this zone. Increasing rainfall or wind speed generates low probabilities of frosts, delimited by specific thresholds.



(a) Scatter graph: WindSpeed and LowTemp (b) Scatter graph: RainRate and LowTemp

Figure 8. Scatter graphs analyzing rain rate and wind speed with respect to frosts.

Next, we considered Hypothesis 2 (*How do specific weather factors influence frosts before they occur?*). In this case, we analyzed the 13 variables using the *Dynamic Dataset Villa Regina(n)* from $n = 0$ to $n = 138$ for a period of 24 h. Thus, we generated 24 different datasets and compared each other with the low temperature. In Figure 9, we show the results, in which we can see the influences of the different variables during the 24 h. For example, *SolarRad* (graphic 11) has a moderate/high influence (≈70%) between the first 2 and 4 h, which decreases in the next 20 h before starting to increase again at the 23rd h. Something similar happens to *HighTemp*, *DewPt*, *WindSpeed*, etc. On the other hand, such variables as *OutHum* and *Bar* exhibit an inverse behavior, but with very low values during the 24 h.

Once again, from the complete analysis of variables and correlations, we could not obtain significant conclusions about rain and wind. So, we reorganized the data across the 24 datasets in order to evaluate the wind speed and rainfall needed to achieve temperatures over 0 °C. In Figure 10, we show the relationship between wind and frosts, in which the points denote the minimal wind speed for the absence of frost (the x-axis represents hours, and the y-axis the independent variable (*windspeed*). Each point shows the minimum measurement of this variable for the absence of frost). For example, given a particular time, if, 2 or 3 h before, there was a wind speed of 14.5 km/h, there would be no frost. The wind speed was tending to stabilize from 14 to 23 h at a speed of 32.20 km/h.

We performed the same analysis for the *RainRate* variable in Figure 11. The graph shows that, given a particular time, the absence of frost required 0.8 mm of rain from 0 to 5 h before, versus 1.00 mm from 6 to 12 h before, and this value increased until 2.2 mm 23 h before.

Finally, we documented all the assets of this activity in CoVaMaT by creating new *process-variety assets* in the domain. In Figure 12, we can see three new variation points for (1) *processing*, defined with a *batch* variety due to our processes being performed in batch mode, (2) *correlation*, defining the two coefficients used for the analytics, and (3) *visualization*, denoting the graphs used to organize and analyze the information on frosts, wind, and rain.

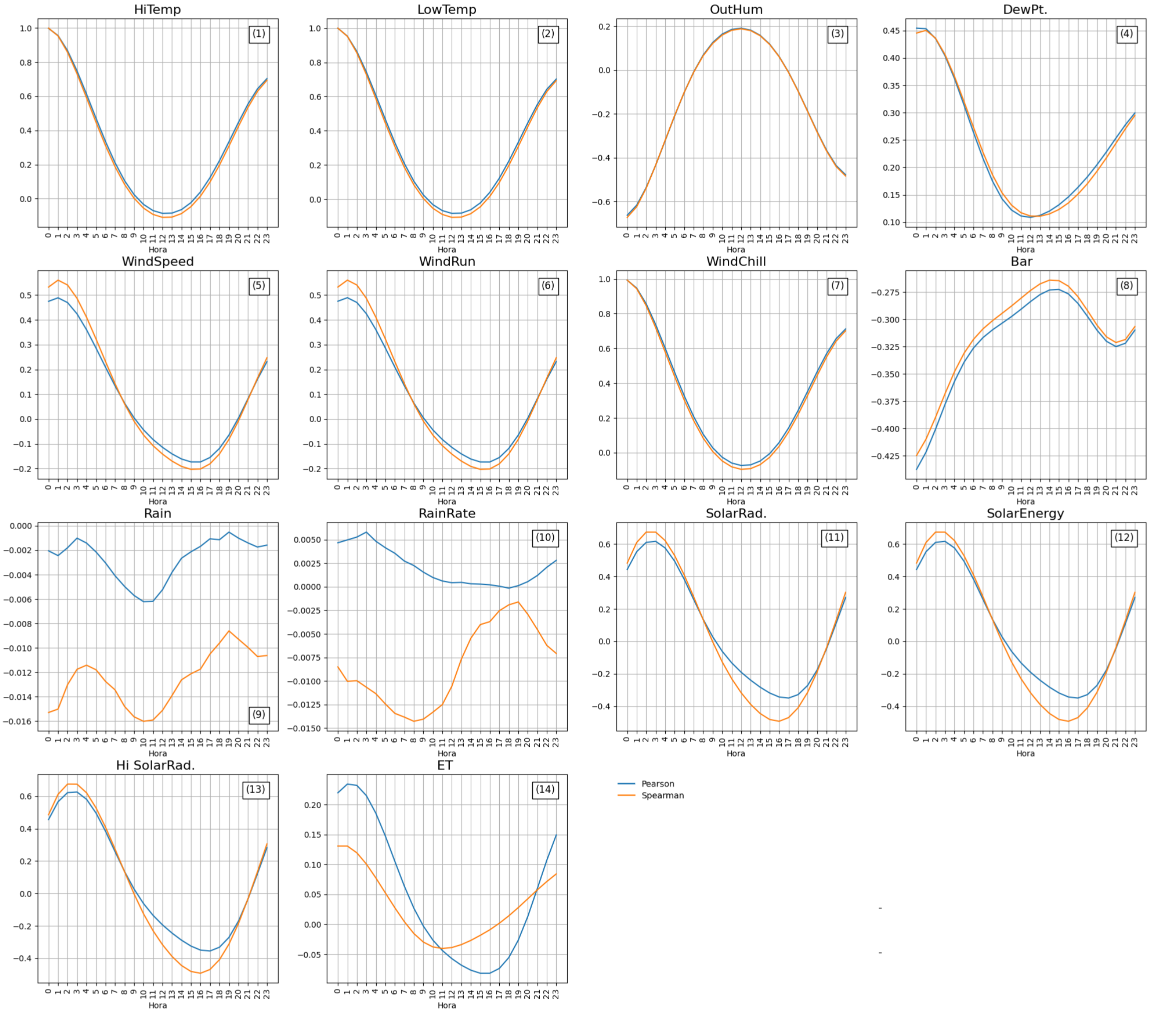


Figure 9. Correlation analysis of *Dynamic Dataset Villa Regina(n)* from $n = 0$ to $n = 138$ according to Pearson (blue) and Spearman (orange).

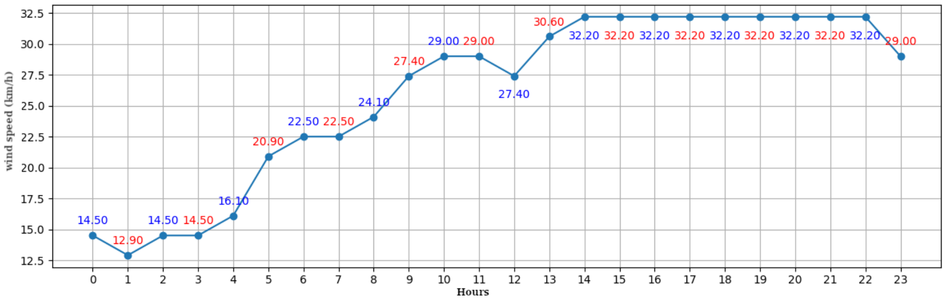


Figure 10. Relationship between *WindSpeed* and *FrostCont* variables, considering the *Dynamic Dataset Villa Regina(n)* from $n = 0$ to $n = 138$.

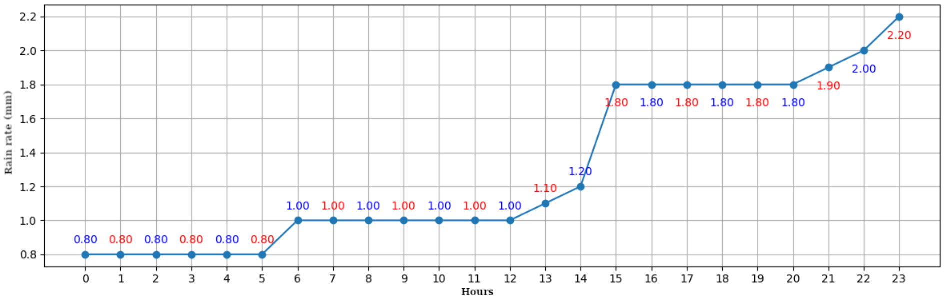


Figure 11. Relationship between *RainRate* and *FrostCont* variables, considering the *Dynamic Dataset Villa Regina(n)* from $n = 0$ to $n = 138$.

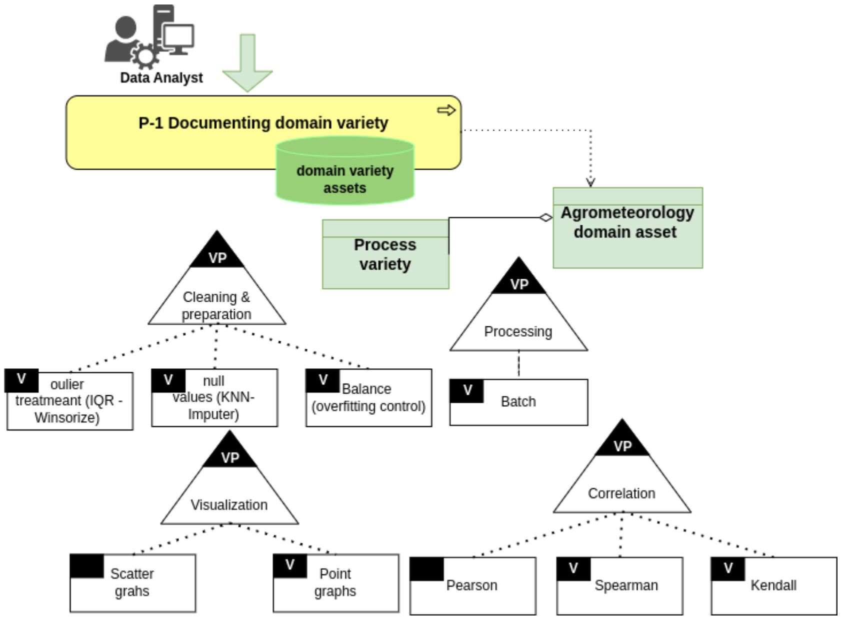


Figure 12. Adding more *process-variety assets* with variants for the processing type, analysis and, visualization techniques applied.

Step 4: hypothesis verification

In this step, we analyzed the information provided via the previous activities to verify (or not) our two hypotheses (Hypotheses 1 and 2). In both cases, we divided the verification into two branches: (1) a general one, relating all the weather variables to low temperatures, and (2) a secondary one, analyzing only rain and wind with respect to temperatures below 0 °C (frost events).

In Hypothesis 1, the results of the first verification showed high/moderate positive correlations between *HiTemp*, *WindChill*, *WindSpeed*, *HiSolarRad*, and *DewPt* and moderate negative correlations between *Bar* and *OutHum*, with respect to *LowTemp*. Then, in the second branch, we obtained very different information, showing a strong relationship between wind–rain and frosts. Here, we could determine which thresholds of wind speed and rainfall were necessary to avoid temperatures below 0 °C.

Secondly, in Hypothesis 2, for the 24 h analysis, the first verification applying correlations showed fluctuations between moderate and low values, but the same relations as in Hypothesis 1 remained. Then, the analysis of rain and wind revealed more interesting relationships, which determined the specific thresholds of wind speed and rainfall that should happen *n* hours earlier to avoid frosts in a given time.

Step 5: reformulation or termination

At this point, for *Domain Case 1—Villa Regina*, we decided to finish. On one hand, we had stored the assets for the agrometeorology domain; that is, we had performed the domain engineering activities. On the other hand, for the creation of the domain case, in the application engineering, we applied the P-2 process *(P-2, instantiating domain case)* for a variety of instantiations of the created assets. The instantiation consisted of selecting only the variants used in the Villa Regina case (VR). At the same time, we also associated the variants with the files used/generated during the execution of the activities (datasets, files, codes, etc.). These files were stored in an external repository (Figure 1).

In Figure 13, we can see the instantiated varieties of *Domain Case VR*, together with these files; for example, the *weather_station_Villa_Regina* variant is associated with the *weather_station_VillaRegina.csv* in the source datasets folder. Also, the variants related to the preparation activity (oultiers, null values, and balance variants) are associated with *Original Dataset Villa Regina* and 24 datasets, from $n = 0$ to $n = 138$, corresponding to *Dynamic Dataset Villa Regina(n)*, all stored in the prepared dataset folder.

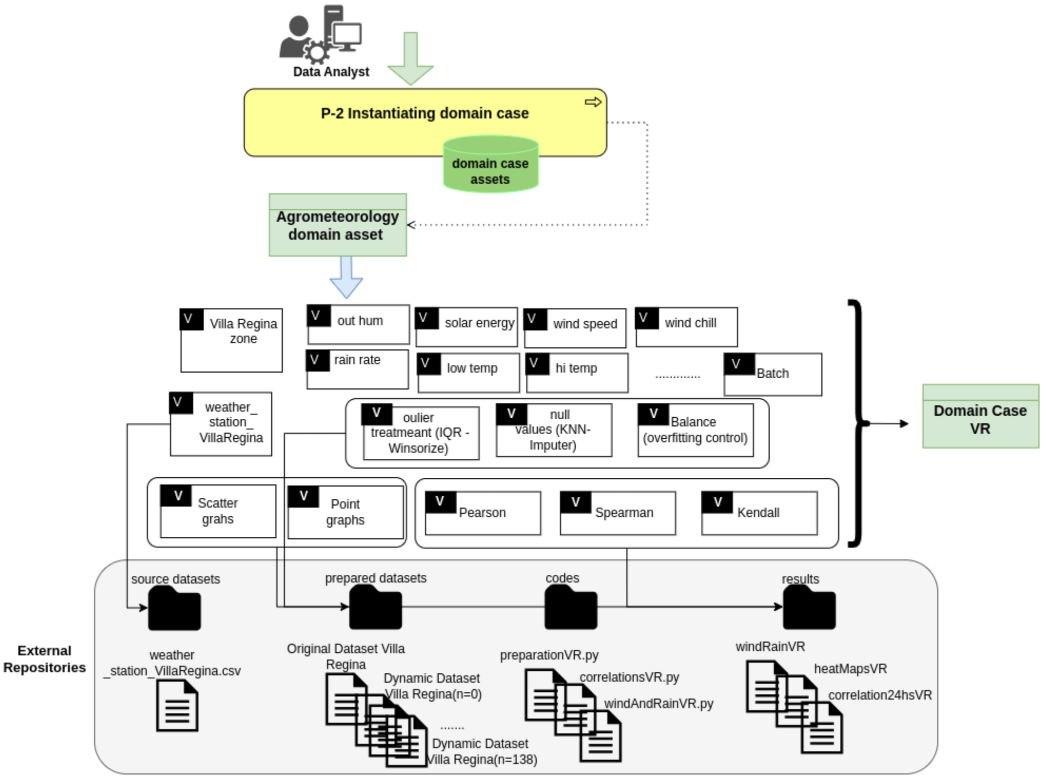


Figure 13. *Domain case VR* created for the instantiation of the *agrometeorology-domain asset*, together with associated files (during the application engineering phase).

3.2. Domain Case 2—Guerrico

As we describe at the beginning of this section, this second case is built with a zone-based contextualization for the agrometeorology domain under study (Villa Regina and Guerrico—Figure 4). During the development of the Villa Regina domain case, we stored and documented all the *agrometeorology-domain assets* of the first zone, and this information is now available for reuse during the development of this new case.

In Figure 14, we show how the assets were recovered from the repositories. The first task applied the P-4 process *(P-4, querying similar domain-case assets)* of CoVaMaT to find out whether there was a similar domain-variety asset stored. We set as inputs the *agrometeorology domain* (extracted from AGROVOC), the context variety with the *Guerrico zone*, and the main objective (labeled as 1 in Figure 14). The process searched in the repositories of *domain-variety assets* (labeled as 2) and *domain-case assets* (labeled as 3) for similar assets, recovering the *agrometeorology-domain assets* and *Domain Case VR*, respectively.

Next, we continued with the last four steps of the top-down approach (Figure 3) for developing this new domain case. The first step was already performed by defining the context variety (of the Alto Valle zones) in the domain.

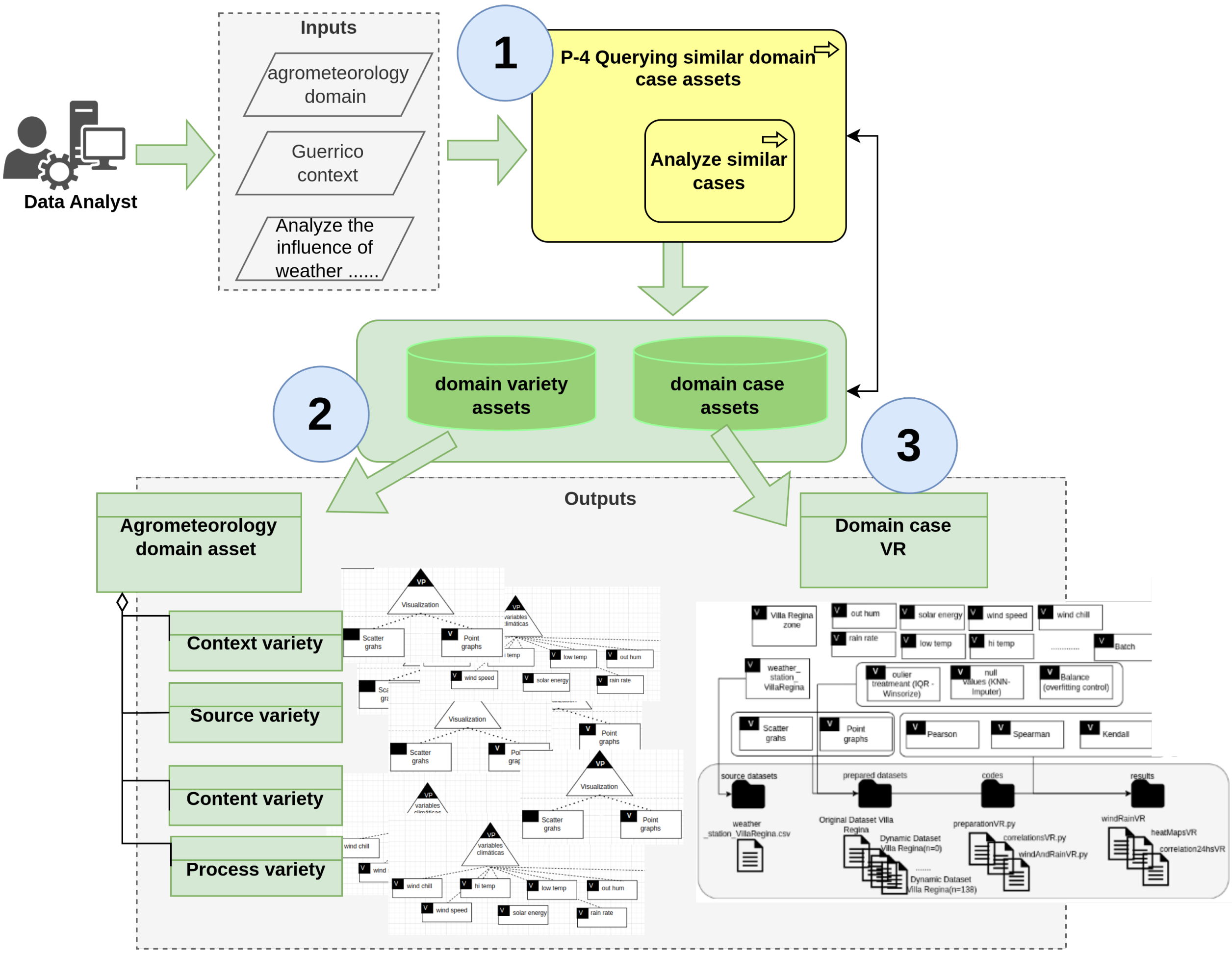


Figure 14. P-4 process retrieving stored assets: (1) query assets, (2) search for *domain-variety assets*, and (3) search for *domain-case assets*.

Step 2: hypothesis definition

We defined here the same two hypotheses as in the Villa Regina case, and we added a new user expert's requirement. Users wanted to analyze the different behaviors of weather variables in different zones (as we documented in CoVaMaT as a *context variety* in Figure 4). As Guerrico is a colder zone than Villa Regina, with more frosts in spring, user experts wanted to know about variations in the behavior of some weather variables. In other words, we addressed the influence of context on reusable variables during BDS development. So, we added a new hypothesis:

Hypothesis 3. *Do the variables analyzed for Villa Regina have the same influence in Guerrico with respect to frosts?*

Step 3: hypothesis processing

We again performed the five activities of the big data process. In order to simplify the description of the work, we highlight here the new tasks performed and the documentation/creation/reuse of new assets (in CoVaMaT).

For the documentation of the new case (Figure 15), we applied the P-3 process (*defining reusable case*) of CoVaMaT, which already had the assets recovered via steps (2) and (3) in Figure 14. In Figure 15, the 1 label denotes the creation of the *Gerrico reusable case*, and the 2 label represents the instantiation of the *Guerrico zone* for the context variety.

Next, during the ***collection activity*** of the big data process, we created a new *source-variety asset* representing the *Guerrico weather station* as part of the *semi-structured dataset* variation point (labeled as 3 in the figure). As this variant did not exist previously, it was firstly created via the P-1 process as part of the *source-variety asset* (of the *agrometeorology-domain asset*) and then instantiated to the reusable case; therefore, here, we interacted with the two phases (domain and application). We can see this new variant in orange in the figure. Also, in this activity, we stored the *weather_station_Guerrico.csv* file in the source dataset folder.

Next, as we applied the same techniques as in the ***preparation activity*** for the Villa Regina domain case, we only instantiated existing variants in the reusable case. Thus, in this activity (labeled s 4 in Figure 15), we created the new datasets *Original Dataset Guerrico*, with 14 weather variables, and the 24 *Dynamic Dataset Guerrico(n)* datasets with the new *FrostCont* variable containing the value of the low-temperature n records ahead (from $n = 0$ to $n = 138$).

During the ***analytics activity*** of the big data process, we applied the same techniques and processes as in Villa Regina. We only created a new variation point (*Comparison*), including one variant to perform a *percentage change analysis* [26,27]. It was added via the P-1 process (domain phase) and instantiated (application phase) through the P-3 process (labeled as 5 in the figure).

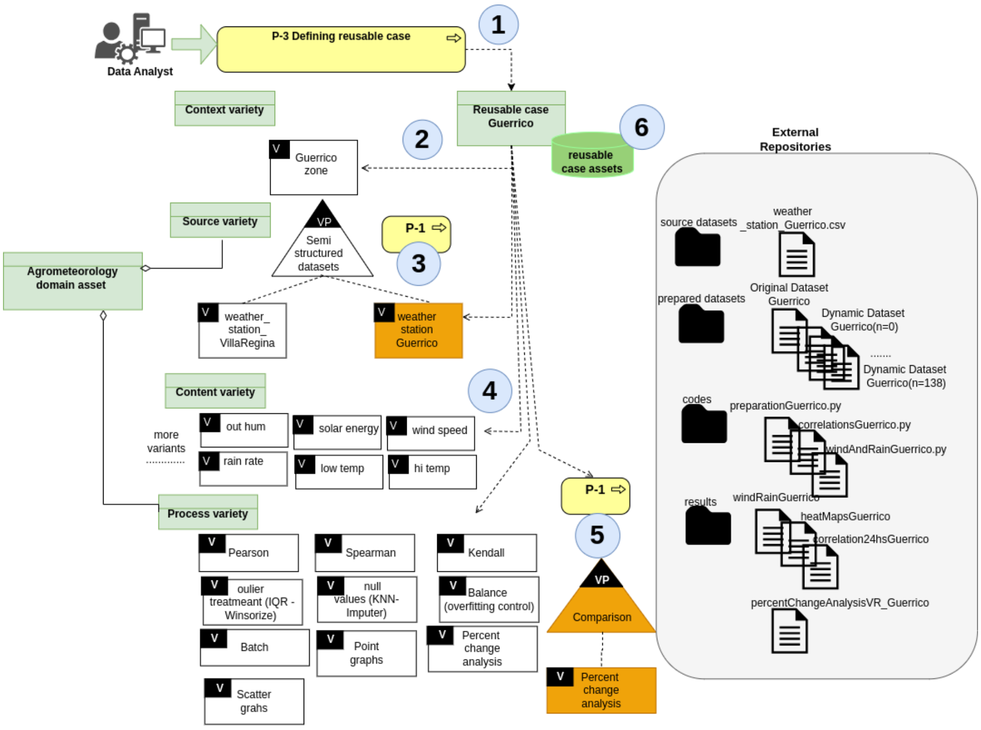


Figure 15. P-3 process for documenting *Reusable Case Guerrico* from the *agrometeorology-domain assets*.

For Hypothesis 1, in the ***analytics activity***, we performed the correlation analysis of the 13 variables with respect to low temperature. Pearson and Spearman coefficients were again applied, obtaining some similar results with respect to high, moderate, and low influences as in Villa Regina. Also, for Hypothesis 2, the application of the coefficients for the 24 h period generated some similar behaviors. Next, for the analysis of the wind and rain for both hypotheses, we also noted some similar results, but with different impacts. For example, in Figure 16, which denotes the relationship between wind and frosts, we can see, unlike Figure 10 for Villa Regina, that a wind speed greater than 24.10 km/h is necessary 4 h before a given time for the absence of frost, and a speed greater than 41.8 km/h 8 h before (considering both events separately).

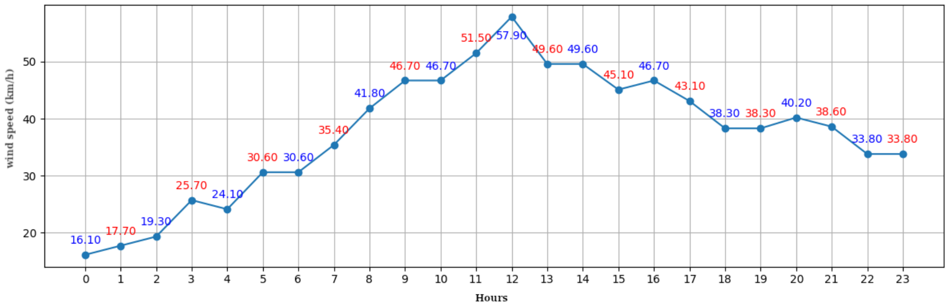


Figure 16. Relationship between *WindSpeed* and *FrostCont* variables considering *Dynamic Dataset Villa Regina(n)* from $n = 0$ to $n = 138$ in Guerrico.

In order to verify Hypothesis 3, we applied the *percentage change analysis* for figures obtained from the two zones—Villa Regina and Guerrico.

In Figure 17, we show this analysis for the two correlation coefficients. In the graph, a positive result indicates that the Guerrico zone obtained a higher value than Villa Regina for the variable and coefficient, and a negative result indicates the inverse; that is, Villa Regina obtained a higher coefficient for the variable. We can see, in some cases, significant differences between both zones. For example, *Bar* presented positive variations of more than 22%, and *SolarEnergy* more than 9%. On the other hand, *DewPt* generated a negative percentage change with ($\approx -19\%$) or ($\approx -11\%$) for *OutHum*. Other variables, such as *WindSpeed*, *WindRun*, *HiTemp*, and *WindChill*, registered a low percentage change, denoting no variation between the two zones. Something really different occurred with *RainRate* and *Rain*, for which Spearmean coefficients generated highly positive variations.

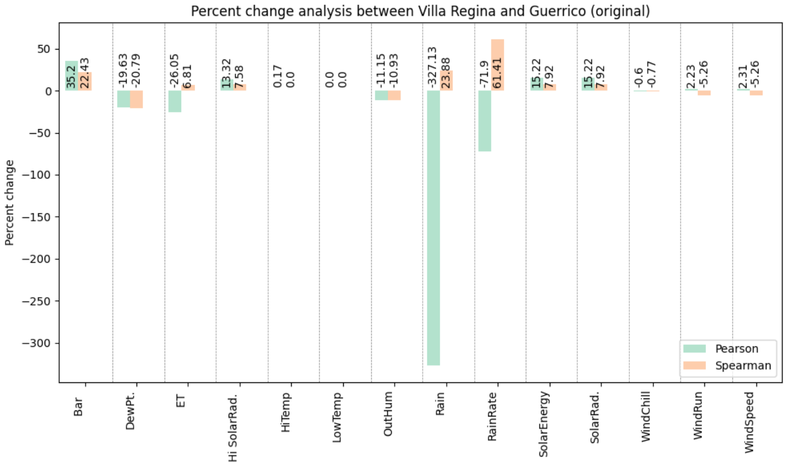


Figure 17. *Percentage change analysis* between Villa Regina and Guerrico for Hypothesis 1 of the 13 variables wrt low temperature

Similarly, in Figure 18, we show the *percentage change analysis* for both dynamic datasets (*Dynamic Dataset Villa Regina (n)* and *Dynamic Dataset Guerrico(n)* from $n = 0$ to $n = 138$).

As we can see, *WindSpeed*, *WindRun*, *HiTemp*, and *WindChill* again obtained low percentage changes.

Comparing the percentage changes between original and dynamic datasets (Figures 17 and 18), we can observe that original datasets generated fewer variations in coefficients of both zones. At the same time, in some cases, positive variations of the first datasets were negative in the dynamic datasets. This happened specifically for wind variables (the last three in both figures). Also, for rain variables, in both cases, we observed great differences among the two coefficients.

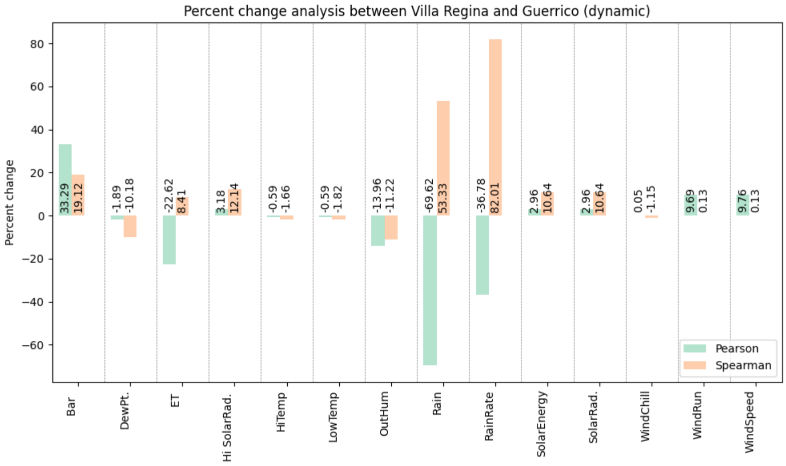


Figure 18. *Percentage change analysis* between Villa Regina and Guerrico analyzing a 24 h period.

From a deeper analysis, in Figure 19, we again show the *percentage change analysis* of wind and rain variables between both zones during a 24 h period. The figure shows the percentage differences needed in Guerrico with respect to Villa Regina to avoid frost. As we can see, all percentages are positive, indicating that Guerrico needs more wind speed and rain to prevent frost events. For example, with respect to *WindSpeed* (Figure 19a), Guerrico needs 49.69% more wind speed 4 h before a given time than Villa Regina, and 11 h before, it needs 111.31%. With respect to *RainRate* (Figure 19b), in the first 3 h, there is no difference between zones; however, between the 4th and 14th h before, Guerrico needs more than 60% of rainfall to avoid frost.

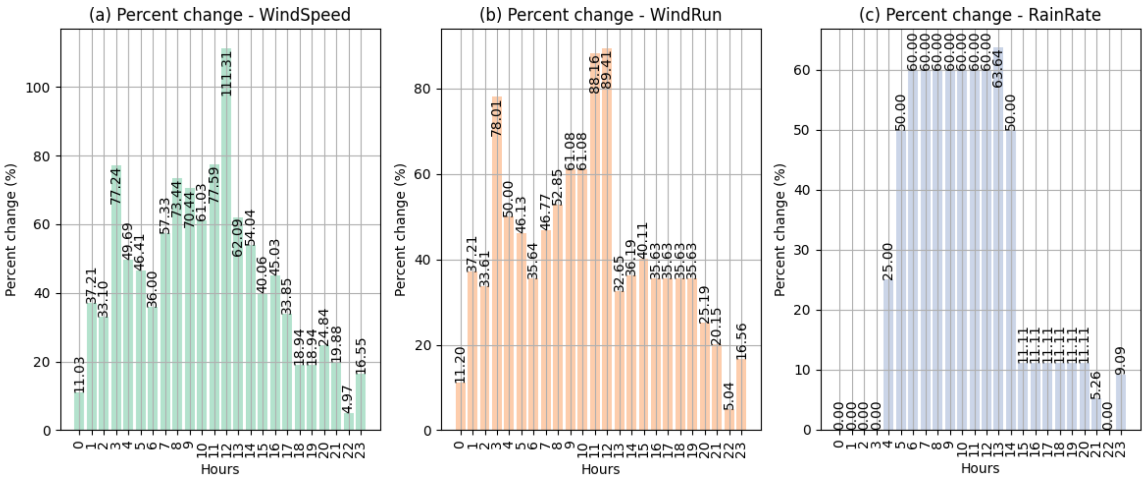


Figure 19. *Percentage change analysis* for (**a**) wind speed, (**b**) wind run, and (**c**) rain rate variables in a 24 h period for Villa Regina and Guerrico,

Step 4: hypothesis verification

In this step, we analyzed the information provided during the previous analysis to evaluate the three hypotheses. The first two produced similar results regarding which variables had more or less influence on frost events. The main differences were specifically based on the fact that percentages were always greater in Guerrico than in Villa Regina. This fact was then verified through Hypothesis 3.

From the analysis of these three hypotheses, we can conclude the following:

- Guerrico has *lower min temperatures* than Villa Regina.
- Guerrico is *less influenced* by *high temperatures, the dew point, evapotranspiration, and humidity* than Villa Regina in terms of frosts. Guerrico is *influenced more* by *atmospheric pressure, solar energy, solar radiation, wind speed, wind chill, and rain* than Villa Regina in terms of frosts.
- Guerrico requires *between 11.03% and 111.31% higher wind speeds* than Villa Regina to prevent frosts.
- Guerrico requires *between 11.2% and 89.41% more wind chill* than Villa Regina to prevent frosts.
- Guerrico requires *at least 63% more rain* than Villa Regina to prevent frosts.

Step 5: reformulation or termination

In this step, we decided to finish *Domain Case 2—Guerrico*. We applied the P-2 process for variety instantiation, which stored the case in the repository as a *reusable case asset* (labeled as 6 in Figure 15).

3.3. Domain Case 3—Cinco Saltos

As aforementioned, this domain case is not described here due to space limitations. However, since it was considered for evaluating the proposal, it is important to highlight three main characteristics of this case:

- Cinco Saltos was developed as a reusable case, and it was documented and stored in CoVaMaT as *Reusable Case Cinco Saltos* (by applying P-3 and P-4 processes).
- The hypotheses defined in Cinco Saltos were the same as in Guerrico (three hypotheses). Thus, we could reuse, in addition to the assets developed for the agrometoeorology domain, the assets for the Villa Regina and Guerrico cases. In the case of the assets defined for *Reusable Case Guerrico*, we could reuse the assets created for the third hypothesis regarding the influencing variables in the three zones.
- We did not have to add any new domain assets (within the agrometerology asset) because they could be completely reused.

4. Results

We analyzed results from two different points of view. On the one hand, we compared our case studies to related works in Table 1. Like the works [14,22–24], we defined a main objective to analyze the influence of weather variables on late frosts. We used similar weather variables as relevant features, concluding that temperature was the most influential factor for frost occurrence. Differently from those studies, we did not use information about elevation or soil types. But the most relevant difference was revealed with respect to wind and rain since none of the related studies found strong influences. We found not only that these factors were important for analysis (at the time they occur and in the hours before) but also the differences between their thresholds in both zones (Villa Regina and Guerrico). We concluded that the influences depend on the context, and we determined specific thresholds in which the wind and rain should be analyzed. These results can be considered as starting points to define new hypotheses for new domain-case applications that allow expert users to detect or prevent frost.

On the other hand, as in any other methodology that proposes software reuse, it was necessary to evaluate the *reusable component development* (https://insights.sei.cmu.edu/library/a-framework-for-software-product-line-practice-version-50/ accessed on 16 October 2024). This means we should analyze the time required to develop the services of the reusable components, which are pre-built pieces of software that can be used in multiple applications. In our case, reusable components encapsulate functionality as big data processing services in order to reduce the amount of work needed for the application development phase, that is, the creation of domain and reusable case assets.

To clarify the notion of service in the context of the BDS development process carried out in our case studies, in Table 4 (in the table, times are defined in days), we detailed the set of seventeen services/tasks (from S0 to S16), along with the days required for each activity: Domain Problem Definition, Collection, Preparation, Analysis, Visualization, and Access. Services were analyzed by dividing the cases into four categories, including information from all developed cases (Villa Regina, Gerrico, and Cinco Saltos):

- *Traditional application case Villa Regina* (TCaseVR) looked at the time required for the development of the activities of the big data process without documenting variability, like a traditional big data application. It was the base case for comparison.
- *Domain-case application Villa Regina (DCaseVR)* looked at the time required when the T-VIP approach was applied, so identifying variety became an additional activity alongside the traditional process of the previous case.
- *Reusable Case Guerrico (RCaseG)* reused assets from *Domain Case Villa Regina* using CoVaMaT, so identifying a variety suitable for the requirements of the new area helps take advantage of the case with an already documented variety (DCaseVR).
- *Reusable Case Cinco Saltos (RCaseCS)* is similar to the case of Guerrico (RCaseG), but Villa Regina and Guerrico were already stored in CoVaMaT.

Table 4. Time required in days for each case category defined.

Activity	Services/Tasks	TCaseVR	DCaseVR	RCaseG	RCaseCS
Domain problem collection	S0. Defining the domain-case problem	30	90	10	10
Collection	S1. Collecting data from sensor	5	10	5	5
Preparation	S2. Deleting useless variables	10	15	2	2
	S2. Deleting useless variables	10	15	2	2
	S3. Imputation of null values	30	35	2	2
	S4. Adding new variable (FrostCont)	15	20	2	2
	S5. Balancing data for late frosts	3	8	2	2
	S6. Dealing with oultier values	30	35	2	2
	S7. Creating two datasets: original and dynamic	30	35	10	10
Analytics	S8. Computing a correlation analysis on the original dataset	15	20	5	5
	S9. Computing a deeper analysis of the rain and wind variables in the original dataset	25	30	5	5
	S10. Computing a correlation analysis on the dynamic dataset	10	15	2	2
	S11. Computing a frost analysis of the rain and wind variables in the dynamic dataset	25	30	5	5
	S15. Computing percent change analysis	—	—	36	15
Visualization	S12. Visualizing results from H1	10	15	5	5
	S16. Visualizing results for H3	—	—	10	5
	S13. Visualizing results from H2	10	15	5	5
Access	S14. Making results available	30	35	10	10
TOTAL in days		278	408	118	92

For instance, during *S0, defining the domain-case problem*, we can see that the definition of the problem takes 30 days for TCaseVR when we conduct the first communication with the experts, determine their needs, identify data, and form the hypotheses. Then, in DCaseVR, the number of days increases to 90 because it includes not only the activities mentioned in the traditional case but also the search for related work in the literature, the definition of the domain, the structuring of varieties, etc. Finally, the reusable cases RCaseG and RCaseCS took 10 days, including the adaptation of each case, and we firstly verified whether the hypotheses were exactly the same, and then we continued with the analysis of reusability and subsequent reuse. We should remark here that the calculation of the time involved some informal separation of tasks between those related to traditional big data processes and our reuse-based development. For instance, structuring variety was taken into account only during the applications of CoVaMaT processes, without interfering during the traditional processing.

In another example, during *S8, computing a correlation analysis on the original dataset*, we can see that, in TCaseVR, it took us 15 days since the Pearson and Spearman correlation analysis was performed based only on the original dataset; however, in DCaseVR, it took us 20 days because we had to not only compute the correlation but also document the results as assets of CoVaMaT. In the reusable cases RCaseG and RCaseCS, it took only 5 days because it involved only remaking the correlations with the recovered assets for the sets of the other reusable cases.

In addition, in the table, we can see that services *S15, computing percentage change analysis* and *S16, visualizing results for H3,* were only developed for the RCaseG and RCaseCS cases since those services refer to reusable cases only. We can also see that the times for Guerrico for these services were greater than for Cinco Saltos because the latter used the previously created assets (of RCaseG).

Figure 20 shows a simplified view of Gantt charts (the Gantt charts are specified in months due to space limitations), in which we can graphically observe the time for the four categories of cases. The first case, which represents the traditional big data process (TCaseVR), took a little over nine months to be developed. However, the domain case Villa Regina (DCaseVR), developed by applying our methodology, took almost thirteen months, that is, four months more. This was expected because, in this last case, we had to create the agrometeorology-domain assets and the domain-case assets. But the real advantages appeared during the next developments of the Guerrico and Cinco Saltos cases (RCaseG and RCaseCS), in which time was reduced abruptly.

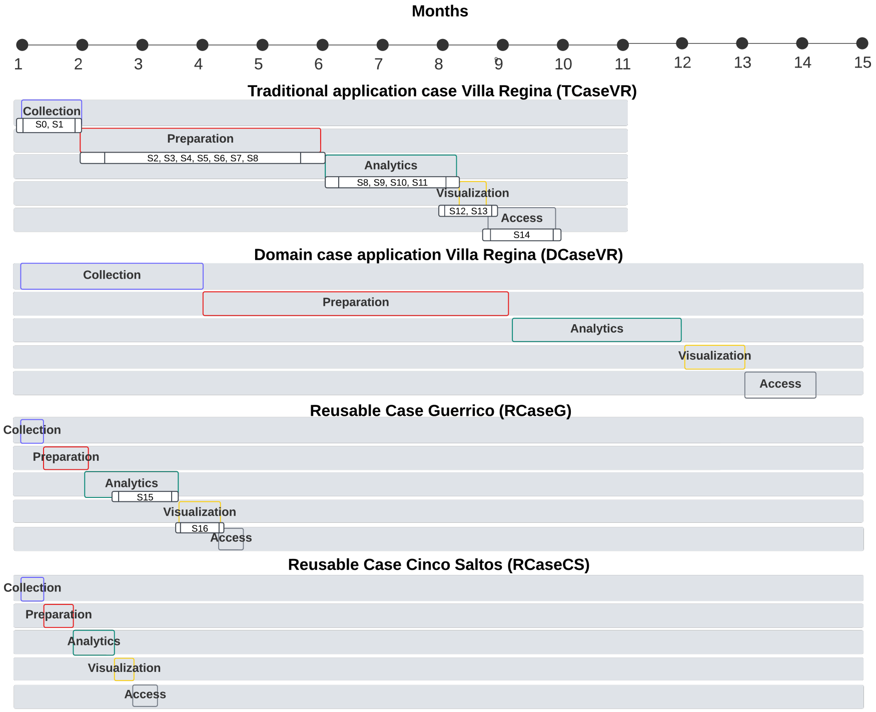


Figure 20. Simplified view of Gantt charts for the four cases evaluated.

In the same sense, in Figure 21, the bar chart shows the differences in the percentages of the time required for TCaseVR compared to the other three cases. Clearly, the chart shows that DCaseVR took 46% more time than the traditional case. However, for the two reusable cases, RCaseG and RCaseCS, the difference was negative, indicating a time reduction of 55% and 65%, respectively. Therefore, for these two cases, the savings are really interesting.

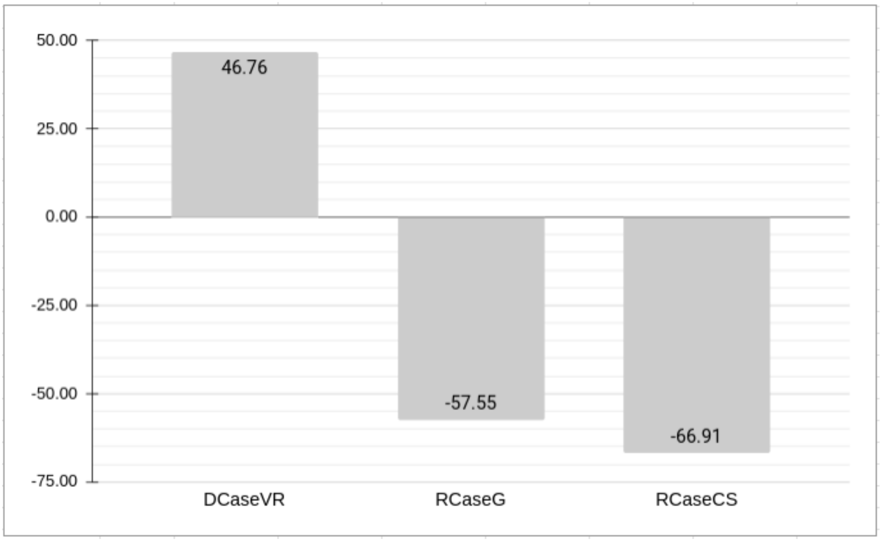


Figure 21. Differences in the percentages of time required for TCaseVR compared to the other three cases.

Threats to Validity

When faced with the results of a study, we have to ask ourselves whether they are correct, without errors or biases, whether they can be attributed to chance, and whether they are applicable to other contexts. In general, two types of study validity are defined: internal validity and external validity. *Internal validity* assesses the degree to which the results obtained in a particular study are correct. It determines whether the observed results can be attributed to the new proposal evaluated. There are two main errors that could threaten the internal validity of a study. These are biases (also called systematic errors) and random errors. *External validity* (or generalization) assesses the degree to which conclusions drawn from a study can be extrapolated or generalized beyond the cases studied. It depends on the size and characteristics of the cases and the context in which it is to be applied.

Systematic biases can lead to an overestimation or underestimation of the true effect of a proposal under study. In our evaluation, the way we performed measurements might have introduced a calculation bias when determining the duration of an activity. Then, in our case, a major concern involves the calculation of times associated with the activities of the big data process (Table 4). Time was defined for the four cases, and it can include variations, depending on the domains or contexts. Furthermore, the definition of the seventeen activities has been specially formulated only for the developed cases, and it could not be applicable in a similar way to other domains. Thus, our validation has a strong dependence on the tasks/times defined.

Other errors might come from collecting information differently, depending on the stakeholder's role (i.e., data scientist, software engineer, or domain expert), including their profiles, expertise, and domain knowledge. In our case, information was collected and processed by following an agreed-upon conceptualization of terms; however, we are aware that a more formal procedure is needed to encourage generalization to inter- and intra-domain applications, in which a multiplicity of actors can bring also multiple interpretations. Therefore, in part, these biases can be avoided with good protocols that comprehensively define what information should be recorded and how, including all activities of the big data process, and applying them exactly the same for all participants.

A different but related thread involves the selection of participants during running/evaluating the cases. Obviously, participants' knowledge can introduce biases due to previous experience. A more extensive evaluation would require further experimentation involving the application of reuse cases by people other than those who created the original domain-variety models.

In order to improve the generalizability and reliability of the research findings, future studies should also include a validation of the reusability in CoVaMaT. Its knowledge base should provide a set of standard services for each activity (probably related to particular domains), allowing validation to be comparable within and across domains.

5. Discussion

From our experiences in designing the methodology and its application on domain and reusable cases, we can draw the following lessons learned:

- *Quality is more than performance for big data systems*. Nowadays, most research in the field of BDS quality features is focused on performance, reliability, and other properties evaluated at the execution time. However, BDSs are software systems that must be designed and evaluated by considering the other properties as well, including modularity, reusability, and so on. Traditionally, developing software systems as SPLs has brought the benefits of reusability to reality through domain and application engineering. Therefore, treating BDSs as any other software system would afford similar benefits, at least potentially speaking. We have learned here that this is perhaps possible. We have adapted the traditional way of developing SPLs to a possible world for BDS: the world of domain assets and case assets. By conceptually separating what is relevant as domain features and instantiating them as cases, we might mirror a traditional SPL development and, consequently, focus BDS design on what is common and variable. This would provide a way of dealing with modeling reusable assets from different activities during BDS development, separating concerns and focusing on what a BDS needs to reuse during each stage of its lifecycle.
- *Variability turns into variety, and context is everywhere*. Variable and common features may be managed through variability management techniques, such as datasheets in OVM notation, now turning the focus into *variety*. We have redefined variety for BDSs, including source, content, and process diversity; however, when analyzing any system domain, usually, we register contextual features as constraints of the model, and BDSs are not an exception. Traditionally, the same system will behave differently, depending on its targeted organization, since usually, that system requires adaptation. For BDSs, *context variety*, as we defined, is transversal to every activity, and it constrains or drives the steps of the development process. From identification to reuse, context variety is immersed in our thoughts when selecting similar cases, discharging other ones, keeping some features, etc. So, we have learned that traditional variability also applies to BDSs, and it is a core concept that can be modeled in practice.
- *Many stakeholders but one holistic goal*: Our methodology should be transferred to industry, and to do so, understanding its rationale is essential. The development team groups people with different backgrounds and expertise, which make this team more valuable indeed. However, now, every participant must understand that he/she is working under a larger umbrella, which is called "variety identification". Every time he/she is doing the work, the mandatory question should be, "Is there anything here that we can use similarly in the future"? In this way, identifying common and variable domain assets becomes part of the stakeholder's daily work, not only a verification activity conducted every so often over a certain period of time. We have learned that changing the way of thinking and integrating a variety identification team takes a while; however, everything becomes easy once the team has "changed its mind".
- *Nothing is possible without support*. Successful reuse means classifying and storing assets for reuse, and this is impossible without a supporting tool. CoVaMaT was conceived of as such a tool, although its services are quite elementary yet. In spite of this, we are aware that an efficient and perhaps intelligent repository is the key to recovering the most appropriate domain assets. In our limited experience with

CoVaMaT, we have learned that the best tool is always desirable, but an in-progress prototype might be enough to start showing the benefits of reuse. And this is an important aspect of reaching stakeholder's support and commitment. In conclusion, supporting the variety identification and reuse project is a core element, and it cannot be reached without a supporting tool; however, not having the best one is not an excuse for not trying.

- *Reuse should not be a silver bullet*. We all know that reuse offers many benefits, but we should be certain that reuse is effective, too. To do so, we must obtain measurements that are beyond doubt. Identifying candidate assets for reuse is just a first step since the effort and time invested in reuse might hinder the entire process. For instance, we might identify a case asset, such as a data analysis process, which seems interesting for a new case; however, adaptation might require adding different data preparation, feature selection, etc. So, is this reuse really effective? We have learned that our methodology and tool need an extension that allows for measuring how effective the reuse is. And this fact sets the basis for one line of future work, as we summarize in the next section.
- *Thinking of reuse may help in more than one way.* Finally, with respect to the case studies, the new findings mostly depended on our context-based perspective of the frost analyses, which helped in (1) improving the findings themselves by adding new possible influencing factors and (2) setting a base for future reuse by structuring intermediate and final assets as domain-application cases.

6. Conclusions

Reusability is one of the most valuable software qualities for software systems due to its benefits, including increased efficiency, cost reduction, a faster time to market, etc. However, unlike other development paradigms, such as service-oriented or component-based options, of which reuse is a key aspect, in BDSs, it has not been addressed yet. Reuse mechanisms are only oriented towards the use of existing methodologies, aiming to make them adaptable to this kind of system. However, the development of BDSs is not trivial, and it involves several particularities that are not present in other paradigms. Thus, we propose a new methodology that takes advantage of domain-oriented mechanisms to build domain assets for reuse. In our methodology, each domain activity of the big data process builds these assets, which can be reused when developing BDSs (or case applications). Then, the separation into domain and application phases provides support for development *for and with reuse*. We have described the top-down perspective on variety identification through a case study of frost analysis. The study revealed interesting findings for the domain, as well as a proof of concept for our proposal.

Clearly, a two-phase development is not enough to ensure reusability. The new methodology should provide mechanisms to support the construction of reusable software artifacts. In this line, we proposed the concept of context variety as a key element to promote flexibility. Commonalities and their variabilities are defined iteratively during the two phases and instantiated in application development, allowing the creation of one or more systems.

We have also implemented a supporting tool, called CoVaMaT. Its prototype allows developers to document and retrieve assets to be used during the development of specific case applications. However, the current version of the tool needs further development. Firstly, code-level assets should be stored in repositories managed by the tool so that all the different assets are reached uniformly. Secondly, a more sophisticated search algorithm would be nice to improve similarity checking, which is currently just simple checking. And finally, more services might be added to make a domain-oriented search possible (perhaps by standardizing the domain vocabulary through the use of domain standards).

On the other hand, as we mentioned in the previous section, we are working on evaluating successful reuse during the application of our methodology. Thus, we are elaborating a *reuse analysis model*, adapted from three previous works [28,29], in which we evaluated reuse during the development of software product lines. We will evaluate assets that can be completely reused (as they were defined), partially reused (with adaptations), and/or used to detect the need to add new ones. The main idea of the model is to create *reuse scenarios* [30] based on the set of predefined operations associated with CoVaMaT. This reuse analysis model is still in a preliminary stage.

Author Contributions: Conceptualization, A.B. and A.C.; investigation, A.B. and A.C.; methodology, A.B. and A.C.; software, F.S.; writing—original draft preparation, A.B.; writing—review and editing, A.C.; supervision, A.B.; data provision and resources, A.M. (Ayelén Montenegro), A.M. (Angel Muñoz), and A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: CoVaMaT is available for download from https://github.com/IngSisFAI/CoVaMaT (accessed on 16 October 2024). Datasets and Python codes for domain cases developments are available at https://github.com/IngSisFAI/AgrometeorologyDomainCases (accessed on 16 October 2024).

Acknowledgments: This work was partially supported by UNComa project 04/F019, “Variety modeling in Big Data Systems” 2022–2025.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BDSs	Big data systems
SPL	Software product line
OVM	Orthogonal variability model
CoVaMaT	Context-based variety management tool
T-VIP	Top-down approach to the variety identification process
B-VIP	Bottom-up approach to the variety identification process

References

1. Carmel, E.; Agarwal, R. Tactical approaches for alleviating distance in global software development. *IEEE Softw.* **2001**, *18*, 22–29. [CrossRef]
2. Clements, P.C.; Northrop, L. *Software Product Lines: Practices and Patterns*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2001.
3. Pohl, K.; Böckle, G.; Linden, F.J.v.d. *Software Product Line Engineering: Foundations, Principles and Techniques*; Springer: Secaucus, NJ, USA, 2005.
4. van der Linden, F.; Schmid, K.; Rommes, E. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*; Springer: Secaucus, NJ, USA, 2007.
5. Käkölä, T. ISO Initiatives on Software Product Line Engineering: Vision and Current Status—Invited Talk for Variability. In Proceedings of the ER Workshops, Brussels, Belgium, 31 October–3 November 2011 ; p. 119.
6. Pasquetto, I.; Randles, B.; Borgman, C. On the Reuse of Scientific Data. *Data Sci. J.* **2017**, *16*, 8. [CrossRef]
7. Custers, B.; Uršič, H. Big data and data reuse: A taxonomy of data reuse for balancing big data benefits and personal data protection. *Int. Data Priv. Law* **2016**, *6*, 4–15. [CrossRef]
8. Borrison, R.; Klöpper, B.; Chioua, M.; Dix, M.; Sprick, B. Reusable Big Data System for Industrial Data Mining—A Case Study on Anomaly Detection in Chemical Plants. In Proceedings of the Intelligent Data Engineering and Automated Learning—IDEAL 2018, Madrid, Spain, 21–23 November 2018; Springer International Publishing: Secaucus, NJ, USA, 2018; pp. 611–622.
9. Xie, Z.; Chen, Y.; Speer, J.; Walters, T.; Tarazaga, P.A.; Kasarda, M. Towards Use And Reuse Driven Big Data Management. In Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries, Knoxville, TN, USA, 21–25 June 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 65–74.
10. Klein, J. Reference Architectures for Big Data Systems, Carnegie Mellon University’s Software Engineering Institute Blog. 2017. Available online: http://insights.sei.cmu.edu/blog/reference-architectures-for-big-data-systems/ (accessed on 9 June 2021).
11. Mavi, H.; Tupper, G. *Agrometeorology: Principles and Applications of Climate Studies in Agriculture*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2004. [CrossRef]

12. Zhou, I.; Lipman, J.; Abolhasan, M.; Shariati, N. Minute-wise frost prediction: An approach of recurrent neural networks. *Array* **2022**, *14*, 100158. [CrossRef]
13. Gobbett, D.L.; Nidumolu, U.; Crimp, S. Modelling frost generates insights for managing risk of minimum temperature extremes. *Weather. Clim. Extrem.* **2020**, *27*, 100176 [CrossRef]
14. Diedrichs, A.L.; Bromberg, F.; Dujovne, D.; Brun-Laguna, K.; Watteyne, T. Prediction of Frost Events Using Machine Learning and IoT Sensing Devices. *IEEE Internet Things J.* **2018**, *5*, 4589–4597. [CrossRef]
15. Talsma, C.J.; Solander, K.C.; Mudunuru, M.K.; Crawford, B.; Powell, M.R. Frost prediction using machine learning and deep neural network models. *Front. Artif. Intell.* **2023**, *5*, 963781. [CrossRef] [PubMed]
16. Bosch, J.; Olsson, H.H.; Brinne, B.; Crnkovic, I. AI Engineering: Realizing the Potential of AI. *IEEE Softw.* **2022**, *39*, 23–27. [CrossRef]
17. Osycka, L.; Cechich, A.; Buccella, A.; Montenegro, A.; Muñoz, A. CoVaMaT: Functionality for Variety Reuse through a Supporting Tool. In Proceedings of the XI Conference on Cloud Computing, Big Data & Emerging Topics (JCC-BD&ET), La Plata, Argentina, 27–29 June 2023.
18. Buccella, A.; Cechich, A.; Villegas, C.; Montenegro, A.; Muñoz, A.; Rodriguez, A. A Model of Reusable Assets in AIE Software Systems. *J. Comput. Sci. Technol.* **2023**, *23*, e13. [CrossRef]
19. Zhuoheng Chen, Stephen E. Grasby, K.G.O. Relation between climate variability and groundwater levels in the upper carbonate aquifer, southern Manitoba, Canada. *Hydrology* **2003**, *290*, 43–62. [CrossRef]
20. Delle Rose, M.; Martano, P. Datasets of Groundwater Level and Surface Water Budget in a Central Mediterranean Site (21 June 2017–1 October 2022). *Data* **2023**, *8*, 38. [CrossRef]
21. Verdes, P.F.; Granitto, P.M.; Navone, H.D.; Ceccatto, H.A. Frost prediction with machine learning techniques. In Proceedings of the VI Congreso Argentino de Ciencias de la Computacion, Buenos Aires, Argentina, 4–8 October 2000.
22. Liya Ding, K.N.; Shibuya, K. Frost Forecast using Machine Learning—From association to causality. *Procedia Comput. Sci.* **2019**, *159*, 1001–1010. [CrossRef]
23. Ding, L.; Noborio, K.; Shibuya, K. Modelling and learning cause-effect—Application in frost forecast. *Procedia Comput. Sci.* **2020**, *176*, 2264–2273. [CrossRef]
24. Kotikot, S.M.; Flores, A.; Griffin, R.E.; Nyaga, J.; Case, J.L.; Mugo, R.; Sedah, A.; Adams, E.; Limaye, A.; Irwin, D.E. Statistical characterization of frost zones: Case of tea freeze damage in the Kenyan highlands. *Int. J. Appl. Earth Obs. Geoinf.* **2020**, *84*, 101971. [CrossRef]
25. El-Hashash, E.F.; Shiekh, R.H.A. A Comparison of the Pearson, Spearman Rank and Kendall Tau Correlation Coefficients Using Quantitative Variables. *Asian J. Probab. Stat.* **2022**, *20*, 36–48. [CrossRef]
26. Anderson, M. Calculating and Interpreting Percentage Changes for Economic Analysis. *Appl. Econ. Teach. Resour. (Aetr) Agric. Appl. Econ. Assoc.* **2019**, *1*, 25–31. [CrossRef]
27. Niraula, R.; Meixner, T.; Norman, L. Determining the importance of model calibration for forecasting absolute/relative changes in streamflow from LULC and climate changes. *J. Hydrol.* **2015**, *522*, 439–451. [CrossRef]
28. Buccella, A.; Cechich, A.; Porfiri, J.; Diniz Dos Santos, D. Taxonomy-Oriented Domain Analysis of GIS: A Case Study for Paleontological Software Systems. *ISPRS Int. J. -Geo-Inf.* **2019**, *8*, 270. [CrossRef]
29. Buccella, A.; Pol'la, M.; Cechich, A. Improving Variabilty Analysis through Scenario-Based Incompatibility Detection. *Information* **2022**, *13*, 149. [CrossRef]
30. Tomer, A.; Goldin, L.; Kuflik, T.; Kimchi, E.; Schach, S. Evaluating software reuse alternatives: A model and its application to an industrial case study. *IEEE Trans. Softw. Eng.* **2004**, *30*, 601–612. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.