UNIVERSIDADE FEDERAL DE JUIZ DE FORA

INSTITUTO DE CIÊNCIAS EXATAS

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Jonas Silva Gomes

An IoT Architecture for Decision Support System in Precision Livestock

Juiz de Fora

2023

Jonas Silva Gomes

An IoT Architecture for Decision Support System in Precision Livestock

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. José Maria Nazar David

Coorientadora: Prof. Dr. Regina Maria Maciel Braga

Juiz de Fora

2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Silva Gomes, Jonas.
An IoT Architecture for Decision Support System in Precision Livestock / Jonas Silva Gomes. – 2023.
99 f. : il.

Orientador: José Maria Nazar David
Coorientadora: Regina Maria Maciel Braga
Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, INSTITUTO DE CIÊNCIAS EXATAS. Programa de Pós-Graduação em Ciência da Computação, 2023.

1. Decision-making. 2. Internet-of-Things. 3. Architecture. I. David, José Maria Nazar, orient. II. Prof. Dr.

Jonas Silva Gomes

An IoT Architecture for Decision Support System in Precision Livestock

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Ciência da Computação.

Aprovada em 26 de abril de 2023.

BANCA EXAMINADORA

Prof. Dr. José Maria Nazar David - Orientador
Universidade Federal de Juiz de Fora

Profª. Dra. Regina Maria Maciel Braga Villela - Coorientadora
Universidade Federal de Juiz de Fora

Prof. Dr. Victor Ströele de Andrade Menezes
Universidade Federal de Juiz de Fora

Prof. Dr. Wagner Antonio Arbex
Universidade Federal de Juiz de Fora

Prof. Dr. Valdemar Vicente Graciano Neto

Universidade Federal de Goiás

Juiz de Fora, 11/04/2023.

Documento assinado eletronicamente por **Jose Maria Nazar David**, **Professor(a)**, em 26/04/2023, às 11:32, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Jonas Silva Gomes**, **Usuário Externo**, em 26/04/2023, às 12:39, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Valdemar Vicente Graciano Neto**, **Usuário Externo**, em 26/04/2023, às 12:59, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Regina Maria Maciel Braga Villela**, **Professor(a)**, em 26/04/2023, às 13:40, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Wagner Antonio Arbex**, **Professor(a)**, em 28/04/2023, às 17:36, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Victor Stroele de Andrade Menezes**, **Professor(a)**, em 02/05/2023, às 10:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1230005** e o código CRC **4AE25B1B**.

I respectfully dedicate this work to the cherished memory of my beloved grandmother, Luci Helena. Her memory will forever be cherished and her love and inspiration will continue to guide me throughout my life.

ACKNOWLEDGEMENTS

I would like to express my appreciation to my mother, father, and Luma for their unwavering love and support throughout my academic journey. Their encouragement has been invaluable in helping me to achieve my goals.

Additionally, I extend my gratitude to my supervisors, José Maria and Regina Braga, as well as all the other esteemed professors in the department. Their guidance and mentorship have played an integral role in paving my path into the field of computer science.

I would also like to express my deepest gratitude to Valdemar, Victor and Wagner for your invaluable contribution to my dissertation. Your careful reading, thoughtful analysis, and insightful feedback have helped me to refine and improve my work, and for that, I am truly thankful. I appreciate the encouragement and support you offered throughout the entire process. Your positive feedback and constructive criticism provided me with a sense of direction and purpose, and I am confident that your contributions will have a lasting impact on my future academic pursuits. Once again, I extend my sincerest thanks to each and every one of you for your invaluable contribution to my dissertation. It has been an honor to work with such a knowledgeable, skilled, and supportive group of individuals.

Sincere gratitude is due to EMBRAPA for making important data available for my research. Access to this valuable information enabled me to develop my study with the necessary quality and precision. I also want to thank the Programa Residência Zootécnica Digital (RZD) and all involved in organizing this project. The technical visit to the Compost Barn was crucial to the success of my work, and each person's dedication and commitment made it possible. I appreciate the time taken to show me the Compost Barn, provide precise and clear information, clarify techniques used, share data, and answer my questions with patience.

Furthermore, the opportunity to experience the reality of the Compost Barn firsthand, understand the challenges faced, the solutions found, and the prospects for the future is greatly appreciated. These experiences will enrich my research. I reiterate my thanks to all those involved in the technical visit for making this unique opportunity possible and for demonstrating exemplary commitment to agricultural research and development. Opportunities like this are fundamental to the training and improvement of professionals committed to developing innovative solutions for agriculture. I thank Wneiton, Leonardo, Carlos, Wagner, and all researchers involved for your collaboration and support.

I would like to express gratitude to the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001 - for their financial support during my research. Their funding was invaluable in enabling me to carry out my studies and complete this dissertation. I am grateful for the opportunities and resources provided by CAPES, which allowed me to enhance my knowledge and skills in the field of computer science. Once again, I extend my heartfelt appreciation to CAPES for their invaluable support.

"Any sufficiently advanced technology is indistinguishable from magic."Arthur Clarke

RESUMO

Na indústria pecuária, a produção animal sustentável é o principal objetivo do desenvolvimento tecnológico. Porém, é fundamental manter boas condições no ambiente devido à suscetibilidade dos animais a variáveis como temperatura e umidade, que podem causar doenças, perdas de produção e desconforto. Assim, os sistemas de produção pecuária requerem monitoramento, controle e mitigação das condições indesejadas através de ações automatizadas. A principal contribuição deste estudo é a introdução de uma arquitetura auto-adaptativa denominada e-Livestock para apoiar as decisões relacionadas à produção animal. Foram conduzidos dois estudos de caso, envolvendo a arquitetura e-Livestock, que foi utilizada no sistema de produção Compost Barn - ambiente e tecnologia onde ocorre a produção de gado leiteiro. Os resultados demonstraram a utilidade do e-Livestock para avaliar três aspectos principais: (i) abstração de tecnologias disruptivas baseadas em Internet das Coisas (IoT) e Inteligência Artificial, e sua incorporação em uma arquitetura única, específica para o domínio da pecuária, (ii) suporte para a reutilização e derivação de uma arquitetura auto-adaptativa para apoiar o desenvolvimento de uma aplicação de apoio à decisão para o subdomínio da pecuária e (iii) suporte para estudos empíricos em uma fazenda inteligente real para facilitar a transferência de tecnologia para a indústria. Portanto, a principal contribuição dessa pesquisa é o desenvolvimento de uma arquitetura combinando técnicas de machine learning e ontologia para apoiar decisões mais complexas ao considerar um grande volume de dados gerados nas fazendas. Os resultados revelaram que a arquitetura e-Livestock pode apoiar monitoramento, controle, previsão e ações automatizadas em um ambiente de produção de leite/Compost Barn.

Palavras-chave: Sistema de Apoio à Decisão. Internet das Coisas. Arquitetura e-Livestock. Arquitetura auto-adaptativa.

ABSTRACT

Sustainable animal production is a primary goal of technological development in the livestock industry. However, it is crucial to master the livestock environment due to the susceptibility of animals to variables such as temperature and humidity, which can cause illness, production losses, and discomfort. Thus, livestock production systems require monitoring, reasoning, and mitigating unwanted conditions with automated actions. The principal contribution of this study is the introduction of a self-adaptive architecture named e-Livestock to handle animal production decisions. Two case studies were conducted involving a system derived from the e-Livestock architecture, encompassing a Compost Barn production system - an environment and technology where bovine milk production occurs. The outcomes demonstrate the effectiveness of e-Livestock in three key aspects: (i) abstraction of disruptive technologies based on the Internet of Things (IoT) and Artificial Intelligence and their incorporation into a single architecture specific to the livestock domain, (ii) support for the reuse and derivation of an adaptive self-architecture to support the engineering of a decision support system for the livestock subdomain, and (iii) support for empirical studies in a real smart farm to facilitate future technology transfer to the industry. Therefore, our research's main contribution is developing an architecture combining machine learning techniques and ontology to support more complex decisions when considering a large volume of data generated on farms. The results revealed that the e-Livestock architecture could support monitoring, reasoning, forecasting, and automated actions in a milk production/Compost Barn environment.

Keywords: Decision Support System. Internet-of-Things. e-Livestock Architecture. Self-adaptive Architecture.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
INMET	Instituto Nacional de Meteorologia
DSS	Decision Support System
ML	Machine Learning
CS	Case Study
IOT	Internet of Things
AI	Artificial Intelligence
IS	Intelligent Systems
OWL	Web Ontology Language
KR	Knowledge Representation
OBDA	Ontology-Based Data Access
IDSS	Intelligent Decision Support Systems
RQ	Research Question
PICOC	Population, Intervention, Comparison, Outcome and Context
DSR	Design Science Research
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Square Error
R2	R Squared

SUMÁRIO

1 INTRODUCTION

This chapter presents the motivation that inspired this study, the main problems and the research methodologies employed, leading to the development of the solution proposed.

1.1 CONTEXTUALIZATION

In recent years, the Internet of Things (IoT) has started to connect several devices and sensors, which has generated opportunities in several sectors, including agriculture and livestock (Zhai et al., 2020). Wearable IoT devices, for example, can be used to track the activities of humans and animals. In this sense, IoT in animal health uses biosensors and software to monitor and maintain animal health records. The large volume of data generated by these devices can be interpreted by intelligent systems. Subsequently, these data can support producers and managers in decision-making (Symeonaki et al., 2020).

The use of sensors to monitor crops and soil quality has also been extended to monitor animal health. In the livestock context, to ensure animal welfare, it is necessary to monitor, control, and intervene in the environment and make real-time decisions that, hopefully, can positively impact the animal's health. However, the continuous monitoring of animals through sensors has transformed farms and brought new challenges beyond sensors and IoT devices (Farooq et al., 2019).

Agribusiness is a critical domain where reckless actions or negligent monitoring can lead to animal deaths, crop failures, financial loss, and even national economic unbalance. In livestock which involves raising animals for human consumption, inappropriate levels of certain variables, such as humidity and/or temperature in the production environment can, for instance, lead to the development of diseases and inflammation in the glands of animals that may cause a reduction in production or even total loss (Graciano Neto et al., 2022). To mitigate these risks, agribusiness Production Systems (PS) should use software-based solutions alongside manual labor to (i) monitor the environment, (ii) self-regulate its behavior to act on the environment, preserving the ideal conditions to leverage productivity, (iii) automate actions and (iv) predict not favorable conditions that could harm production, enabling the system to change the environmental conditions before those conditions occur. Difficulties such as enriching decisions, developing systems capable of adapting to uncertainties and dynamic factors, and making production predictions still need to be explored. With these demands, PS have evolved into smart farms with numerous sensors and actuators that generate vast amounts of data, requiring processing and reasoning to prevent negative outcomes such as animal diseases. These sensors and actuators generate massive data, which demands processing and reasoning to avoid unfortunate situations. All these requirements and technologies have made PS environments to become highly dynamic and complex. It is not unusual that new producers or technologies must be joined to the PS structure. Then, supporting PS demand an abstraction that satisfies the requirements posed by the livestock context.

1.2 MOTIVATION

The integration of IoT data management in precision livestock farming is becoming increasingly complex with the emergence of Agriculture 4.0 and the widespread use of smart applications. This work discusses the role of data integration in precision livestock farming and how it is impacted by dynamic and adverse factors, such as changing climate patterns. The work also highlights the complexities of monitoring animal health and environmental conditions in confined animal systems, such as Compost Barns.

Precision livestock farming involves the collection of data from different contexts, which is used for decision-making purposes. For example, in a confined animal system such as a Compost Barn, regular adjustments of the internal temperature are necessary to ensure maximum comfort for the animals. By integrating data collected from sensors with weather station data, the internal temperature of the environment can be controlled appropriately, reducing the impact of sudden temperature changes on milk production. Furthermore, integrating data with geolocation services can help maintain accurate monitoring of animals on pasture. Thus, data management and integration play a crucial role in precision livestock farming.

However, the complexity of precision livestock farming is compounded by the movement of animals in space and exposure to adverse situations. Compost Barns require constant monitoring of environmental conditions and animal health, as animals are confined in a covered shed with a freely accessible communal bedding area. The bedding area is composed of sawdust or wood shavings, where most of the waste is retained, without partitions. North American producers introduced this production system, which has been adopted in Brazil since 2001. The shed has a ventilation system designed to remove heat produced by the composting of bedding and animal waste (MilkPoint, 2016).

The constant generation of data by sensors increases the complexity of data management, as sensors are subject to mechanical failures and environmental interference. Therefore, decision-making processes need to consider multiple dynamic factors that can impact animal production economically and environmentally. Furthermore, the integration and management of data in precision livestock farming play a crucial role in optimizing animal production and maintaining animal health. However, the complexity of confined animal systems such as Compost Barns and dynamic and adverse factors, make data management and decision-making processes more challenging. Therefore, it is essential to consider multiple factors and develop robust data management strategies that consider the specific needs of different livestock systems.

Regarding the livestock of dairy cattle, some diseases stand out due to the economic impact they generate. Mastitis is a disease that typically involves bacterial infection and leads to inflammation in the udder of a cow, which in turn results in elevated somatic cell count in the milk. One of the primary reasons for milk disposal is mastitis, which occurs when animals are undergoing treatment. This is due to the medication's impact on the milk, making it unsuitable for consumption. The best way to avoid the impacts of mastitis is through disease control and prevention. Control can be done through early treatment, separation of sick animals to prevent contamination of other animals, and sometimes even the disposal of animals with chronic diseases. Prevention can be achieved by maintaining environmental conditions, as a well-controlled humidity environment helps to avoid the proliferation of environmental bacteria, reducing the risks of contamination. Another important factor is animal hygiene, cleaning the animal before and after milking.

Traditional architectures need to prepare to deal with such a complex domain, whose context can change rapidly. New animals can arrive at the farm, climate changes can vary throughout the year, and systems must be able to continue functioning correctly, supporting producers on their farms. To solve this problem, an architecture is needed to construct a Decision Support System (DSS) for smart farms. Farmers can suffer significant losses due to the complexity of decision-making when it comes to mastitis control and prevention. Failure to effectively manage mastitis can result in reduced production and even the loss of animals.

Recent secondary studies have been conducted to answer relevant questions about decision support systems in the precision livestock domain. For example, Villa-Henriksen et al. (2020) present a review of interoperability standards. This secondary study discusses the challenges of integrating open data with the various data generated on farms. In Bahlo et al. (2019), the authors review network, physical device, and application problems and highlight the role of middleware in data integration. In Zhai et al. (2020), the authors raise issues related to decision-making and the need for additional elements, such as weather station data, due to the influence of temperature and humidity on animal production.

In these studies, the authors cover the importance of integrating data, whether with open databases, weather station services, geolocation services, or even between farms. Although data integration is discussed for decision-making, the authors need to bring a general perspective on how to use the data collected on farms, and what software models and architectures are used in decision support systems.

Other researchers have been working to discover how meteorological data, linked to intelligent prediction models, could be used in agriculture (Newlands et al., 2019). Some works are interested in how Artificial Intelligence (AI) systems support the decision-making process (Gualdi and Cordella, 2021) and how AI improves agricultural productivity (Lakshmi et al., 2020).

To address the issues faced in agribusiness, integrating self-adaptive abilities can enhance smart farm PS using IoT software systems and software engineering methodologies. This creates a comprehensive vision that includes smart services, smart products (IoT), Agribusiness 4.0 systems, and self-adaptive software engineering (Weyns, 2020). An architecture for decision-making, combined with self-adaptation techniques, meets these requirements by accommodating the necessary software-based modules, including (i) an Internet of Things (IoT) module for sensing and acting on PS environment, (ii) self-adaptive characteristics to modulate the environment based on stimuli, (iii) Artificial Intelligence (AI) techniques to reason about the large amounts of data collected to support automated actions, and (iv) a complex, flexible and reusable architecture open to receiving new contributions to the PS. Hence, aiming to tackle the challenge of providing an architecture that meets the needs of a smart farm, encompassing processing, integration, and intelligence, we developed an architecture for DSS, called e-Livestock.

Therefore, the research problem addressed in this work is ***to support automated monitoring, reasoning, and automated actions in smart farms to enhance milk production***. The e-Livestock architecture was designed to gather new knowledge at runtime to resolve uncertainties, reason about itself, its context, and goals, and adapt based on actuators to achieve goals. We conducted a case study in a Compost Barn PS for dairy cattle to assess the proposed solution. The goal was ***to analyze*** *the support to monitor the environment, reason on data, and automate actions* **from the** *researcher's/farmers' point of view,* ***in the context of*** *a smart farm system.* The research question posed within the scope of this study is ***"How can e-Livestock support automated monitoring, reasoning and actions in smart farms?"***

1.3 OBJECTIVES

This work aims to support producers in decision-making, through an approach that uses intelligence to enrich farm information. This architecture, called e-Livestock, aims to help rural producers of dairy cattle to understand the situation of animals on the farm, and thus, favor decision-making through predictions and inferences.

Using Machine Learning (ML) techniques, producers can have a view with predictions about milk production, while ontologies provide a retrospective view of the production data. By combining these approaches into one system, presenting both views (predictive and retrospective), with the support of graphical visualization, producers can make more enriched decisions.

Therefore, the main contribution of our research is the development of an architecture combining ML techniques and ontology to support more complex decisions when considering a large volume of data generated on farms. To support our approach, an evaluation was carried out in a Compost Barn, a production system related to improving the production of confined dairy cattle. This Compost Barn is located at Coronel Pacheco, Minas Gerais – EMBRAPA, a Brazilian Agricultutral Research Corporation.

The results show that the solution supports farms in animal management and well-being in the decision-making process. In addition, it was possible to infer past data using an ontology model, providing agribusiness managers with insights derived from past data.

To achieve these objectives, the following specific objectives were considered:

- Specify a layered architectural model capable of being maintainable, taking into account various types of data coming from sensors;
- Develop an architecture capable of supporting decisions in smart farms, through intelligence with predictions and inferences;
- Specify and implement a knowledge base through an axiomatically rich ontological model, capable of extracting implicit knowledge about dairy animals;
- Develop a layer responsible for managing the intelligent models and capable of storing them (ontology and machine learning;
- Implement the proposed architecture with real-world historical data from the farm.

1.4 OUTLINE

This work is divided into six chapters. Chapter 2 presents the concepts involved in this work. Chapter 3 presents the related works and the systematic mapping of the literature. Chapter 4 presents the methodology used, called Design Science Research, the proposed solution to support decision-making in precision livestock, detailing the conceptual aspects and the implementation of the solution. Chapter 5 presents the evaluation of the solution, highlighting its planning, execution, and results obtained. Chapter 6 presents the final remarks, highlighting the contributions of the work, its limitations, and future work.

2 THEORETICAL FOUNDATION

This chapter introduces the main concepts related to the research area, i.e., decision support system, Internet of Things (IoT), ontology, and Intelligent Systems (IS).

2.1 INTERNET OF THINGS

The concept of the Internet of Things (IoT) is related to the connection of a network of "objects"through the Internet without direct human intervention (Yang et al., 2013). Through geographically distributed sensors, IoT derives real-time information, comprising many applications, among which the most notable are smart houses, personal healthcare, intelligent logistics, Industry 4.0, and real-time conditions monitoring. In the latter context, meteorological data are used as a source of information for climate warning systems. In this vein, farmers have been using it to make better tactical decisions to avoid harvest loss, mining companies employ it to monitor soil conditions, and military bases use it to have prior knowledge of abrupt changes in environmental parameters (Chavan and Momin, 2017).

Many applications can benefit from the periodically sensed data, which are collected from wireless sensors that constitute the smart environment. The collected data is often transmitted to a cloud platform, where many users can access it. IoT has been affecting the way data is produced and used, provoking the emergence of new software products and services due to the dynamic environment. The enormous amount of generated sensor data must be stored, processed, and presented transparently, efficiently, and easily understood.

The application of IoT in agriculture and livestock has advantages due to the possibility of monitoring and controlling many different parameters in an interoperable, scalable, and open context, with the increasing use of automated sensors (Villa-Henriksen et al., 2020). In agribusiness, one of the innovations is its combination with data processing, allowing greater support in decision-making. One of the objectives of using IoT in livestock is to increase accuracy in using information for decision-making, as we will discuss in the following chapters.

2.2 ONTOLOGY

To support the sharing and reuse of knowledge between different systems, it is necessary to define a common vocabulary for representing this knowledge. In this sense, Gruber (1995)borrowed the term ontology from philosophy and defined it for computation as a formal and explicit specification of a shared conceptualization. This conceptualization is a simplified and abstract view of the world that one wants to represent for some purpose. In general, an ontology specifies a domain vocabulary, composed of definitions of classes, relationships, and functions. Ontologies are used to share a common understanding about the structure of information between people or software agents; support domain knowledge reuse; explain assumptions about the domain; separate domain knowledge from operational knowledge and analyze domain knowledge (Gruber, 1995).

The Ontology-Based Data Access (OBDA) strategy uses ontology as the mediated schema, from where queries can be posed in the data integration system. In addition, data sources are described according to the classes and the schema mappings are specified following the properties that link concepts and instances thereof (individuals) in the ontology. An example of the application of ontologies in the management of complex data domains can be found in the omics study (genomics, proteomics, transcriptomics, metabolomics, etc). In such context, Knowledge Representation (KR) techniques and data integration methods are essential to process the highly heterogenous datasets to allow the extraction of comprehensive knowledge from all the fuzzy information necessary to understand the diverse variables - often presented under different data types - involved in the complex phenomenon of a disease, for instance (Louie et al. (2007); Zitnik et al. (2019)).

The Ontology Web Language (OWL) was designed to facilitate the interpretation of Web content using ontologies by providing additional vocabulary along with formal semantics, more complete than other languages such as XML, RDF, and RDF Schema (RDF-S) (McGuinness and Van Harmelen, 2004). The advantage of OWL is that it can be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL was developed as a language for building ontologies that provide high-level descriptions of Web content. These ontologies are created by building class hierarchies that describe concepts in a domain and relating classes to each other using properties (McGuinness and Van Harmelen, 2004).

OWL and Semantic Web Rule Language (SWRL) are the main languages of the Semantic Web. OWL can also represent data as instances of OWL classes – called individuals – and provides mechanisms for reasoning and manipulating the data. OWL also provides an axioms language to define how to interpret concepts in an ontology precisely (O'Connor et al., 2008). SWRL allows users to write rules that can be expressed in terms of OWL concepts and that can reason about OWL individuals. One of the most powerful features of SWRL is its ability to support built-ins (Horrocks et al., 2004). Built-ins are user-defined predicates that can be used in SWRL rules. Several core built-ins for common math and string operations are defined in the SWRL proposal. SWRL allows new libraries of built-ins to be defined and used in rules. Users can define built-in libraries to perform a wide variety of tasks. Such tasks might, for example, include currency conversion, temporal manipulations, and taxonomy searches. In general, the arguments to these inners must be OWL DL property values, that is, literals or individuals. However, some built-in libraries may also support class or property built-in arguments, although such built-ins should only be used in OWL Full ontologies (O'Connor et al., 2008).

2.3 DECISION SUPPORT SYSTEM

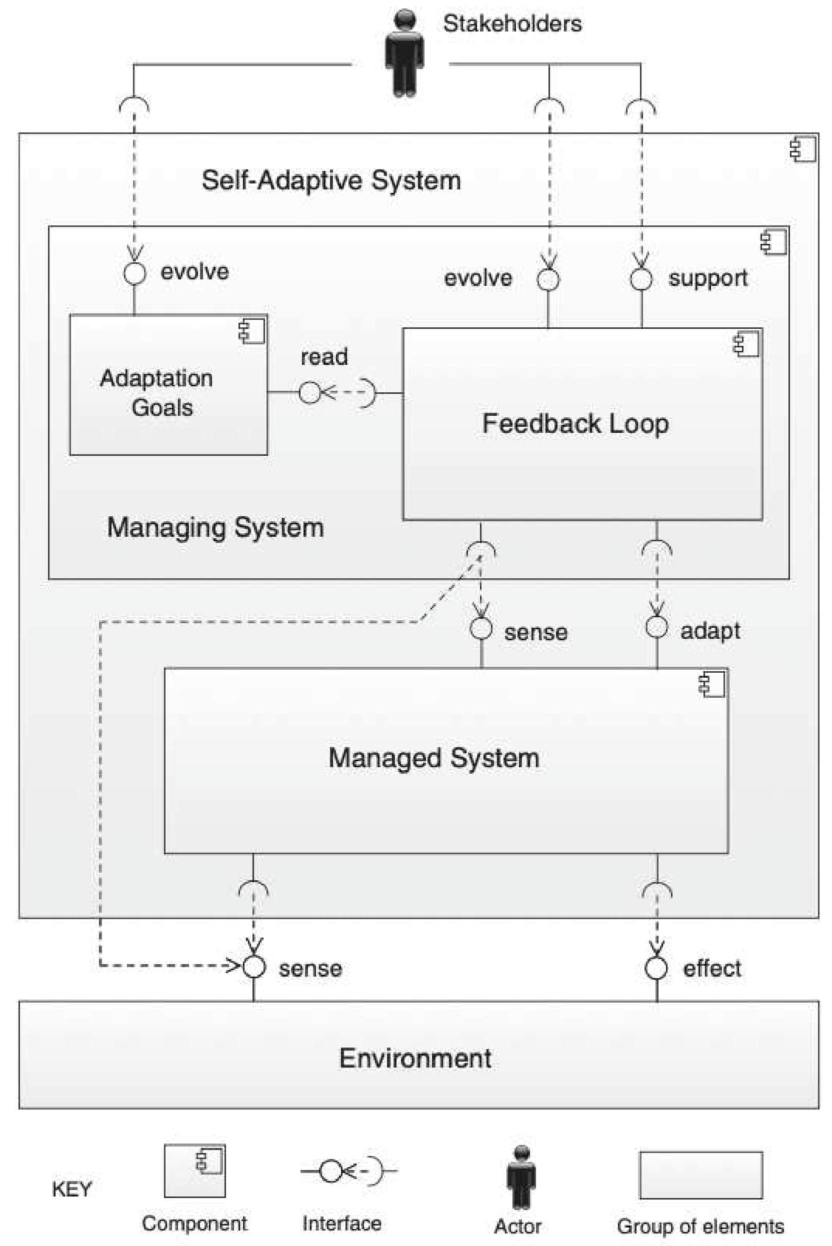
A Decision Support System is an application that provides support for different decision-making activities in a specific domain (Belciug and Gorunescu, 2020). DSSs are used in various domains such as medical diagnosis, engineering project evaluation, business management, agricultural production, livestock, credit verification, air travel industry, railway management, and forest management. A DSS can be a system that responds to a simple query or can model a complex human decision-making process (Belciug and Gorunescu, 2020).

In a more updated definition, DSS is expected to operate under uncertain conditions without interruption. Possible causes of uncertainties include changes in the operational environment, resource availability dynamics, and user objectives variations. Traditionally, it is the task of system operators to deal with such uncertainties. However, these management tasks can be complex, error-prone, and costly. In this vein, arises a new concept to deal with these uncertainties, called self-adaptation. Self-adaptation aims to enable the system to collect additional data on uncertainties during operation to manage itself based on high-level goals. As many IoT systems today need to be operational 24 hours a day, seven days a week, uncertainties must be resolved at runtime. Self-adaptation is about how a system can mitigate such uncertainties autonomously or with minimal human intervention (Weyns, 2020).

Self-adaptive systems are a relatively new style of decision support system that can adjust themselves in response to changing conditions. These systems use machine learning algorithms and other artificial intelligence techniques to monitor and analyze data from various sources, including sensors, databases, and other inputs. The self-adaptive system then adjusts its decision-making algorithms in real-time based on the data it has collected, allowing it to make more accurate and timely decisions. One of the main advantages of self-adaptive systems is their ability to respond to unexpected changes in their environment. For example, a self-adaptive system used in manufacturing might detect a problem with a machine and automatically adjust its operations to compensate. Similarly, a self-adaptive system used in transportation might detect a change in traffic patterns and adjust its route accordingly. These systems are also able to learn from past experiences and use that knowledge to improve their decision-making processes over time (Weyns, 2020).

A self-adaptive system comprises two distinct parts: the first part interacts with the environment and is responsible for domain concerns - i.e., the concerns of the users for whom the system is built; the second part consists of a feedback loop that interacts with the first part (and monitors its environment) and is responsible for adaptation concerns, i.e., about domain concerns (Weyns, 2020). Based on the two basic principles of self-adaptation, a conceptual model for self-adaptive systems can be defined, describing the basic elements of such systems and the relationship between them. The basic elements are intentionally kept abstract and general but align with the basic principles of self-adaptation. Figure 1 shows the conceptual model of a self-adaptive system (Weyns, 2020).

Figure 1 - Conceptual model of a self-adaptive system



Source: Weyns (2020).

The conceptual model of a self-adaptive system typically includes several key components as Figure 1 shows: Sensing and monitoring: This component is responsible for gathering information about the system and its environment, such as data on system performance, user behavior, and changes in the operating environment. Managing System:

This component processes the data gathered by the sensing and monitoring component and makes decisions about how to respond to changes in the environment. This may involve selecting from a range of possible actions, depending on the specific context and goals of the system. Managed System: This component is responsible for implementing the decisions made by the managing system, which may involve modifying the behavior or configuration of the system in response to changing conditions. Feedback Loop: This component is responsible for evaluating the effectiveness of the system's responses to changes in the environment and providing feedback to the managing system, which can then use this information to improve its future decisions.

2.4 INTELLIGENT SYSTEMS

Intelligent Systems (IS) represent an interdisciplinary research domain that brings together Artificial Intelligence (AI) and a variety of related domains, such as psychology, linguistics, and neurology, connected by many interdisciplinary relationships. Nowadays, a wide variety of ISs have been developed, such as expert systems; fuzzy systems; artificial neural networks; evolutionary computation (genetic/evolutionary algorithms, genetic programming, evolutionary strategies); support vector machines; particle swarm optimization; ant colony systems; memetic algorithms; ant colony optimization; clustering; Bayesian (learning) model; deep learning; and hybrid models (neuro-genetic, neuro-fuzzy, fuzzy-genetic, etc), among others (Belciug and Gorunescu, 2020).

Intelligent Decision Support Systems (IDSS) is a DSS that uses (AI) methods. Researchers are trying to develop computational tools that behave like a human advisor: an entity that can collect and analyze evidence, that can diagnose after identifying a problem, and finally suggest certain solutions (Belciug and Gorunescu, 2020). Adding machine learning into IDSS offers many flexible algorithms that are well suited for analyzing large, complex datasets. Therefore, the application of such algorithms for herd management analysis and performance data or computerized decision-making on commercial dairy farms looks very promising (Pietersma et al., 1998). Machine learning algorithms offer greater flexibility regarding problems of multicollinearity, missing values, or complex interactions between variables.

2.5 FINAL REMARKS OF THE CHAPTER

In conclusion, the theoretical foundation chapter has provided a comprehensive overview of decision support systems, intelligence and ontology. It has explored the history, and current state of decision support systems, including their key features, components, and types. Additionally, it has delved into the concept of intelligence, particularly artificial intelligence, and its integration into decision support systems. This theoretical foundation will serve as a valuable guide for researchers, practitioners, and decision-makers in the field of information systems, helping them to harness the power of decision support systems and intelligence to enhance organizational decision-making processes.

3 METHODOLOGY

The following chapter discusses a systematic mapping conducted to support our conjectures and related works.

3.1 SYSTEMATIC MAPPING

Through this systematic mapping, our focus is to list the techniques and approaches that are used to support data integration on farms so that Decision Support Systems can support decisions in the context of precision agriculture and livestock.

This systematic mapping aims to answer four research questions defined below: **(i) How to integrate Decision Support Systems data from IoT devices in precision agriculture and livestock?** Through this question, we seek to investigate the state of the art on the integration of data generated on farms by IoT devices to support decisions in precision agriculture and livestock; **(ii) RQ2: How semantic data is handled to integrate Decision Support Systems in precision agriculture and livestock?** With this question, we seek to identify, in the results of RQ1, which semantic integration techniques are being used to enrich data to take decisions in precision agriculture and livestock; **(iii) RQ3: What intelligent models are used to support Decision Support Systems in precision agriculture and livestock?** Given the increasing use of computational intelligence, pointed out in secondary works, this question aims to explore the models, techniques and intelligent algorithms that are being used in smart farming; **(iv) RQ4: How is computing technology applied in precision agriculture and livestock?** Considering that the Internet of Things has been widely used in Agriculture 4.0, as shown by secondary studies, the purpose of this question is to analyze how the solutions used contribute to the advancement of computing on farms, whether monitoring through sensors, in the use cell phone, drones or smart devices.

3.1.1 Planning

The methodology used in this work follows the guidelines proposed by Kitchenham (2004). During the mapping process, the support tool called Parsifal[1] was used.

According to Kitchenham (2004) recommendations, the research was divided into three parts, planning, conducting, and reporting. During planning, the need for revision, the research questions that should be answered, and the protocol to be followed were identified. After that, the conduction process began, through which we identified and selected the studies, and performed the backward and forward snowballing techniques, according to the hybrid search technique (Mourão et al., 2017). The hybrid search technique consists of executing the search string in a database that indexes other databases, such

[1] https://parsif.al/

as Scopus. Then perform backward and forward snowballing on the articles returned by the string. After that, extraction and synthesis of the data obtained were performed for quality analysis, and we present the results.

Search terms were defined considering Population, Intervention, Comparison, Outcome and Context (PICOC) (Kitchenham and Charters, 2007) to identify keywords, synonyms and build the search string based on the combination of key terms. Comparison is not relevant in this work since this mapping is an exploratory study. To select the articles to be analyzed to answer the research questions, some criteria were defined. Inclusion criteria aimed to add those works that present architectures, data integration and intelligent models to support decisions in precision agriculture and livestock, and studies in English. The exclusion criteria adopted were: (i) book chapters; (ii) studies published only as abstracts; (iii) studies whose version is older than another study already considered; (iv) articles not in English; (v) works that do not present an abstract; (vi) keywords are not present in the abstract. More details of the selection process can be seen in this repository[2]

The search was carried out in the Scopus electronic database, since it indexes several other bases, allowing the execution of only one search string (Mourão et al., 2017). Two primary studies (control articles), i.e., Villa-Henriksen et al. (2020) and Helfer et al. (2019) were defined. Budgen et al. (2008) suggest control articles used for the accuracy of the search string in databases of selected data and whether the search retrieve the right relevant studies. Keywords from these articles were analyzed to find new relevant terms to include as part of the search string. Experts in Decision Support Systems also participated in the verification of the search string. The final search string was described as follows:

("internet of things" OR iot) AND (dss OR "decision support system") AND (agriculture OR livestock).

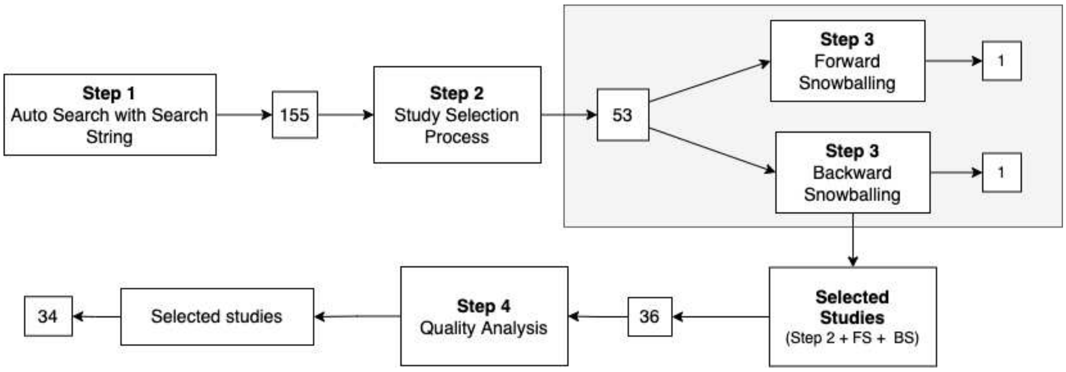
3.1.2 Conduction

The first step in this phase was to execute the search string, considering the selected sources. Subsequently, the study selection process was carried out, illustrated in Figure 2, which comprised 4 main stages.

In Step 1, search results from databases were merged into Parsifal, and duplicated papers were removed, totaling 155 articles. Step 2 involved the analysis of the title and abstract considering the inclusion/exclusion criteria. Those articles that did not address Decision Support Systems and IoT were excluded. The Introduction, Theoretical Reference and Conclusion sections of the articles were read. As a result, 102 articles (65.8%) were excluded and 53 (34.2%) were included.

[2] https://gist.github.com/jjthegomes/a907809e75c9583e975228db77e62866

Figure 2 - Data collection process



Source: Mourão et al. (2017)

According to the guidelines proposed in (Kitchenham, 2004), researchers should develop quality checklists to assess individual studies. During Step 3, backward and forward snowballing (Webster and Watson 2002) were applied to the 53 selected articles resulting in 1 more article for backward snowballing and 1 for forward snowballing.

In Step 4, the papers were read in full and their quality was assessed using the list of quality assessment questions, which was customized for the context of this mapping. At this stage we also perform data extraction, verifying which approach the solutions used to support Decision Support Systems and their application in precision agriculture or livestock. This selection was carried out by specialists in the areas of Decision Support Systems. At the end of step 4, of the 36 articles, 34 (66%) were included. This reduction in studies can be justified by the following aspects: (i) absence of proposed solutions to support decision-making; (ii) many studies discussed the relevance of integration in the context of IoT and decision-making, without introducing a solution for its realization; and (iii) the quality assessment checklist score. Once the papers were mapped, a deeper analysis was necessary in order to identify, evaluate and interpret the studies selected to answer the systematic mapping research questions.

According to the results found in the mapping, it is possible to observe that the use of semantic web and semantic knowledge bases are being widely used. The application of these solutions ranges from irrigation of crops and soil fertilization to the care and management of animals so that they can increase their production. About 14 articles (41%) selected for mapping address some semantic web technique, be it the use of ontologies to infer new knowledge and semantic marks for automatic analysis of intelligent algorithms. The other articles vary their approach in relation to the use of computational intelligence. Neural networks, Support Vector Machine (SVM), and deep learning are also some of the techniques used. Table 1 summarizes the mapping findings.

3.1.3 Results

RQ1: How to integrate Decision Support Systems data from IoT devices in precision agriculture and livestock? Research conducted in [Visconti et al. (2020), Karim et al. (2017), Kamath et al. (2019), Suakanto et al. (2016), Khanna and Kaur (2020), Sudha et al. (2022)] demonstrates the use of a wireless sensor network (WSN) to integrate all IoT sensors in the field in order to provide the largest data set for the decision support systems. In Karim et al. (2017) a prototype of a mobile application integrated with the wireless sensor network is proposed, capable of issuing alerts via SMS so that producers can quickly make decisions about the situation of the soil. Kamath et al. (2019) describe the implementation of a wireless sensor network to monitor rice crops with image capture. The authors integrated the system with RabbitMQ (Kamath et al., 2019), a messaging service capable of persisting sensor data. In (Suakanto et al., 2016) a conceptual model and system design are proposed for decision support in smart farming. The work uses middleware to support the communication and interoperability of smart devices, the detection and characterization of events in real-time and the collection of events for big data analysis. Khanna and Kaur (2020) use a wireless sensor network integrated with cloud servers. In this way, producers can monitor different information from plantations, such as temperature and soil moisture, and intervene in physical locations to balance the environment. The research proposed in (Kamilaris et al., 2016) describes the Agri-IoT, an IoT-based framework capable of integrating various data streams via the Global Sensor Network (GSN), providing complete semantic processing. In [Ting et al. (2022), Sakthi and DafniRose (2022)] the authors propose using blockchain for IoT data integration to ensure security and reliability in the data used for decision-making.

RQ2: How semantic data is handled to integrate Decision Support Systems in precision agriculture and livestock? The works [Symeonaki et al. (2020), Kamilaris et al. (2016), Rotondi et al. (2019), Fawzi et al. (2021)] propose the use of a layer that deals with semantics to interpret data from IoT devices, and Symeonaki et al. (2020) still aggregates context data such as temperature, pressure, wind speed, humidity, lightning UV, among other data that can influence the behavior of animals, crops and plantations. Once these data are taken into account, it is possible that the systems indicate more appropriate decisions in the management of the farms. Kamilaris et al. (2016) explore how the semantic integration of information from various sources, such as: sensors, social media, connected farms, government alerts, regulations, among others, can increase productivity in smart farming. In (Sowmya et al., 2020) the authors propose the use of semantic web to enrich the data generated by IoT devices and then apply machine learning algorithms to optimize soil fertilization. As a result, the evolution from traditional agriculture to precision agriculture and livestock, connected systems based on semantic knowledge, have transformed the physical and technological environment of rural properties. Owners, producers and veterinarians gain a new role in this context: enriching systems through a shared knowledge base, in such a way that decisions are increasingly efficient.

RQ3: What intelligent models are used to support Decision Support Systems in precision agriculture and livestock? The articles [Kakamoukas et al. (2019), Rezk et al. (2021), Sowmya et al. (2020), Borisenko et al. (2019), Kale et al. (2019), Tripathy et al. (2021), Balan et al. (2020), Ting et al. (2022), Sudha et al. (2022), Elijah et al. (2022), Catalano et al. (2022)] presented computational intelligence solutions ranging from SVM techniques, fuzzy logic, decision tree, Petri nets and neural networks. By collecting data generated by IoT devices in the field, it is possible to use machine learning techniques to predict events such as harvesting, irrigation and soil fertilization [Sowmya et al. (2020), Tripathy et al. (2021), Balan et al. (2020)]. All these works presented cloud computing as an important component of the solution since the communication between sensors, systems and repositories takes place through the web. The papers [Mikhaylenko et al. (2019), Rekha et al. (2017), Yusianto et al. (2020), Loret et al. (2020), Zhang et al. (2017), Sakthi et al. (2020)] use knowledge bases with specific rules in the field of agriculture. These rules serve as a basis both for systems to indicate decisions, and for intelligent algorithms to have the correct parameters to make a reliable prediction. In these studies, it is possible to notice variations in approach, such as the automation of decisions through the prediction of events indicated by machine learning algorithms [Kakamoukas et al. (2019), Borisenko et al. (2019), Kale et al. (2019), Sowmya et al. (2020), Tripathy et al. (2021), Balan et al. (2020), Anbananthen et al. (2021), Dayalini et al. (2021), El Hachimi et al. (2021), Ting et al. (2022), Sarma et al. (2022)], to knowledge that can be inferred using the semantic web [Visconti et al. (2020), Karim et al. (2017), Kamath et al. (2019), Khanna and Kaur (2020)]. Additionally, layers that deal with semantics are used to refine machine learning analysis to support Decision Support Systems, making decisions increasingly rich and reliable.

RQ4: How is computing technology applied in precision livestock? After analyzing the studies, it was possible to see how computing is present on farms. In (Dabre et al., 2018) the authors propose two integrated mobile applications, one for the producer and the other for the seller of agricultural products. In the survey, the producer can monitor and send his plantation data to the seller, who, in turn, can indicate the best fertilizer, pesticide and amount of water needed. Some works, such as (Kamilaris et al., 2016), demonstrate the use of “drones” to collect images of agricultural environments. This can be extended to locating animals in the pasture, monitoring their movement activities and detecting sick animals. In this way, “drones” can speed up the decision-making process by providing information in real-time. In [Vazquez et al. (2021), DAYIOĞLU and Turker (2021)] the authors review the use of AI and highlight the use of robots to automate harvesting in a sustainable way, that is, avoiding water waste. The creation of a wireless sensor network also contributes to a system that can manage more than one farm geographically distributed, being able to indicate different decisions simultaneously according to the data collected by the sensors.

3.2 THREATS TO VALIDITY

This mapping aimed to explore issues related to Decision Support Systems in the context of precision agriculture and livestock. However, some threats to validity can be highlighted. Regarding the search string, even using a term review process, it is possible that some term has not been considered for the context of precision agriculture. To mitigate this threat, we use the snowballing technique.

The use of the Scopus database also poses a threat to validity. Other databases not indexed by the Scopus database may contain researches relevant to this mapping. To mitigate this threat, it would be necessary to explore other specific bases, mainly focused on research in agribusiness. However, the database (Scopus) used in this research offered a broad overview of the state of the art of research related to Decision Support Systems focused on Agribusiness with a contribution to advances in computing research.

3.3 DISCUSSION

The related works selected for our research have been identified based on the systematic mapping. By building upon this mapping, we aim to present an analysis of the selected works and their connection to our current research, while also identifying any gaps that they may have left. This analysis will serve to further justify the architecture that will be outlined in the following chapter, providing a solid foundation for our research.

In article (Kamilaris et al., 2016), a framework called Agri-IoT is proposed. It features a semantic framework for IoT-based smart farming applications, which supports reasoning over multiple streams of heterogeneous sensor data in real-time. Although this article describes a framework for intelligent agriculture, it does not address processing through intelligent algorithms, limiting itself only to extract semantic knowledge from information and ontologies to describe the relationship between data.

In order to provide decision support, Anbananthen et al. (2021) use hybrid machine learning technologies that use specialized clustering methods, stacked generalization, gradient boosting, random forest, and least absolute shrinkage and selection operator (LASSO) regression. Although the stacked generalization technique, a model that learns how to combine the best predictions of two or more models trained on the data, is promising, the article does not provide details of how actions in agriculture can be supported. The authors only compare the techniques using cross-validation to identify the most accurate performers for the agricultural dataset. Aiming to support the decision in relation to the soil, articles (Dayalini et al., 2021), (Sarma et al., 2022), (Sudha et al., 2022) and (Elijah et al., 2022) vary their approach, but all have in common the use of machine learning. Article

Dayalini et al. (2021) presents a DSS (Agro-Mate) that helps farmers, through intelligent predictions with machine learning, to determine soil quality, select the best crop, predict rice diseases and predict disasters. Similarly, Sarma et al. (2022) presents a system that has a package of sensors that provides continuous data capture of temperature records, air and soil humidity and a camera for obtaining infrared (NIR) images of the leaves of the plant for use with an AI decision support system. Elijah et al. (2022) proposes a decision support platform (chili-DSP) to detect disease and nutrient deficiency and make prevention decisions. The platform provides a real-time classification of chili diseases. The work (Sudha et al., 2022) presents the Smart Soil Nutrition Prediction (SSNP) system, which adopts sensors to monitor soil conditions. The SSNP aims to support decision-making by predicting soil nutrients in relation to which crop is most suitable to plant.

Despite using intelligent techniques for classification and prediction in relation to the soil, whether for planting rice or pepper, all the works cited above (Dayalini et al. (2021), Sarma et al. (2022), Elijah et al. (2022) and Sudha et al. (2022)) do not explore aspects related to the flexibility of adding other types of data. Furthermore, the systems presented are limited to only the specific domain in which they were developed. Such systems do not address scalability or extensibility to other problems and domains. They are not prepared to deal with adverse data, or adapt to a change in data, such as the emergence of new diseases, soils with different conditions, but which may be conducive to planting, and they are also limited to the training datasets used initially. By using semantics, algorithm training could be improved and thus provide the best decision-making in relation to climate data, diseases and other external variables.

The work Catalano et al. (2022) proposes a new approach for detecting anomalies in intelligent agricultural systems. The authors' objective is to mitigate weaknesses, intentional and unintentional failures in data and information management in IoT environments. Although the design of the proposed architecture is based on an algorithmic approach to machine learning by a multivariate linear regression (MLR) and a long-term memory neural network (LSTM) algorithm, the authors do not explore the use of other algorithms, not address the sustainability of the architecture, or use semantics to extract strategic information from the data that is analyzed by machine learning.

Table 2 compares all related articles considering the architecture proposed in this work.

3.4 FINAL REMARKS OF THE CHAPTER

This systematic mapping identified, classified and analyzed computational solutions in Decision Support Systems in precision agriculture and livestock. Initially, 155 articles were identified by the selected source. They were filtered, resulting in 53 articles. After this selection, the snowballing technique was performed, adding 2 more new articles, totaling

55 papers. After the quality analysis, the result totaled 34 articles.

According to the results of this mapping, research opportunities can be observed from the search for state of the art in DSS in Agriculture 4.0, related to IoT and sematic. By advancing knowledge on the subject, we identified opportunities and integration techniques used in the works, demonstrating the importance of using data from IoT sensors to support decisions. As a result, complex decisions aimed at agribusiness can be enriched with information existing in other systems used by rural properties. Once the literature mapping is carried out, we define the problem and requirements for the solution. We aim to answer the following research question: "How can e-Livestock support automated monitoring, reasoning, and actions in smart farms?"

To answer the research question, we carried out two implementation cycles under the Design Science Research (DSR) methodology. In the first cycle, after carrying out the systematic mapping, we conducted technical interviews with researchers and specialists from Embrapa Gado de Leite to better understand the application domain. During these interviews, we collected important animal management, welfare, and health information. It was also important to understand the whole milking process and the necessary care for the animal so that the milk has quality. After the interviews, we went to the development phase of the solution, where we built the e-Livestock architecture artifact.

As stated above, we adopted the Design Science Research (DSR) methodology to evaluate and guide the proposed architecture. The next chapter details the architecture design, as well as each DSR cycle starting from the mapping.

Table 1 – Summary of mapping report findings

Studies	Application
Symeonaki et al. (2020)	Context and Middleware Awareness
Borisenko et al. (2019)	Computational Intelligence with Petri Net
Kale et al. (2019)	Deep Learning
Balan et al. (2020)	Computational Intelligence with Neural Networks
Kamilaris et al. (2016),Rotondi et al. (2019)	Ontologies and Semantic Web
Rezk et al. (2021), Sowmya et al. (2020)	Computational Intelligence with Decision Tree
Ting et al. (2022), Sakthi and DafniRose (2022)	Blockchain
Sowmya et al. (2020), Tripathy et al. (2021)	Compute Intelligence with SVM
Dabre et al. (2018), Yusianto et al. (2020), Suakanto et al. (2016)	Statistical Analysis
Visconti et al. (2020), Karim et al. (2017), Kamath et al. (2019), Khanna and Kaur (2020)	Semantic Web
Mikhaylenko et al. (2019), Rekha et al. (2017), Loret et al. (2020), Zhang et al. (2017), Sakthi et al. (2020), Fawzi et al. (2021)	Semantic Knowledge Base
Kakamoukas et al. (2019), Vazquez et al. (2021), DAYIOĞLU and Turker (2021), Anbananthen et al. (2021), Dayalini et al. (2021), El Hachimi et al. (2021), Sarma et al. (2022), Sudha et al. (2022), Elijah et al. (2022), Catalano et al. (2022)	Machine Learning

Source: Elaborated by the author (2023).

Table 2 – Summary of techniques from related works.

Work	ML	Semantic	Self-adaptive	Flexible	Extensible
Agri-IoT	-	X	-	X	-
DSS with hybrid ML algorithms	X	-	-	X	-
Agro-Mate	X	-	-	-	-
DSS with NIR	X	-	-	X	X
Chili-DSP	X	-	-	X	-
SSNP	X	-	-	-	-
Anomaly detection	X	-	-	-	-
e-Livestock	X	X	X	X	X

Source: Elaborated by the author (2023).

4 ARCHITECTURE DECISION E-LIVESTOCK

The previous chapter presented the main related work through a systematic mapping. Based on the results, we identified some gaps in these works regarding the need for a more detailed analysis of the data coming from sensors. None of the identified works combined semantic models and AI algorithms for more precise analysis. Therefore, we developed e-Livestock, an architecture to support data analysis from sensors and other devices used in precision livestock farming to assist decision-making in agribusiness. The e-Livestock was used in the Compost Barn infrastructure. The Design Science Research (DSR) methodology was used in this study. In this chapter, we detail the DSR process and the construction cycles of the architecture.

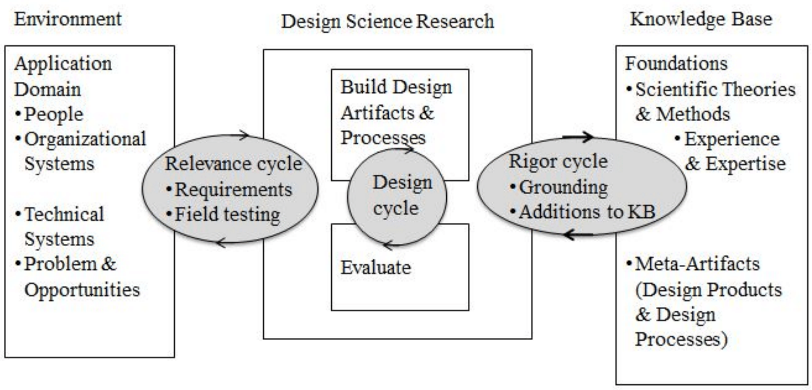
4.1 METHODOLOGY

Design Science Research (DSR) is a methodology driven by the continuous improvement of a solution by introducing new artifacts and the construction processes of these artifacts (Simon, 1996). An application domain comprises people, organizations, and technological systems interacting toward a goal. DSR research usually starts by identifying and representing opportunities and problems in a real-world application environment.

Thus, the relevance cycle initiates the research with an application context that not only provides the requirements for the research (e.g., the opportunity/problem to be addressed) as inputs but also defines acceptance criteria for the final evaluation of research outcomes. The research output should be returned to the environment for study and evaluation in the application domain. Domain study of the artifact can be conducted through appropriate technology transfer methods such as applied research (Cole et al. (2005); Järvinen (2007)).

The results of the domain test will determine whether additional iterations of the relevance cycle are needed in this Design Science project. The new artifact may need more functionality or inherent qualities (e.g., performance, usability) that may limit its usefulness in practice. The resulting artifact may indicate a need for new requirements and even reveal mistaken or incomplete requirements. Another iteration of the relevance cycle will start with feedback from the research environment and reaffirm research requirements as discovered from experience. During the relevance cycle, we raise the requirements for the artifact to achieve its goal. Also, in this cycle, it is necessary to analyze the application context, the people involved, and the organizational systems. This analysis is done to ensure that the requirements are in line with the problem to be solved. The design cycle encompasses the construction of the artifact, the implementation and development process, and evaluation. We can generate a product, process, or scientific knowledge at the end of the artifact construction. The rigor cycle is where theoretical grounding occurs, guiding the construction of the artifact, whether by methods, theories, or processes available in the literature. In this cycle, it is verified whether the initial theoretical conjectures are correct based on the evaluation of the artifact. A three-cycle view in Design Science Research is presented in Figure 3, where the relevance, design, and rigor cycles mentioned above can be seen.

Figure 3 - DSR Flow



Source: Hevener (2007)

In the next section, we define the functional and non-functional requirements that guided the implementation of the architecture.

4.2 REQUIREMENTS

For the architecture specification, we identified which functional and non-functional requirements were a priority for its development. Therefore, to meet the needs of a decision support system for Agriculture 4.0, such as sensor data processing, storage and visualization, functional and non-functional requirements were derived.

4.2.1 Functional Requirements

FR 001. The architecture must be able to process different sets of Compost Barn data.

FR 002. The architecture must be able to integrate data from external sources, such as meteorological services and geolocation, to help enrich the Compost Barn data.

FR 003. The architecture must be able to perform semantic and prediction analysis.

FR 004. The architecture must allow efficient storage of data extracted from the Compost Barn.

FR 005. The architecture must provide mechanisms for viewing Compost Barn data in graphs or tables, thus supporting users in interpreting information and making decisions.

FR 006. Data must be pre-processed to be sent for intelligent analysis.

4.2.2 Non Functional Requirements

NFR 001 (Dependability). The solution must allow communication with external data sources.

NFR 002 (Product). The solution must respect the principles of extensibility so that the architecture accommodates the system's future growth.

NFR 003 (Product). The solution must respect the principles of flexibility. Flexibility is an attribute that reflects the range of behaviors that the existing architecture can be configured to address, for example processing animal health, production, and environmental data.

NFR 004 (Product). The solution must respect the principles of scalability. Scalability considers adapting the system to new size and scope specifications, so the architecture must allow adding new data sources and dynamically performing intelligent processing.

NFR 005 (Sustainability). The solution must consider the sustainability of the software through the continuous training of the intelligent models (when new datasets arrive), to keep the insights offered by the system (for example, prediction and estimation of sick animals) consistent and updated.

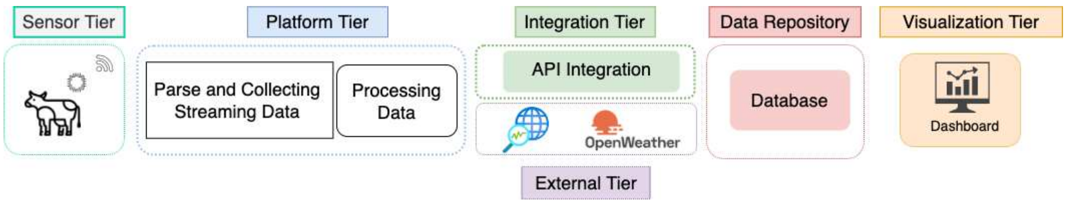
NFR 006 (Usability). The system must be user-friendly. In addition, malfunctions during data visualization can prevent the user from making the most appropriate decision. In this way, the user would be negatively affected, threatening the main function of the solution, which is decision-making..

4.3 FIRST CYCLE OF E-LIVESTOCK ARCHITECTURE

The DSR methodology operates in cycles, during which artifacts and processes are developed, revisited, improved, and evolved. New requirements, problems, and opportunities for improvement may emerge during each cycle. During the implementation of the architecture, multiple versions were developed, with new requirements being added and the solution being improved at the end of each cycle. A systematic mapping was conducted to establish the conjectures for the first version of the architecture.

Following an analysis of the studies, an architecture based on tiers was developed to analyze data from sensors and other devices related to precision livestock. Initially, these tiers collected data from internal and external sources, processed the data, and presented it in a dashboard. Six tiers were defined for the initial version, including the processing, integration, external data, data repository, and visualization tiers, which were developed to meet both functional and non-functional requirements. The first version of the architecture is illustrated in Figure 4. During the Case Study, we will provide a detailed account of the operation of each layer, as well as the manner in which data was integrated and stored.

Figure 4 - First version of the architecture



Source: Prepared by the author (2023)

The Sensor Tier is responsible for data generated by sensors, while the Platform Tier collects and processes different data types according to FR001. The integration tier adds external data to the sensors data, as required by FR002, NFR001, NFR002, NFR003, and NFR004, before storing the data (FR004). Finally, the data are presented in a dashboard with graphs (FR005) to allow producers to visualize and interpret the data. During the first cycle, the state of the farm was analyzed in terms of total milk production, the amount of food ingested by the animals, the total number of sick animals over time, and data on environmental conditions (temperature and humidity), among other information. The dashboard shown in Figure 5 provided producers with the data required to make day-to-day decisions based on the Compost-Barn data captured through sensors.

Figure 5 - Compost Barn Dashboard

Source: Prepared by the author (2023)

The classification of animals in batches based on their milk production is denoted by the terms "Lote 1", "Lote 2", and "Lote 3."Notably, the animals in the "Lote 3"exhibit the highest milk production, while those in the "Lote 1"demonstrate comparatively lower milk yields. However, the need for more detailed analysis arose, such as checking milk production for the next month or visualizing weather conditions for the week. The data presented in the dashboard did not allow for such analysis or predictions to be made, such as milk production forecasts, animal food consumption estimates, or relating mastitis cases with data from sensors in the environment. Therefore, functional requirements FR003 and FR006, and non-functional requirement NFR006, were identified as necessary and addressed in the next cycle.

4.4 SECOND CYCLE OF E-LIVESTOCK ARCHITECTURE

The second cycle encompasses the introduction of the Intelligence tier, a new analysis service that enhances decision-making on farms by providing forecasts for producers. The FR003 and FR006 requirements have been met, allowing for semantic and predictive analyses, which offer accurate predictions regarding milk production, food consumption, and animals with mastitis, among other factors. With these insights, producers can plan for the future with confidence.

The Intelligence tier comprises a semantic model (an ontology) extracting implicit data knowledge. This knowledge includes relationships between milk production and animal feeding, weight evolution, and diseases like mastitis. Ontological rules can be applied to derive missing data that intelligent models may not be able to infer, such as temperature and humidity records inside the Compost Barn that were not captured by sensors.

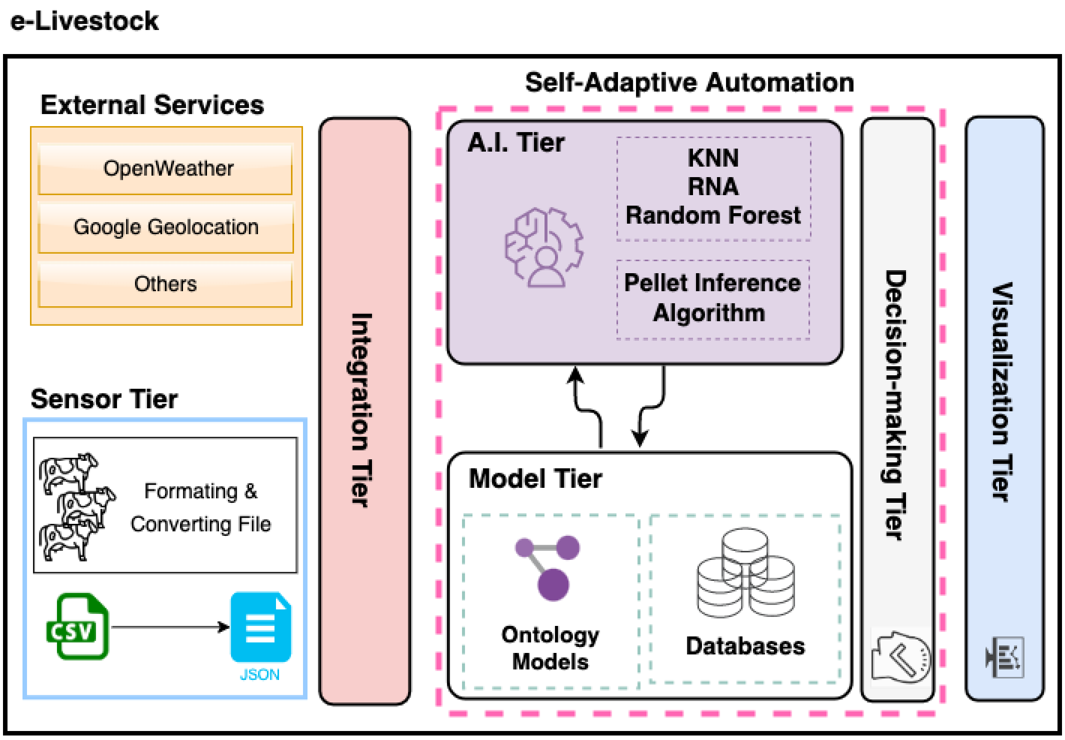
Furthermore, the Intelligence tier can classify animals as "healthy"or "sick"and mitigate the risk of epidemics by sorting mastitis-infected animals by lot. These features are accessible through SWRL rules processed by reasoners. Figure 6 provides an overview of the architecture, including its respective tiers.

4.5 ARCHITECTURE COMPONENTS

We have created a comprehensive diagram illustrating the various components and their connections. The architecture is based on the MVC pattern (Deacon, J., 2009), separating different project parts to minimize dependencies. We can easily meet scalability and extensibility requirements by ensuring uniformity in software structure. The reduced code complexity also makes it easier to maintain applications, facilitates documentation for future work, and allows for the reuse of system modules. The tiers of the architecture are described in detail below.

Sensor Tier: Collects data generated by sensors deployed on the farm. Sensors gather information such as temperature and humidity, among others. In addition, the

Figure 6 - e-Livestock Overview



Source: Prepared by the author (2023)

sensor tier also handles different data formats generated by internal systems on farms. These data are sent to the integration tier.

Integration Tier: That tier is responsible for processing the collected data to be integrated with information from other sources, services, and external APIs, such as context data, and environmental information (temperature, humidity, and weather forecast). The main advantage of this tier is that it can aggregate external data to enrich information for decision-making process. For instance, the REST API allows communication with other tiers, such as the Intelligence tier. The AI tier communicates with the Integration and Model tiers by receiving the data already processed. Then, it executes the most suitable intelligent algorithm for a given dataset. The AI tier sends the results, predictions, or classifications, to the API, to be persisted.

External Services Tier: it represents external services, databases, historical bases, and any external data sources that can add value to the data collected by the Sensor Tier. We can add new sources to the system through the Integration Tier as needed. For example, by aggregating weather forecast data, it is possible to provide a new perspective for decision-making. By recording data source, sensor, and data type, it is possible to track and analyze the context of decisions that used this information. Once the data is aggregated and stored, it is possible to use the ontology to run inferences in the new data and retrieve new relationships.

Model Tier: This tier deals with farm data. By integrating external data, it is possible to store data generated on the farm and enable the generation of dashboards. By inferring relations between animal feed and milk production, we can identify the best diet to increase milk production. The Model tier also stores the models' metadata, such as model accuracy, average errors, algorithm type, and input dataset used. Consequently, it is possible to analyze the results, which can be used to make future decisions. For example, with the prediction of food consumption, the researcher can estimate the expected cost of purchasing inputs and plan storage according to the probability of consumption of animals indicated by the algorithm.

Decision-making Tier: In this tier, the business logic is defined and aggregated to the Intelligence tier results to present the resulting data in the Visualization tier. For example, alerts and notifications whether the production is low or many animals are contaminated with mastitis or other diseases. In these cases, rules are set in this tier. Besides, all relevant information generated in the previous tiers is organized and prepared to be viewed by the user.

Visualization Tier: The visualization tier allows the researcher to visualize the data in real-time through a panel according to a time interval. The researcher can also analyze and interpret data at different granularities. It allows users to visualize (FR 005) the results of the AI tier.

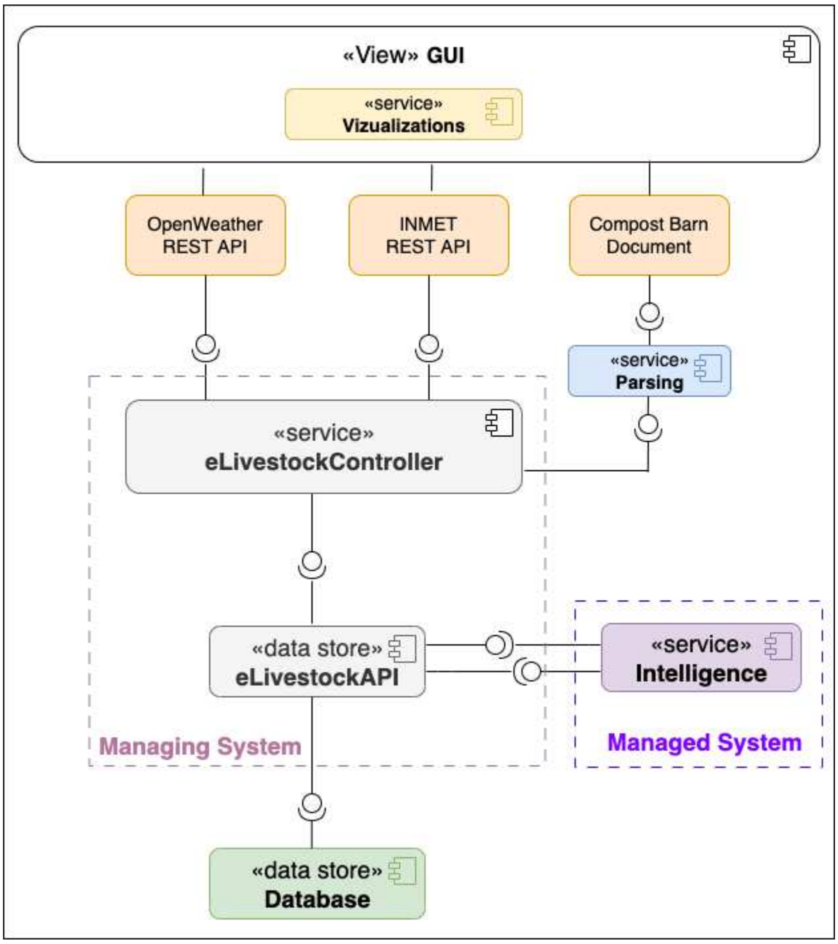
The visualization tier comes with a user-friendly graphical interface that displays aggregated data from the farm and external APIs. The visualization tier can be easily modified to suit other devices like mobiles and wearable techs. This is possible because we developed the visualization tier to access data via REST API. Therefore, if the device has internet access and the appropriate user interface, it is possible to receive the data via HTTP request and display it.

We developed Figure 7 based on Figure 1, a conceptual model of a self-adaptive system, in which the eLivestockAPI is the managing system and the intelligence is the managed system.

For collecting the farm's external data such as temperature and humidity, we used the API of the Instituto Nacional de Meteorologia (INMET). Additionally, we can connect other weather station APIs, such as Open Weather, if needed. Our architecture's low coupling and high cohesion also allow for the addition of other services to support producers, such as geolocation services. Moreover, the Compost Barn data can be easily viewed, which is managed by eLivestockController and pre-processed by parsers. These parsers are present in the integration layer. They are scripts that we developed to handle and process incoming data to ensure that it is sent in a normalized, organized, and easy-to-visualized format in the graphs.

The eLivestockController communicates with the eLivestockAPI component (that

Figure 7 - Component Diagram.



Source: Prepared by the author (2023)

can be considered as the Managing System), which is responsible for interacting with the Model tier where the database is stored. This interaction takes place for both reading and writing. Our architecture allows adding other databases by simply connecting them to the eLivestockAPI. Furthermore, we have designed our architecture to be flexible, allowing the connection to other APIs. This flexibility ensures that the system meets the demands of different contexts.

Finally, the eLivestockAPI also handles requests related to the Intelligence tier, which we will discuss in depth in the next section. Please refer to Figure 7 for the component model used in implementing our architecture.

4.6 ARCHITECTURE DEVELOPMENT

We used Python and Javascript language to develop the components. We chose Python in the parsing and intelligence layer due to the facilities for handling text files, CSV, and spreadsheets. In addition, it also allows the manipulation of ontologies with the OWL2Ready[3] library and has several other libraries to implement AI models, such as Scikit-learn[4]. We developed a RESTful API with Node.js (Javascript) for the integration control tier and used a NoSQL database in the repository tier, i.e., MongoDB. This non-relational database was chosen for the flexibility to create new collections and unstructured data.

The communication between the parsing tier, external services, and the intelligence tier was made via HTTP request. Hence, we can concentrate and standardize the requests in the architecture, regardless of the requested data. Thus, the Visualization tier can be simplified and focused only on presenting data to users to support decision-making. To present the dashboards, we developed a mobile application in Javascript, using ReactNative framework, called eLivestock Monitor. The application is a visualization tool, where the producer can consult farm data, perform intelligent analysis, and monitor the environment.

In the next section, we detail the intelligence processing in e-Livestock, considering that this tier is the main contribution of the new architecture version.

4.7 INTELLIGENCE AND DECISION E-LIVESTOCK

Smart farm systems, or Agriculture 4.0 systems, support high connectivity through connected sensors or IoT devices. In this type of system, we have a scenario composed of sensors that share information about events, context information, and computer systems capable of storing, processing, and analyzing data. Moreover, it is possible to connect sensors, wired or not, and transmit status signals or even data related to the grain production results or animals' health, contextual information (such as environmental conditions), and production process. A smart farm architecture uses these resources to collect, store, process, analyze data, and adapt the system to the new detected and sensed conditions.

Monitoring and collecting data from sensors are aggregated, preprocessed, and stored before being sent to the intelligence tier. To process the information, e-Livestock uses two techniques: inference processed over an ontology and predictions based on Machine Learning (ML) algorithms. After preprocessing, it is possible to instantiate the data in the ontological model and execute the inference mechanisms, processing SWRL rules. Additionally, it is possible to compose a dataset with the stored data for training and testing purposes for intelligent ML algorithms.

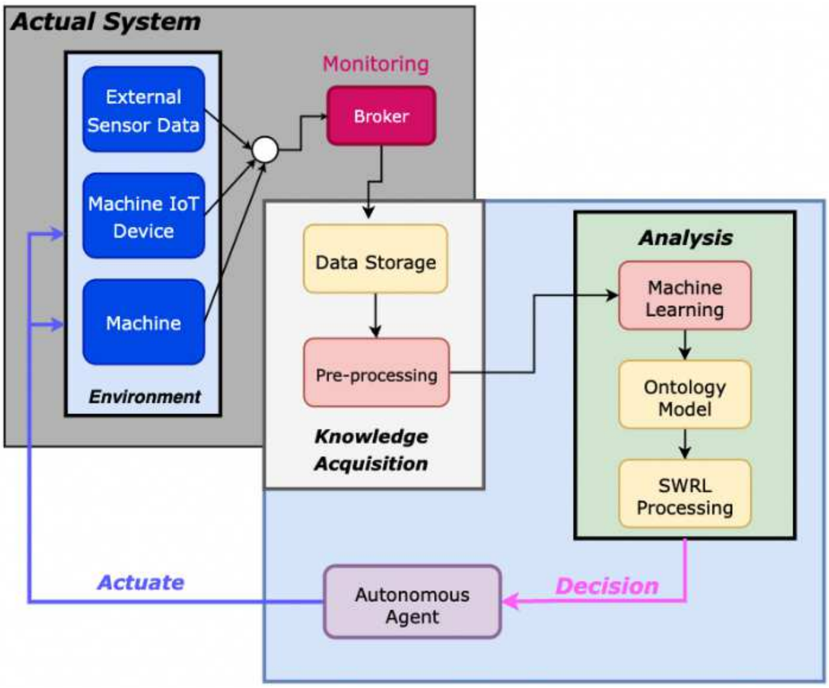
The Intelligence tier was developed to meet the requirements of flexibility (NFR003), scalability (NFR004), and sustainability (NFR005). Hence, the architecture must be flexible and allow the execution of different ML models. In addition, we promote scalability by allowing new analyses and new sensors to be made and installed. The sustainability of the

[3] https://owlready2.readthedocs.io/en/v0.37
[4] https://scikit-learn.org/stable

architecture takes place through the self-adaptative capability and continuous training of intelligent models (when new datasets arrive), which allows for keeping the insights offered by the system (for example, prediction and estimation of sick animals) consistent and up-to-date. Even if the pattern of the farm changes, considering the adoption of a new breed of animals, or a new diet with components that have yet to be used, this should not impact the system's functioning. The architecture must be able to deal with change and continue to offer results to support decision-making in this new scenario.

Therefore, to support these demands, we specified self-adaptive capabilities as a new specialized component, part of the e-Livestock architecture, to support sensor data analysis in the Agriculture 4.0 scenario. Figure 8 presents the main components of the e-Livestock self-adaptive architecture, showing the scenario where the devices communicate and the infrastructure provided for collecting, storing, and processing information. The e-Livestock architecture solution encompasses two main components: the AS (Actual System) and the MS (Managing System) as shown in Figura 1.

Figure 8 - e-Livestock self-adaptation main components.



Source: Prepared by the author (2023)

The AS system is responsible for the system's day-by-day operation and data capture. It has two main modules, i.e., "environment" and "knowledge acquisition". The first module, "Environment", connects the sensors to the data network and transmits sensor data. IoT devices connected to animals and other farm features, such as harvesters, allow data such as health conditions or grain production efficiency to be computed and transmitted through existing communication sockets. This module is also responsible for real-time capturing the context data from the environment and the closest operators so that MS can create specific alerts for quick decision-making by the operators. The second AS module, “Knowledge Acquisition”, has the “Pre-processing” component responsible for data cleaning and formatting. After pre-processing, the data is processed by machine learning. The architecture automatically performs clean-up and pre-processing tasks when the system detects new sensor data. In this model, we have the devices connected to a broker that acts as a dispatcher for IoT devices and connected sensors, and then the subsequent data is stored on a server.

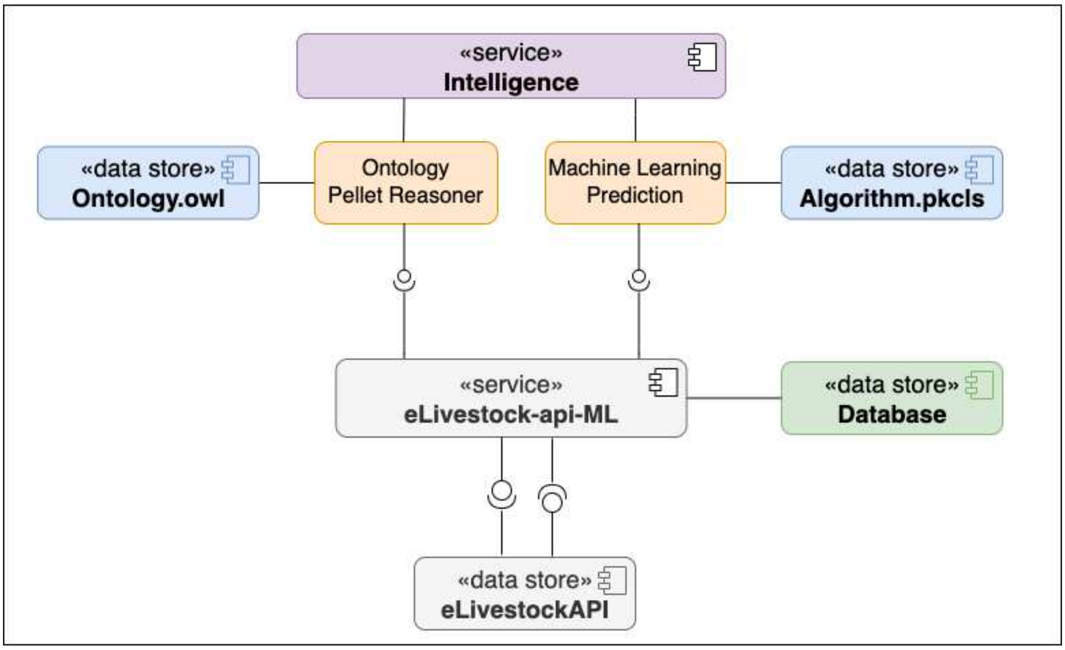
The Managing System (MS) encompasses an autonomous agent capable of detecting environmental changes. When identifying that a sensor failed to read temperature data, the system can automatically run the ontology reasoner to infer the data and keep the system updated and cohesive. Another case in which the system adapts to the situation is when milking data (milk production) is added or imported; the system automatically performs training and then runs ML algorithms to update the dashboard showing the (estimated) prediction of milk production based on the new data. In addition to milk predictions, when registering new animals on the farm, the system instantiates the data in the ontology. It performs inferences on the data of healthy and sick animals (with mastitis) to organize the batches and predict milk production for these animals.

We can observe, from Figure 8, that there is an overlay between the IoT components and the self-adaptive mechanism. The Actual system, which interacts with the environment, relies on sensors and actuators so that IoT components feed reasoning and self-adaptation mechanisms. The MS system provides strategic information to support the decision process. The “Machine Learning” module analyzes animals’ or grain production and environmental data to estimate the probability of production loss or animal disease, among other critical events.

The data collected by the sensors are sent to the Ontology module to organize the data. Based on logical rules, “SWRL Processing” provides information and relationships that can activate the “Autonomous Agent”. Once sent to the Machine Learning module, we can more accurately indicate the possibility of production loss or sick animals through the semantic data extracted from Ontology. The “Autonomous Agent” can trigger alerts or modify the AC, based on information from the previous two MS modules. Therefore, the Autonomous Agent can search for the available devices or operators that must receive the alerts and/or process changes directly on the devices or in the AC functioning, such as turning on a temperature regulator or sending alerts to the harvesters´ operator to speed up the harvest, for example. As the sensors collect new data, the autonomous agent continuously monitors the data to process it.

The ML module was designed to give insights based on the farm data set. These insights are related to milk production, mastitis type classification, and food consumption estimation, to name a few. Based on the results from the ML algorithms, we can identify gaps in the production, check the average consumption per animal and type of food and it is possible to optimize the amount of adequate feed to be supplied to the animals. We can contribute to a more sustainable farm as they can prevent food waste. With this action, farmers can avoid spending unnecessary resources purchasing commodities such as corn and soybeans, reducing the economic impact on the farm. Figure 9 presents the component diagram of the Intelligence service.

Figure 9 - Intelligence Tier Component Diagram.



Source: Prepared by the author (2023)

The Intelligence service includes two modules. The first module, Ontology.owl, uses Pellet as Reasoner's algorithm to make inferences and save the results in an OWL file. The second is Machine Learning Prediction, which exports the trained algorithms in the pkcls extension. These two modules communicate with the Integration tier through a Python API called eLivestock-api-ml. This API was developed using the Flask library and received all requests related to the Intelligence tier from the Integration tier (eLivestockAPI). The eLivestock-api-ml API can also communicate directly with the database, offering more performance when instantiating the ontology. You don't need to go through the Integration tier when you query data. This communication was possible thanks to the PyMongo[5] library.

In the following sections, we detail the "Ontology Model" and "Machine Learning" modules of the e-Livestock architecture, considering their importance to the e-Livestock architecture.

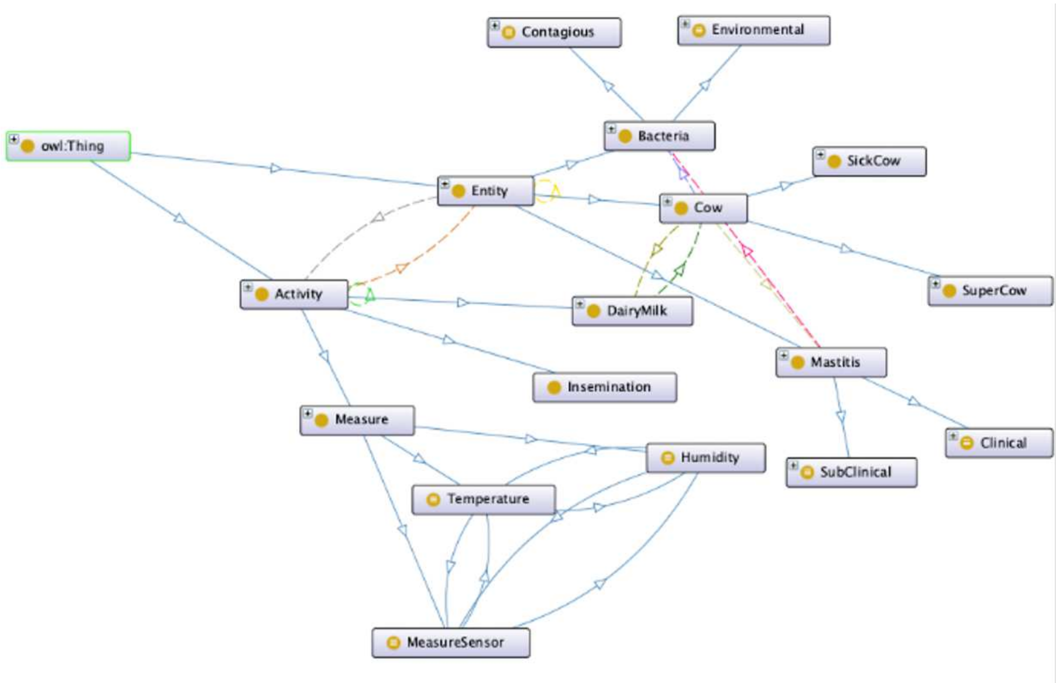
[5] https://pymongo.readthedocs.io/en/stable/

4.7.1 Decision e-Livestock Ontology

The ontology model was designed to capture the relationships between data for dairy cattle. The use of ontology helps to understand how data is connected and to generate better datasets that can be sent to ML algorithms. Through inference mechanisms, we can fill gaps instead of discarding data during training AI algorithms. Figure 10 illustrates a partial view of the ontological model.

The entities represent animals, agents are the farmers/researchers/producers, and activities are any action carried out on the farm. Activities can be described as insemination, milking, or processing data. With this model, it is possible to identify the data sources and the interactions that researchers and farmers carry out. As a result, it is possible to track decisions related to these specific activities.

Figure 10 - DairyCattleOntology main classes and associations.



Source: Prepared by the author (2023)

To implement the model shown in Figure 10, we used Ontology Web Language (OWL) 2.0. The ontology model is based on Competency Questions (CQ) to accommodate dairy cattle production necessities. A CQ is a natural language sentence expressing a pattern for a question that people/computational applications expect an ontology to answer (Uschold and Gruninge, 1996). We elaborated on these CQ based on interviews with researchers and farmers and documents related to the Compost Barn production system. We used ontological concepts, including their classes, relations, and inference rules to answer these CQ. The ontology model was developed to help producers make decisions based on data. The Competency Questions describe what was expected from the ontology to answer in the dairy cattle domain considering the e-LiveStock architecture. Through this model (Figure 10), we provided the following Competency Question (CQ).

(CQ1) How much was an animal's production reduced due to inflammation (mastitis)?

(CQ2) Did the average production of an animal drop due to a temperature change?

(CQ3) Did the temperature variation make the animal spend more energy maintaining body temperature than producing milk?

(CQ4) Did the average mastitis cases grow due to increased humidity, favoring the proliferation of environmental bacteria?

Considering ontology, we used object properties (OWL constructs) to implement the relationships between classes. To answer CQs and discover new associations between farm activities and animals, we created SWRL rules (Chen et al., 2021). We built the specific rules (Figure 11) and executed them (Figure 12) based on information from Compost Barn, provided by researchers from the EMBRAPA.

Figure 11 - SWRL Rule – Assessing Temperature and Humidity..

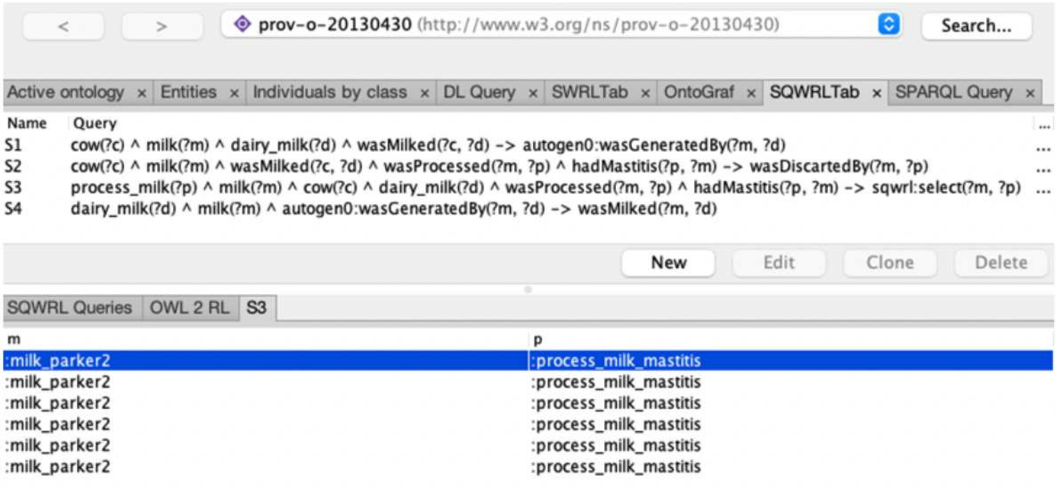
```
humidity_rule.set_as_rule("""Measure(?m), internal_measure_value(?m, ?v), greaterThan(?v, 95)
                            -> is_alert(?m, 1)""")

temperature_rule.set_as_rule("""Measure(?m), internal_measure_value(?m, ?v), greaterThan(?v, 23.5),
                               lessThan(?v, 45) -> is_alert(?m, 1)""")

sickcow_rule.set_as_rule("""Cow(?x), is_mastitis(?x, ?m), equal(?m, 1) -> SickCow(?x)""")

supercow_rule.set_as_rule("""Cow(?c), DairyMilk(?d), cow_milked(?d, ?cm), dairy_value(?d, ?v),
                            greaterThan(?v, 30), is_mastitis(?c, ?m), equal(?m, 0), SameAs(?c, ?cm) -> SuperCow(?c)""")
```

Source: Prepared by the author (2023)

By utilizing the stated model (explicit knowledge) in conjunction with the incorporation of particular SWRL rules (depicted in Figure 11) and an inference mechanism, our ontology model deduces the relationships among instances in response to the activities transpiring on the farm (implicit knowledge). These relationships are new knowledge from processing SWRL rules and inference engines about ontological instances. The "humidity rule" infers an alert for humidity, and the "temperature rule" infers an alert for temperature. The "sick cow rule" looks for animals that have mastitis and classifies them as sick animals. The "super cow rule" seeks cows with good milk production and ranks them as super cows.

Figure 12 depicts the outcomes derived from the ruleset as follows: Rule S1 infers milked animals; Rule S2 looks for animals with mastitis for which the milk has been

Figure 12 - SWRL Results.



Source: Prepared by the author (2023)

discarded; Rule S3 looks for milk discards caused by mastitis; Rule S4 seeks milk production. By mixing the rules and combining their results, we can answer the CQs. To answer CQ1, we can run Rule S2 to get the animals that mastitis events discard the milk. The ”temperature rule” and Rule S1 can answer CQ2, which returns the list of animals milked and those days with higher temperatures. A crossing between these two lists can justify the reason for low production. The “temperature rule” also answers CQ3. Through the “humidity rule” and Rule S3, we can find those milk discards caused by mastitis (CQ4).

For instance, we need to combine ontology with machine learning to calculate future milk production according to the month of the year and based on the weight of an animal. Due to the need for consistent meteorological data throughout the year, ontology inferences can provide the necessary data to fill these missing columns. Once we fill these gaps, we can send them to ML algorithms alongside the animal’s weight and then make the correct milk prediction.

4.7.2 Predictions Based on Machine Learning Algorithms

The machine learning module was created to enable comprehensive data analysis on farms, providing valuable insights for decision-making. Simply presenting data on dashboards is not enough to help producers make decisions. Through machine learning, we can provide milk production projections, identify potential sick animals, and suggest the most suitable diet to increase milk production.

To implement the machine learning module, we started by using a tool called Orange Data Mining[6], which is an open-source data analysis tool with a diverse range of

[6] https://orangedatamining.com/

classification and regression algorithms. Despite its graphical features and the ability to export pre-trained algorithms, it did not allow for automated training. This meant that every time we needed to train the data, we had to manually update the algorithms and export them for application use.

We decided to develop the machine learning module using Python and Scikit-learn, which allowed for automated training with a list of algorithms and the selection of the best-performing one. This library also offers more advanced techniques such as natural language processing, shrinking sets of plain text items, and managing association rules, as well as various intelligent algorithms for classification and regression.

The architecture was designed to support a list of ML algorithms, aiming to support the treatment and manipulation of data for training. Hence, if we need to add a new feature, we can extract the entire set of data from the database. For example, if it is necessary to predict the weight of the animals in a herd for the next month, based on their diet, we can query the list of all animals in the herd in the database, the data referring to their feeding, the history of weight and sort by date. After extracting and formatting the data, it is necessary to train the algorithms. It is enough to perform the ML training with the dataset, which in the end will automatically select the algorithm with the best precision for the analyzed problem. Different algorithms may present different performance and accuracy for each problem. However, the architecture will select the one with the lowest mean absolute error (MAE). The Intelligence tier relates to the storage tier by the API, so the ML module has access to data through the controllers of the integration tier.

According to the NFR002 extensibility requirement, the architecture can accommodate new controllers and provide new endpoints for the Intelligence tier to access more farm data. The original architecture project contains mechanisms that facilitate system expansion, as creating an endpoint and connecting it to the Intelligence tier is all that is necessary to provide data access. New intelligent functionalities can be developed, and the data from the endpoints can be consumed, enabling future system growth. Flexibility (NFR003) is an attribute that reflects the range of behaviors that the existing system can be configured to meet. If producers require insights that the system does not provide, this functionality can be created to meet the demand. Therefore, training takes place within the architecture to dynamically provide algorithms with better results and accuracy. Finally, the NFR004 scalability requirement considers adapting the architecture to new size and scope specifications. A non-relational database was chosen to meet this requirement due to the ease of adding and changing fields without causing significant changes to the application. Therefore, if new predictions are necessary, the architecture is ready to scale, whether by storing new sets of different data or increasing the list of available intelligent algorithms.

The construction of the data processing steps by intelligent algorithms begins with the collection of raw data from the farm. First, it is necessary to pre-process this data. Many datasets enter the system through the reading of spreadsheets, so for each row of the file, data is processed. There are several methods for handling the preparation and transformation of the initial data set. All those methods are done automatically. The methods used in the architecture can be divided into:

1. Data cleaning, which consists of filling in missing values, identifying, or removing outliers, and resolving inconsistencies;
2. Data integration, it may be necessary to aggregate from other data sources, such as temperature data, which may cross with meteorological station data;
3. Data transformation is the process of normalization and aggregation, and finally;
4. Data reduction, when representation is reduced in volume, but produces the same or similar analytical results.

During the process of data cleaning, ontologies can be used to add missing data through SWRL rules and derivation mechanisms, as presented in Figure 11. Additionally, the Integration tier aids in this process by aggregating external sources. For the scope of this project, we used the National Institute of Meteorology API as a supplement to temperature and humidity data. By enabling the integration of other sources, we adhere to the principles of dependability (NFR001), flexibility (NFR003), and scalability (NFR004) as the architecture can communicate with other systems as required by the system, thus providing new data for subsequent intelligent algorithm processing. The data transformation process is critical to normalize data units, such as converting grams to kilograms or milliliters to liters. Data reduction, on the other hand, involves eliminating duplicate entries. Once the data is prepared, it is persisted in the database.

Through the Integration tier, we can connect the database tier with the Intelligence tier. Consequently, the architecture allows the creation of custom queries to retrieve data from more than one table or collection, thereby generating a dataset to be used by ML algorithms. Each feature, such as predicting milk production, may require one or more types of data, such as milk production history and animal weight. Therefore, it is essential to organize the data first and then send it to the Intelligence tier.

The controllers are responsible for querying the database, sorting, and aggregating related entities. Once organized, this data can be written to a dataset and exported in CSV, TXT, or XLS formats. With the generated dataset, we can train intelligent algorithms and test each of them to evaluate their accuracy.

The training process begins in the ML module by ensuring that the dataset is available and specifying the number of inputs and expected output type. The default

Figure 13-A - Running list of algorithms.

```python
def run_all_models():
    print("Trainning....")
    values = { 'start': 1, 'end': 2} # peso
    label = 2 # leite
    X_train, X_test, y_train, y_test = split_train_data('data/pesoXleite.csv', values, label)

    print("RandomForestRegressor: ")
    regressor = RandomForestRegressor(n_estimators=200, random_state=0)
    run_model(BASE_NAME+"RandomForestRegressor.pkcls", regressor,  X_train, y_train, X_test, y_test, "Random Forest")

    print("\nNeural Network MLPRegressor: ")
    regressor = MLPRegressor(random_state=1, max_iter=500).fit(X_train, y_train)
    run_model(BASE_NAME+"MLPRegressor.pkcls", regressor,  X_train, y_train, X_test, y_test, "Neural Network")

    print("\nKNN KNeighborsRegressor: ")
    regressor =  KNeighborsRegressor(n_neighbors=2)
    run_model(BASE_NAME+"KNeighborsRegressor.pkcls", regressor,  X_train, y_train, X_test, y_test, "KNN KNeighbors")

    print("\nKNN RadiusNeighborsRegressor: ")
    regressor =  RadiusNeighborsRegressor(radius=1.0)
    run_model(BASE_NAME+"RadiusNeighborsRegressor.pkcls", regressor,  X_train, y_train, X_test, y_test, "KNN Radius Neighbors")

    print("\nAdaBoostRegressor: ")
    regressor = AdaBoostRegressor(random_state=0, n_estimators=100)
    run_model(BASE_NAME+"AdaBoostRegressor.pkcls", regressor,  X_train, y_train, X_test, y_test, "AdaBoost")
```

Source: Prepared by the author (2023)

Figure 13-B - Saving results.

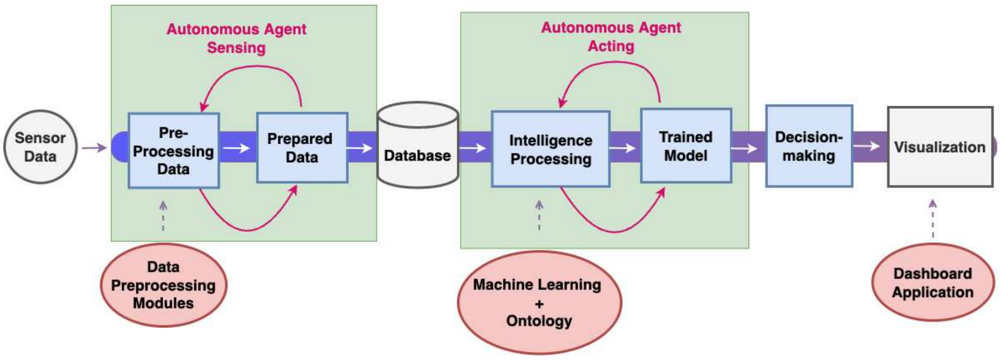
```python
def run_model(name, model, X_train, y_train, X_test, y_test, algorithmLabel):
    try:
        regressor = model.fit(X_train, y_train)
        save_model("./models/"+name, regressor)
        y_pred = regressor.predict(X_test)
        mean, mse, rmse, r2 = getErro(y_test, y_pred)

        postObj = {
            "name": algorithmLabel,
            "fileName": name,
            "desc": "Predict the animal's milk production based on the weigth",
            "type": "peso",
            "mse": mse,
            "rmse": rmse,
            "mae": mean,
            "r2": r2,
        }
        saveData(postObj)

        return y_pred, mean, mse, rmse, r2
    except NameError:
        print(NameError)
```

Source: Prepared by the author (2023)

parameters were used for each training. The training process starts by testing the entire dataset for each available algorithm on the list. Metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 are recorded for each algorithm, along with the algorithm name, prediction type (goal), a brief description, and the path to the trained algorithm (which will be utilized later by the application). A code snippet in Python for training an algorithm for predicting milk production based on animal weight is presented in Figure 13-A, and the Python function encoded to send the statistical results of each trained algorithm to the API is shown in Figure 13-B.

The algorithm with the lowest MAE is selected at the end of this process and used by the application, which in the context of this study is a mobile application. However, the algorithm remains available for use by any other application that consumes the architecture's API. This satisfies the extensibility (NFR002) and scalability (NFR004) requirements. Figure 14 depicts the data processing stages, from the entry into the architecture to the selection of the algorithm to be utilized by the application.

Figure 14 - e-Livestock Workflow.



Source: Prepared by the author (2023)

4.7.3 e-Livestock In Action

An example of the data used during testing and training for a milk production prediction model can be seen in Figure 15. We used weight (kg), the month in numerical form, and the milk produced (L). Using this dataset containing milk production for a whole year, we can train several ML algorithms to identify the most appropriate approach, the one that provides the best prediction.

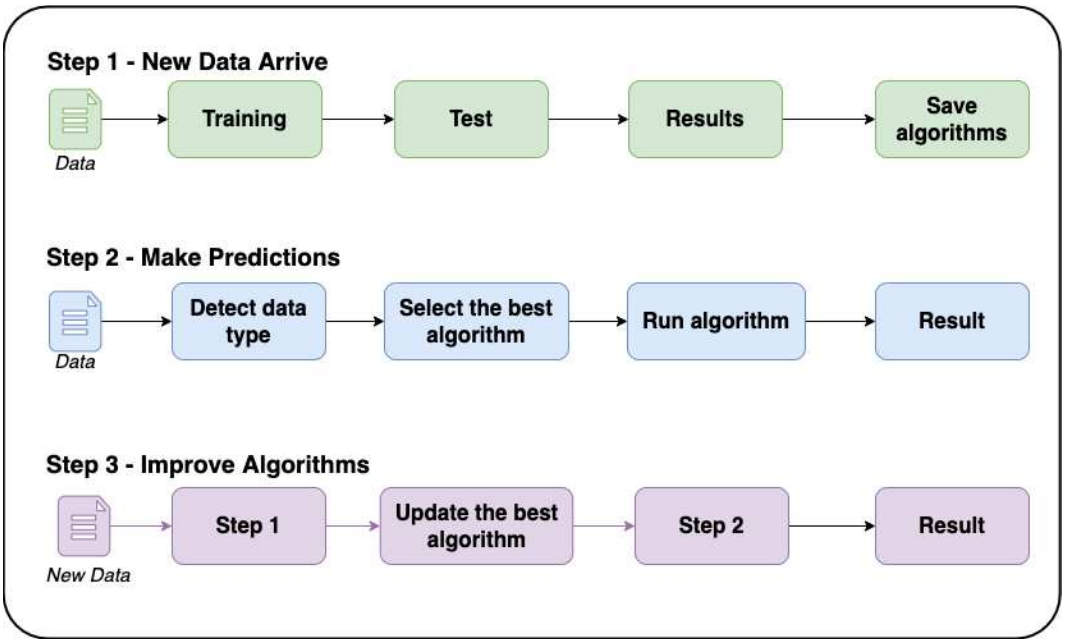
Figure 15 - Training data for milk prediction.

Animal_ID	Weight	Milk_production	Month
1428	650	34.95	2
2414	653	21.27	2
2592	493	25.3	2
3122	639	40.55	2
3142	639	23.5	2
3148	605	46.33	2

Source: Prepared by the author (2023)

Intelligent processing can be divided into three main stages. The first step involves receiving a set of data for training purposes to test various intelligent algorithms and record the accuracy of each. In the second step, the trained algorithms are ready to be used, and the architecture's API automatically selects the most suitable one. The third step aims to promote the sustainability of the architecture by receiving updated data to be processed. Then, the first and second steps are repeated: training, testing, and defining the algorithm with the best results. Figure 16 depicts the flow of steps 1, 2, and 3.

Figure 16 - Training, test and result.



Source: Prepared by the author (2023)

The training focused on predicting milk production on the farm, as it is the most important asset for dairy farms. Figure 17 shows the training visualization for milk prediction based on animal weight. After training, we compared the algorithm's predictions with milk production data to determine the highest accuracy. Figure 17 presents an example of the results for July, displaying the algorithm's outcomes, data, and metadata, such as the average error for each animal.

Given the results, we can store the training metadata, save the model with the best result, and use it in the architecture. As the Intelligence tier communicates with eLivestock via API, sends the input data, and the algorithm will respond with predictive data. By storing the metadata of training and testing of intelligent models, such as information on average errors, model type, and input data, it is possible to analyze and discover the best predictive model for a given data set. Using smart models, producers can have predictions and investigate animals that are not producing as expected, improving the animal's quality of life, food, and health. Hence, the producer can prevent diseases and ensure animal

Figure 17 - Test Results – Milk x Weight.

	Neural Network	Random Forest	Prod.Leite	Brinco	Peso	date
1	23.40	32.48	39.87	1069	764	7
2	18.42	37.02	43.40	1428	661	7
3	16.30	35.83	33.73	2592	569	7
4	18.11	26.73	43.77	3030	653	7
5	18.30	33.64	31.17	3038	658	7
6	18.49	25.05	50.30	3122	663	7
7	17.26	38.19	41.80	3148	629	7
8	20.30	42.08	40.33	3653	705	7
9	16.43	33.23	32.37	3785	594	7
10	19.41	35.29	39.20	3789	685	7
11	21.33	47.00	39.17	4076	727	7
12	23.15	31.50	28.77	4111	760	7
13	23.15	31.50	38.40	4117	760	7
14	21.43	41.94	30.73	4123	729	7
15	21.97	29.67	42.73	4136	739	7
16	17.96	34.71	32.23	4158	649	7
17	22.57	34.29	33.20	5078	750	7
18	19.81	30.17	34.37	5108	694	7
19	17.50	29.16	49.90	5109	636	7

Model	MSE	RMSE	MAE	R2
Neural Network	379.863	19.490	17.503	-3.960
Random Forest	98.621	9.931	7.945	-0.288

Source: Prepared by the author (2023)

welfare. A new dataset can also be added through the external services tier to help in the predictions. Once the smart model is trained and is ready to use, it can request input data from the API and send the results generated by the model back to the API. The metadata used by the model is captured during this communication between the ML module and the API. The next chapter will detail each phase, describing a historical study conducted on a specific experimental farm.

4.7.4 Ontology and Machine Learning Altogether

Previously, we discussed how ontology could be used to extract semantic knowledge from data, perform inferences and derive relationships between entities. We also explained the ML module and how training and prediction are conducted within the architecture. While these approaches can contribute independently, one of the advantages of the architecture is the ability to combine both to offer the complete decision support possible, given the available data.

Thus, we can extract the data inferred by the ontology model and use SWRL rules to filter and select a more enriched dataset for ML training. In addition, we can use ontology to select the best input data to provide consistent predictions that reflect results closer to the reality of the producer. For instance, if milk production is calculated by adding up all milkings for the month minus the milk discarded from sick animals, we can use ontology to extract only healthy animal input data and make a prediction of total milk production, considering the sick animals that didn't contribute to the final sum.

Another advantage of using ontology with ML is to derive data from semantic relationships. This data can identify later events related to the farm environment. For example, in case temperature and humidity sensors fail for various reasons, either due to mechanical or electrical failure or even being damaged by the animals, we can derive missing data by relying on the relation between the external temperature we can collect through the INMET API, and the temperature data collected by the sensor inside the farm. This complete dataset can be used to train intelligent algorithms that predict the weather conditions of that specific farm.

To demonstrate that the architecture can execute both ontology and ML approaches, we have developed a feature capable of predicting the farm's milk production for the month based on the animals' weights. First, we remove mastitis-contaminated animals from the herd. To achieve that, we use SWRL rules to define sick animals, super producers, and "typical"animals, which are not sick and have an average production below 30L of milk. Then, we separate the weight data of each animal and send it to the ML module to predict each animal's milk production. Finally, we summarize all production to reveal the expected liters of milk for the month to the producer. Figure 18 presents the code snippet of the main function that extracts data from ontology to send to the ML algorithm.

Figure 18 - Prediction with ontology data.

```
@app.route('/api/predict/animal/lote/leite', methods=['POST'])
def api_animal_lote_leite():
    onto = get_ontology("./new_provcow.owl").load(reload_if_newer = True)
    data = ontology.getAnimalsPeso(onto)
    listResult = []

    for peso in data:
        req = [{'Peso': peso}]
        df = pd.DataFrame(req)
        result = predict.predict_milk(df)
        result = {"input": peso, "result": result[0]}
        listResult.append(result)

    return jsonify(listResult)
```

Source: Prepared by the author (2023)

4.8 FINAL REMARKS OF CHAPTER

This chapter presented the architectural project, its requirements, Design Science cycles (DSR), and detailed architectural design. The Intelligence tier of the architecture was described, including the use of ontologies and machine learning techniques.

This Intelligence tier was developed in the second cycle of DSR, where we enhanced the architecture by developing the intelligence capability and conducting a second evaluation. We sought to improve intelligence-based decisions by supporting predictions and semantic relationships. Questions that cannot be answered with ML alone were answered through inference algorithms that the ontology allows. ML provides a predictive view of milk production, while ontologies provide a retrospective view of production data. Producers can make more informed decisions by combining these approaches into a single one, presenting both predictive and retrospective views with graphical support, producers can make more informed decisions.

Machine learning techniques are effective when large datasets are available for training, and ontology helps the farmer by filling in missing data. Another example of combining ontology with ML techniques would be detecting a possible mastitis epidemic. By classifying sick animals through ontology and querying them from the database, we can use ML to predict the number of potentially sick animals per batch. Hence, combining both intelligence approaches provide more accurate predictions.

Next, we will present a historical study using real farm data to evaluate the architecture. The results will also be presented and discussed.

5 EVALUATION OF ARCHITECTURE E-LIVESTOCK

This chapter describes the evaluation of the e-Livestock architecture. As stated before, this study was based on the Design Science Research (DSR) methodology. In the study, we instantiate the e-Livestock architecture in an experimental farm, in partnership with Embrapa – Gado de Leite, in Coronel Pacheco, MG. The design cycle (build/evaluate) is the heart of every Design Science research project and where the most intensive work on DSR is done (Hevener, 2007). The entire research process must be described, and rigorous methods must be applied in constructing and evaluating the artifact. The rigor of the research is derived from the effective use of the knowledge base, which is the set of fundamentals and methodologies necessary for carrying out the research. In each cycle of the case studies, the execution of the evaluation generated scientific knowledge. This knowledge helped in the construction of new versions of the services that compose the e-Livestock architecture. The construction of the artifact is done through an iterative process. At each iteration, it was evaluated whether the artifact matched the requirements and whether it solved the problem. Furthermore, the artifact was refined to obtain more accurate results. It is a typical procedure adopted in design solutions in which the design activity variates between conceptual and practical activity.

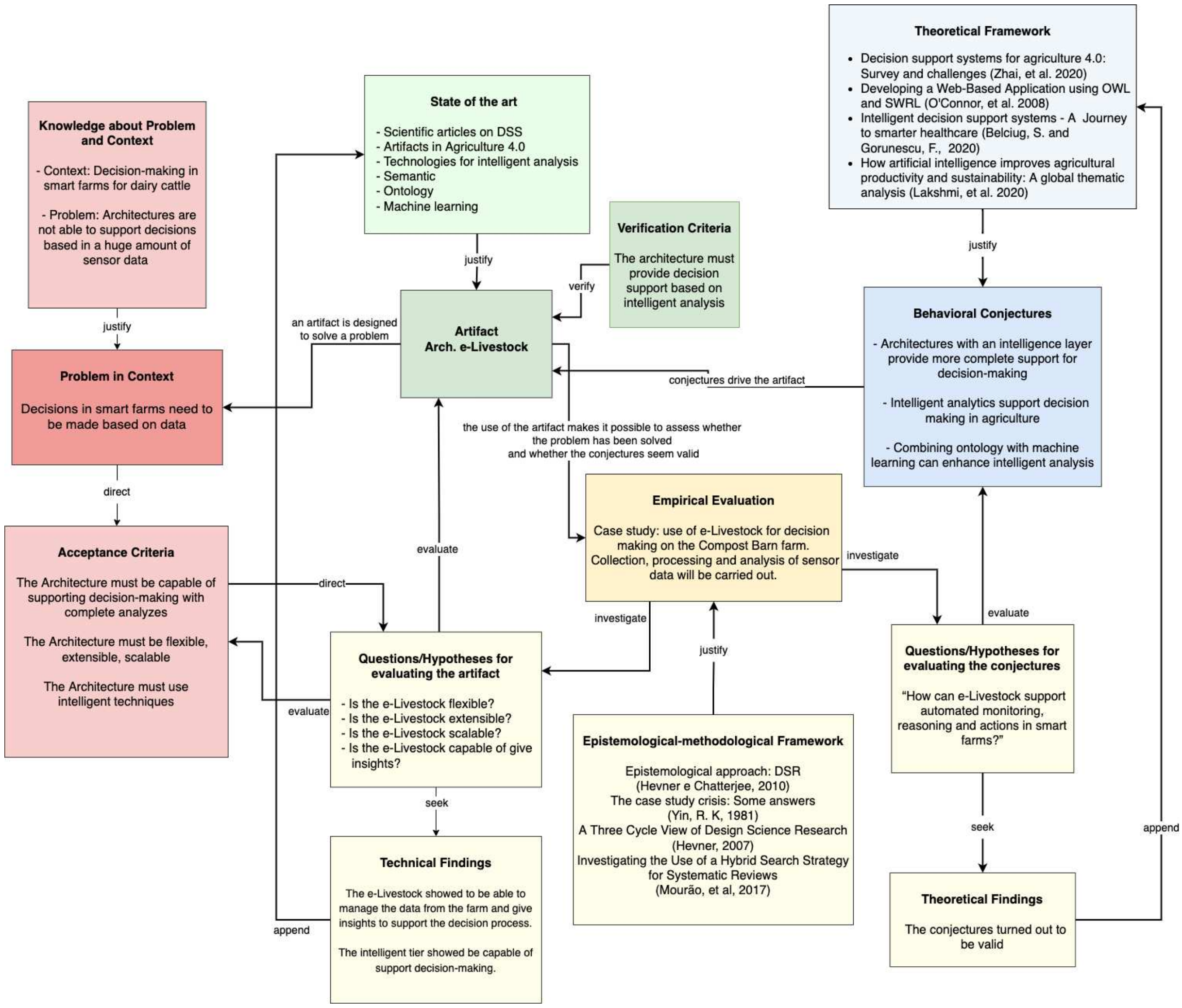
Finally, we analyzed whether the architecture supports decision-making on the farm through the results obtained. To conduct the study, researchers need to collect the data generated by sensors on the experimental farm, to monitor production data and analyze how this data can support decision-making regarding animal production. Therefore, we conducted historical research using the data collected in 2020 and 2021.

5.1 METHOD

The goal of our research was to analyze the support to monitor the environment, reason on data, and automate actions from the researcher's/farmers' point of view, in the context of a smart farm system. From the scope, we defined the following research question RQ: "How can e-Livestock support automated monitoring reasoning and actions in smart farms?".

As such, we claim that to be reliable, an instantiation of the e-Livestock architecture should result in some insights into the environmental conditions under which the e-Livestock operates, considering both its surrounding environment (such as external and internal temperature) and animal's health conditions (such as mastitis and milk production). According to the DSRM (Design Science Research Methodology), we must follow: Definition of the problem; Literature Review and Search for existing theories; Suggestions of possible solutions; Development; Evaluation; Decision on the best solution; Reflection and Learning; and Communication of results. Considering the DSR methodology, we adapted Figure 19 to illustrate the evaluation elements in DSR for this work.

Figure 19 - e-Livestock Architecture Assessment DSR Element Map.



Source: Pimentel, M et al. (2020)
Nota: Adapted Figure.

Based on the theory found in the literature mapping (theoretical framework), we developed and evaluated an artifact capable of supporting decision-making. The evaluations aimed to verify the following: i) if the artifact works; ii) if the theoretical conjectures align with expectations; and iii) if the artifact (architecture) helps in decision-making on the farm. Based on the evaluation results and the scientific knowledge acquired, we confronted the theoretical conjectures raised during the literature review. We perform one more interaction, improve the artifact and re-evaluate it. Finally, we assess whether the architecture answers the research question (RQ) to improve decision-making on the farm, considering more complex analyzes supported by machine learning and ontology. To communicate results, we publish some articles about the cycles through conferences and workshops related to computer science and agriculture (Gomes et al. (2021); Gomes et al. (2021); Gomes et al. (2022)).

Figure 19 shows the e-Livestock Architecture Assessment DSR Element Map. The theoretical approach of the research (elements on the right side of Figure 19), the artifact (elements on the left side), and the application context (elements at the top of Figure 19) are also shown. This separation highlights the correlation between technological-applied development and scientific-theoretical knowledge.

According to the Theoretical Framework presented in Figure 19, the architectures presented in works found in the literature need to be prepared to deal with a complex domain whose context can change quickly. In precision livestock, new sensors can be installed on the farm; new animals can arrive in the herd, and weather conditions can change throughout the year. Even so, the systems must continue functioning correctly, supporting producers in decision-making. Complex decisions, such as mastitis, are usually costly for the farm; effective control and prevention are necessary to represent a drop in milk production and even the loss of animals (animals diagnosed with chronic mastitis are usually discarded from the herd). So, the architecture must allow the construction of a decision support system for smart farms, considering the adversities of the domain.

We implemented two cycles. In the first cycle, we built the e-Livestock architecture to monitor the Compost Barn environment and performed a case study to evaluate the architecture. In the second cycle, we developed an Intelligence tier and conducted the second case study to answer the research question. The case studies refer to a scenario where the solution provided decision support and analyzed data from various farm sectors related to dairy cattle. This scenario assumes that milk production is the focus of the data. Healthy animals that eat properly have adequate weight and have no disease making them more efficient in increasing dairy production.

5.2 EVALUATION SCENARIO

A case study was developed to evaluate using the e-LiveStock architecture in a real-world context (Yin, 2015). The case study was conducted according to the following steps (Runeson and Martin, 2009): (i) case study design (preparation and planning for data collection), (ii) execution (collection of evidence), (iii) analysis of collected data, and (iv) reporting. Our case study scenario consists of data monitoring, collecting, and processing and then analyzing the data from a production system called Compost Barn, located at Embrapa – Coronel Pacheco, Brazil. We used data from sensors (6 different types) collected by researchers from Embrapa between 2020 and 2021.

The environment includes a covered and ample physical space for the cows to rest. The area is lined with sawdust, scrap wood, and composted manure. This space has sensors to monitor the temperature and humidity of the environment. The data is available on GitHub[7]

Figure 20-A - Animal with a neck sensor at Compost Barn.

Source: Prepared by the author (2023)

Figures 20-A and 20-B show the interior of the Compost Barn in the experimental farm at Coronel Pacheco, where we can observe the animals and the sensors used. Continuous monitoring allows for adjustments in the animals' living conditions and increases animal welfare.

5.3 CASE STUDY 1 (FIRST DSR CYCLE)

After analyzing the theoretical conjectures from the literature mapping, we built a layered architecture. Therefore, we developed an External Services and Integration tiers to communicate with the architecture. The External Services tier is related to the architecture's extensibility since the architecture can extend and consume data from any external services that communicate via the HTTP protocol. We developed these tiers to meet the functional requirements FR001 and FR002, where the architecture must be capable of processing different Compost Barn datasets and integrating external sources' information, such as weather services and geolocation, of aiding in farm data enrichment. Requirements FR004 and FR005 refer to the storage and visualization of Compost Barn data, respectively.

For the non-functional requirements, we have NFR001 and NFR002, where the solution must allow communication with external data sources and meet the extensibility attribute. This way, the architecture can accommodate the system's future growth, integrating with other services, adding more sensors and new data sources, and enabling different visualization types on various devices. Regarding NFR003 and NFR004 requirements, the architecture must respect flexibility and scalability principles, respectively. Flexibility is an attribute that reflects the range of behaviors that the existing architecture

[7] https://github.com/jjthegomes/elivestock

Figure 20-B - Sensors at Compost Barn.

Source: Prepared by the author (2023)

can be configured to meet, such as processing animal health data, milk production, and environmental data. Scalability considers the system's adaptation to new size and scope specifications, allowing for the addition of new data sources.

After instantiating the architecture, we prepared the data, collected, analyzed, and reported case study 1. Each step is presented below, and the results are discussed at the end.

5.3.1 Stages of Case Study 1

We conducted Case Study 1 (CS1) in three stages. Firstly, data preparation was done through pre-processing, where we formatted and removed uncollected or incomplete data. Secondly, we imported and stored the data so that it could be made available through a dashboard. Finally, we constructed a dashboard through which users could visualize the farm data in graphs and evaluate if the system would assist in decision-making. The first stage was crucial as it involved working with raw sensor data. For this case study, the datasets entered the system by reading spreadsheets, and we applied data treatment for each row of the files. This parsing process takes place at the sensor tier. Various methods handle preparing and transforming the initial data set (Vassiliadis et al., 2002). The architecture used the Extraction, Transformation, and Loading methods for this case study since we had multiple data sources and needed to transform the data into a suitable format for analysis.

In the second stage, we stored the processed data in the database. Working with spreadsheets, we performed all pre-processing in memory, and at the end of processing each file, we sent the data set to the API via HTTP request. Data integration occurs when it is necessary to aggregate data from many sources, such as external temperature data from the farm. Therefore, for this case study, we used the INMET API to provide meteorological data for the city of Coronel Pacheco, where the farm for this study is located. The INMET API provides data on maximum and minimum temperature, maximum and minimum humidity, wind speed and direction, the current season, and other data. Although any API that provides meteorological data could have been used, we opted for INMET because there is a meteorological station in the city where the farm for this case study is located. Initially, we chose OpenWeather, but the accuracy of INMET's results presented more consistent and precise data. However, the architecture could handle multiple data sources, allowing comparison and cross-referencing of data from different APIs. Figure 21 presents internal data (from sensors) and external data (from the INMET API) for both temperature and humidity on January 1st, 2022.

Once the data had been organized and stored in the database, we used the visualization tier to display farm data in dynamic graphs. Furthermore, we created an interactive dashboard to present the data at various granularities. For this first cycle, we employed ThingsBoard[8] as a visualization tool. ThingsBoard is an open-source server platform that enables the monitoring and control of IoT devices. It is free for personal and commercial use and can be deployed on any computer. ThingsBoard offers an IoT solution that is ready-to-use in its server infrastructure. The advantage of ThingsBoard is that it supports MQTT, CoAP, HTTP, and LwM2M protocols. Additionally, the ThingsBoard platform is horizontally scalable. Each server node in the cluster is unique, and scalability is achieved through a consistent hashing load-balancing algorithm among the nodes in the cluster. The actual performance is contingent upon the device usage scenario.

We collected farm data, processed, stored, and visually presented it. We verified whether the e-Livestock artifact dealt with the problem of assisting decision-making on

[8] https://thingsboard.io/docs/

Figure 21 - Example of Indoor and Outdoor Temperature and Humidity Data.


```json
[{
  "external_measure_value": 22.70,
  "internal_measure_value": 20.70,
  "measure_date": "01/01/2022",
  "type": "temperature",
  "date": "2022-01-01T03:00:00.000Z"
},
{
  "external_measure_value": 79.83,
  "internal_measure_value": 92.83,
  "measure_date": "01/01/2022",
  "type": "humidity",
  "date": "022-01-01T03:00:00.000Z"
}]
```


Source: Prepared by the author (2023)

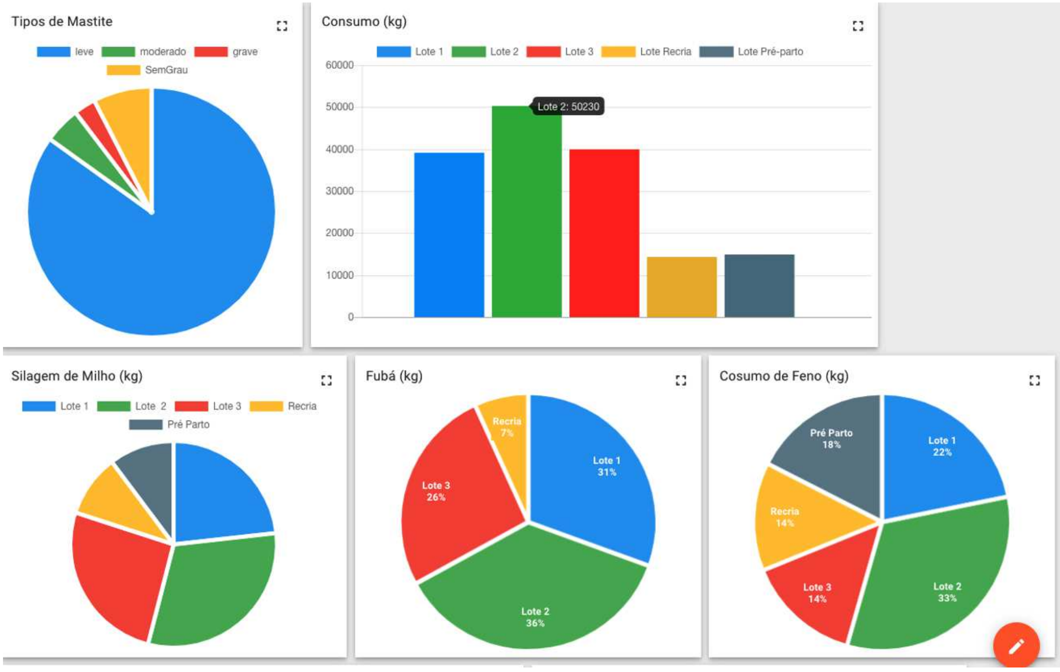
the farm. For this purpose, it was necessary to analyze the information available on the dashboard from the farmers' perspective.

5.3.2 Conducting Case Study 1

To evaluate the first DSR cycle, we created a dashboard containing different graphs representing the farm's processed data, such as feed consumption, mastitis incidence, and total consumption (kg) per batch of animals (herd). Figure 22 presents a screenshot of the dashboard on ThingsBoard, through which it is possible to see in the first pie chart the different degrees of mastitis and the percentage of animals affected concerning all animals on the farm. By positioning the mouse over each part of the graph, it is possible to visualize the precise values of the graph. This analysis allows the producer to visualize the animals' health in general. With that information, we devise an action plan to reduce mastitis and track the effectiveness of the plan as animals recover and fewer animals fall ill. The opposite would also be valid since noticing that many animals are sick could indicate a mastitis epidemic and thus allow producers to act before the situation gets worse.

On the right in Figure 22, it is possible to see the bar graph showing each batch's total food consumption through the different colors. It was observed that batch 2 had a much higher consumption than batch 1 and 3. Something unusual, as batch 3 contains the animals that produce more milk, leads to the conclusion that animals from batch 2 may be receiving an inappropriate diet. However, the rearing and pre-calving batches had a lower consumption; something expected considering that they are young animals. At the bottom of the graph, we present some of the components in the animals' diet, such as the consumption of Soy, Cornmeal, and Hay, in pie charts, in which we can observe the consumption categorized by batch. Each color represents a batch; the percentage refers to the farm's total consumption. In this case, we can see that the pre-calving batch does not consume cornmeal. However, it consumes more hay than the rearing batch, for example. This fact is because prenatal care animals have a different diet than others.

Figure 22 - ThingsBoard with Evaluation Data – Mastitis and Diet.

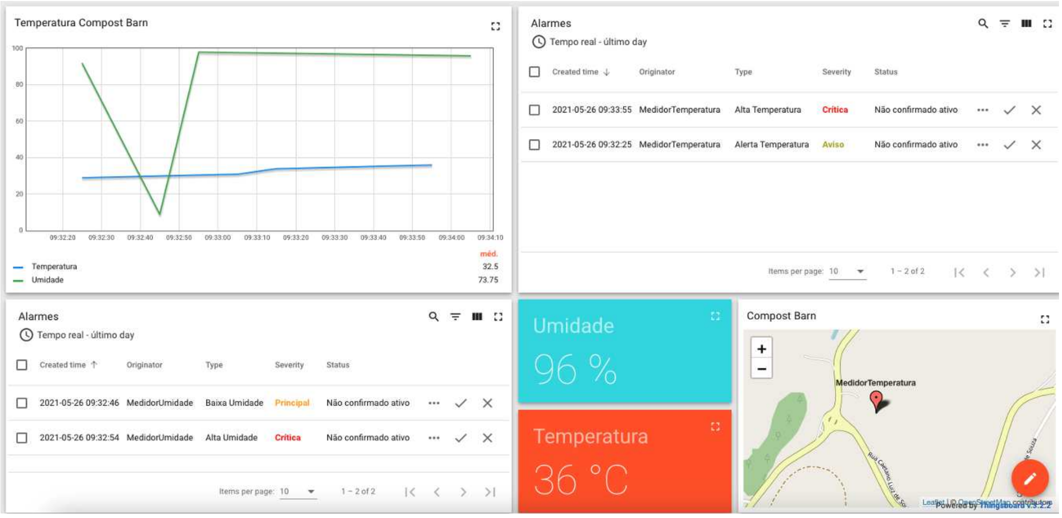


Source: Prepared by the author (2023)

In addition to displaying food consumption and mastitis data, we also created a dashboard to display data on the environmental conditions of the Compost Barn. Figure 23 presents a line graph showing the indoor temperature (blue line) and humidity (green line). It's possible to view two cards that show the outside temperature (orange) and humidity (light blue). To the right of the cards, we have a map showing the geolocation of the temperature sensors on the farm. By visualizing the sudden drop in humidity on the graph, producers and researchers could intervene in the environment and investigate in real-time what was happening. Upon realizing it was a sensor failure, adjusting and repositioning the sensor in the correct position was quickly possible, causing it to measure humidity correctly again. Thanks to the dashboard and sensor monitoring, actions could be taken to improve the environment quickly and efficiently, generating more animal comfort and, consequently, a better production result.

In addition to charts, we also set up alarm rules in ThingsBoard. For example, suppose the temperature reaches a high value, such as 34 degrees Celsius. In that case, we can trigger an alarm via email and SMS so the producer can immediately intervene in the environment physically. The same goes for humidity. A high humidity level can indicate that the shaving bedding, the organic material on which the animals lie, needs maintenance. This worry arises because high humidity levels can favor the proliferation of environmental bacteria that cause mastitis. Monitoring temperature and humidity are essential to maintain animal welfare, as a lack of control in the environment can cause discomfort, stress, and, consequently, a drop in milk production.

Figure 23 - ThingsBoard with Evaluation Data – Temperature and Humidity.



Source: Prepared by the author (2023)

Since the Compost Barn production system has internal measuring equipment installed in the building and exhaust fans to control the temperature, we can support the decision to turn on/turn off this equipment based on rules to analyze the environment. The ideal environmental condition is that the internal temperature is 5 degrees less than the external temperature of the environment. We can turn on more hoods as the temperature increases to cool the environment. If a temperature exceeds the limit of 34º C, the system can communicate through the Integration tier with external services and trigger an audible alarm. Figure 24 partially presents a dataset of internal temperature from the environment, captured by the farms' sensors and used in this evaluation. The colors indicate a heat map: colder blue, normal green, medium yellow, and high orange.

Maintaining a cool environment for dairy cattle is important for several reasons. One reason is that cows are more comfortable and less stressed when kept in a comfortable environment, which can improve their overall health and well-being. A cool environment can also help reduce the risk of heat stress, which can occur when cows are exposed to high temperatures and humidity. Heat stress can lead to several negative effects on cows,

Figure 24 - Internal temperature data.

	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
DAY	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00:00	17,2	17,2	17,1	16,1	15,0	15,7	16,0	16,7	20,1	15,2	17,8	18,2	21,3	17,2	16,2	12,7	16,6	12,7	15,2	17,2	20,3
01:00	17,8	17,0	17,0	16,6	15,0	15,4	15,8	16,5	20,0	15,1	18,1	18,3	21,3	17,9	16,7	12,8	16,3	12,4	15,1	16,5	20,5
02:00	18,2	16,6	17,1	16,0	14,7	15,1	15,5	16,7	19,6	14,4	18,2	17,9	20,8	16,6	16,8	12,6	16,3	12,3	14,4	16,0	20,4
03:00	17,5	16,8	17,2	15,8	14,2	15,2	15,1	16,5	19,8	14,3	18,6	17,6	20,6	15,7	17,0	12,4	15,7	12,2	14,1	16,1	20,2
04:00	17,4	16,2	17,4	15,7	12,9	14,5	14,5	16,4	19,8	12,9	18,7	17,4	20,5	15,3	17,1	12,4	14,4	12,1	12,9	16,0	19,6
05:00	17,2	16,3	17,2	15,2	12,9	14,2	14,5	15,7	20,0	12,7	19,0	17,6	20,3	14,9	17,3	12,3	12,7	11,8	12,9	16,0	19,1
06:00	16,3	16,5	17,0	14,7	12,7	14,3	14,8	16,5	19,7	12,4	19,2	17,9	18,5	15,8	18,7	12,4	12,6	12,0	12,5	15,7	19,8
07:00	15,9	16,7	17,5	15,7	15,5	15,6	14,5	16,1	18,9	14,1	18,7	19,2	19,0	15,1	17,8	12,5	14,3	12,6	13,0	16,6	19,0
08:00	18,8	18,7	19,9	18,0	17,1	17,5	17,7	18,7	20,6	16,4	19,3	20,0	18,5	16,6	18,6	15,8	16,2	14,9	17,2	19,0	18,8
09:00	20,4	21,1	21,6	21,1	20,2	20,6	20,9	20,8	20,3	19,1	20,8	21,1	20,1	19,3	20,0	19,4	19,7	19,0	20,7	20,8	19,2
10:00	21,2	21,7	21,3	20,3	21,2	21,7	21,9	22,1	20,9	19,5	20,6	22,7	20,3	20,6	19,6	20,1	20,4	20,6	21,8	22,1	20,0
11:00	21,8	23,0	23,0	22,1	22,5	22,8	23,1	23,3	21,0	20,6	21,5	23,3	19,5	21,1	20,6	20,4	21,4	21,6	22,3	23,0	20,7
12:00	22,0	23,6	22,6	21,9	22,4	23,0	23,7	24,1	20,6	21,2	22,4	24,0	20,0	21,1	20,7	20,7	21,7	22,1	22,9	23,5	20,4
13:00	22,1	23,6	22,4	22,6	22,3	23,2	24,0	24,6	21,2	20,7	22,9	24,1	20,2	21,0	20,2	21,0	22,0	22,7	23,5	23,8	20,7
14:00	22,8	23,5	22,6	22,6	22,6	23,7	24,2	24,6	20,6	21,0	22,8	24,0	19,7	21,5	21,7	20,6	22,0	22,8	23,2	24,4	20,5
15:00	22,6	23,9	23,4	22,7	22,8	23,7	24,4	24,4	21,1	20,8	23,0	24,6	19,8	21,7	20,7	21,1	21,9	23,1	23,1	24,2	20,7
16:00	22,8	23,8	22,3	22,5	22,7	22,7	23,7	23,4	20,5	21,1	23,0	24,7	19,8	20,8	20,6	20,9	21,7	22,8	23,1	24,0	20,0
17:00	21,4	21,5	21,5	20,5	20,9	21,4	21,9	22,4	19,6	19,8	22,3	23,2	20,3	19,7	17,7	19,2	19,3	20,3	21,6	22,4	19,7
18:00	19,9	20,3	20,2	18,5	19,1	19,3	20,6	20,4	18,8	19,5	21,2	22,3	18,5	17,9	16,7	17,9	17,6	19,1	20,3	21,5	20,2
19:00	19,2	19,2	19,4	17,9	18,6	18,7	19,7	19,8	17,7	18,0	20,3	22,3	18,5	18,3	15,7	16,9	16,6	17,9	19,7	21,3	19,7
20:00	18,5	18,5	18,4	17,0	17,4	17,7	18,8	19,2	17,1	17,6	19,9	21,9	17,9	17,0	15,1	16,4	15,7	17,2	19,1	21,0	19,7
21:00	17,7	18,0	17,6	16,5	17,2	17,4	18,3	19,0	16,4	18,1	19,5	21,6	18,6	16,1	14,5	16,6	15,2	16,9	18,3	20,8	19,5
22:00	17,6	17,8	17,0	15,9	16,9	17,0	17,6	20,8	15,8	18,5	18,7	21,7	17,9	15,7	14,1	16,8	14,5	16,2	17,8	20,5	19,2
23:00	17,2	17,5	16,7	15,5	16,0	16,6	17,4	20,0	15,3	18,4	18,5	21,5	17,2	16,1	12,9	17,1	12,9	15,8	17,5	20,4	19,2

Source: Prepared by the author (2023)

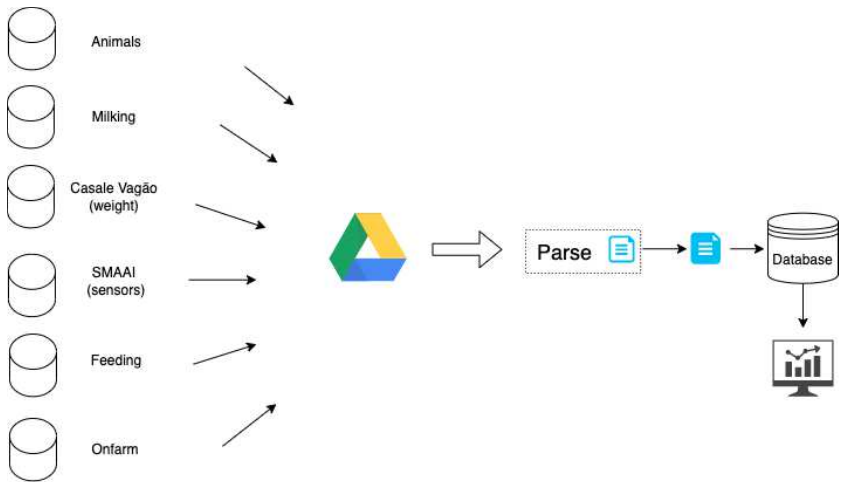
such as reduced milk production, and increased disease risk. Cows produce more milk when comfortable and not stressed; a cool environment can help keep cows calm and relaxed. Overall, maintaining the environment for dairy cattle is important to their health and well-being and can help optimize milk production and improve farm efficiency and profitability.

5.3.3 Analyzing Case Study 1

Through case study 1, we verified whether the architecture supports decision-making in the smart farm based on data monitoring. We could also check if the theoretical conjectures and requirements were aligned with the solution (artifact). Figure 25 presents an overview of this first DSR cycle. According to Figure 25, observing the six data sources used in the case study is possible. The animal dataset contains information identifying the animal, its batch, and its birthdate. Dairy control data includes milking carried out throughout each month. The scale sensors that weigh the animals collect the data from “Casale Vagão”. The indoor environment data is collected from the SMAAI sensors (temperature and humidity). Feed data contains the amount in kilograms of each component of the animals’ diet, such as corn, soybeans, cotton, and hay. Animal health data were made available from an internal Embrapa’s system, whose records indicate the incidence of mastitis, the type of bacteria, the severity of the disease, and the medication used.

By integrating different datasets and making them available for visualization, we verified that the architecture met the extensibility, flexibility, and scalability requirements. Because as new data arrived in the architecture, it could process different types of data

Figure 25 - First Cycle DSR Overview.



Source: Prepared by the author (2023)

and store and make data available to users with satisfactory performance. In addition, we verified that adding different external services to complement the decision-making process is possible.

A system can be considered extensible if it can be easily adapted to meet the constantly changing needs or requirements and can be modified without significant redesign or rework (Amorim et al., 2013). In this regard, the architecture has demonstrated its ability to easily extend or modify to add new features, services, and datasets without major changes. A flexible system can handle various inputs and be used in various ways without requiring significant modifications (da Silva Amorim et al., 2014). Regarding flexibility, the architecture can be used in multiple contexts, such as feeding, animal health, milk production, and environmental conditions. Finally, regarding scalability, the system has handled increasing amounts of workload or data without experiencing a drop in performance. A scalable system can maintain its performance as the size or complexity of the workload or data increases (da Silva Amorim et al., 2014).

The sensor data integrated with environmental context data are presented to the farmers through graphs and alert notifications, helping in the decision-making. Also, they have an overview of the environment being monitored. Through the dashboard interface, triggering alarms based on rules was possible. For example, device "A" performs a temperature reading of 34º C that exceeds the defined limit. As a result, a "High temperature" alert is generated. Each alarm's severity can be defined as Critical, Main, Secondary, Warning, or Indeterminate (ranked by priority in descending order). Users could also receive alert notifications via SMS and email. We used the INMET website to collect the external temperature/humidity at the Compost Barn. By analyzing context data, such as external temperature and climate forecasts, farmers could make more precise adjustments, automating the process of starting exhaust fans. As a result, it was possible to avoid sudden changes affecting the animals' production.

Therefore, the e-Livestock architecture could process Compost Barn data adequately. However, the results cannot be generalized, and additional evaluations can provide more information on e-Livestock suitability to support decisions on smart farms.

5.3.4 Results of Case Study 1

At the end of this first cycle, it was possible to analyze the production results of the farm, considering milk production, the amount of food ingested by the animals, the total number of sick animals over time, and data from environmental conditions (temperature and humidity). With the dashboard shown in Figures 22 e 23, producers could access information to assist in the decisions, such as choosing appropriate nutrients for a given batch. Analyzing the graphs, it was also possible to verify how many mastitis cases were registered in the month and if the disease increased or decreased. It is possible to monitor the environmental conditions of the Compost Barn by analyzing the temperature and humidity collected through the sensors. With those, we open the opportunity to analyze important variables and evaluate the efficiency of the Compost Barn system, such as consumption, production, disease, and wellness. Herein, it is important to cross-reference the information obtained from the sensors to assist in planning future actions.

At the end of the Case Study 1 (CS1), we found that the use of a dashboard presenting data from the farm is not enough to support decisions, in such a way that only the analysis of graphs is not enough to draw projections and make predictions for future farm´s production planning. While charts can visually represent farm data and highlight trends and patterns, it is also important to consider other factors affecting decision-making.

During CS1, our dashboard could display data on milk production, feed consumption, and animal health, which could consider the farm's production objectives and goals. Still, it could not plan feed consumption to adequately prepare agricultural commodities used in the animal's diet (such as corn and cotton, for example). To make decisions in a smart farm, it is usually necessary to consider a wide range of data and information from various sources and use applications that use intelligent techniques such as ML and reasoning algorithms to analyze and interpret the data. We also identified that temperature and humidity sensors sometimes fail, with the lack of energy being the main reason for data reading failure. Consequently, the chart presents gaps and incomplete data useless for decision-making. Additionally, adding an Intelligence tier would require a new quality requirement related to the sustainability of the architecture (NFR005). The sustainability of the architecture must ensure that the system can continue to operate and meet the producers' needs over time so that intelligent algorithms can perform their function accurately. To meet this requirement, the design of the new tier into the architecture should be easy to maintain and update.

In accordance with the Design Science Research (DSR) framework, upon scrutinizing the initial theoretical propositions, it was discovered that the architectural tier is appropriate. Nonetheless, there is a need to address the issue of decision support within the confines of precision dairy farming. At the conclusion of the primary DSR iteration, a dashboard was developed as an instrumental initial step toward comprehending the data. However, it is equally imperative to consider other aspects and employ diverse applications and techniques to facilitate superior decision-making. Furthermore, a noteworthy finding during this cycle was that farmers tend to favor the use of mobile devices instead of a web dashboard, such as ThingsBoard. As a result, we have scheduled modifications to the Visualization tier in the next cycle, intending to provide farm data via a mobile app.

Based on the theoretical framework, we analyzed the papers from the systematic mapping (Section 3). Then, we raised new conjectures and performed a new cycle to verify whether using the artifact solves the decision-making problem in livestock and whether the conjectures are valid. Based on the knowledge acquired in this first cycle, we raised new conjectures regarding using intelligent techniques to improve decision-making on the farm. Smart techniques are varied, but recent results suggest that machine learning and semantic web (ontologies) use are promising.

Machine learning techniques could be used to improve the performance of an Internet of Things (IoT) system in a precision livestock operation. ML algorithms can be used to analyze data collected by sensors and other monitoring equipment and identify patterns and trends that may not be immediately apparent to humans. In this way, the system uses additional data to resolve uncertainties. This fact can help producers and researchers make more informed decisions about managing their animals and improving the farm's efficiency and sustainability. For example, ML can predict when an animal will likely get sick or identify animal behavior patterns that might indicate they are under stress. As a result, we can allow producers to take proactive steps to prevent problems before they occur and improve the overall well-being of their animals. It can also optimize food and water systems or identify opportunities to reduce waste and improve resource efficiency.

On the other hand, ontologies are formalized structures to organize and represent knowledge about a particular domain. They are often used in the context of artificial intelligence and semantic web technologies and can be used to help computers better understand and interpret data. In a precision livestock context, an ontology can be used to help organize and classify data collected by sensors and other monitoring equipment and provide a common language for describing and interacting with this data. For example, an ontology can be used to define the various types of data that are collected on a dairy farm (e.g., feed intake, water intake, behavior patterns, and health indicators) and specify the relationships between these data types. An ontology can also be used to help integrate data from different sources and systems and allow the development of more sophisticated machine learning algorithms that can analyze and interpret the data in more complex ways. Overall, using ontologies in precision livestock can improve the efficiency and effectiveness of these systems and enable more data-driven decision-making in farm actions.

Intelligent techniques can deliver strategic information that can help in future decision-making. The analyzes carried out by e-Livestock in the first evaluation cycle do not allow making these analyzes (predictions) about future farm decisions, such as a prediction of milk production, an estimate of animal food consumption, or relating cases of mastitis with the data of the environmental sensors. Thus, based on the identification of improvements needed and the scientific knowledge generated by the first cycle, a new tier (Intelligence Tier) was specified in the e-Livestock, to assist in deriving strategic information to support decisions. In addition, user interviews highlighted the necessity to develop a mobile application (Visualization Tier) in future architecture versions.

5.4 CASE STUDY 2 (SECOND DSR CYCLE)

Based on the scientific knowledge generated in the first cycle, we improved the artifact and made new propositions to evaluate it. The theoretical conjectures include intelligent analyses supporting machine learning modules and ontologies. To achieve this, we developed a tier for these analyses to handle uncertainties and maintain the previously mentioned quality attributes. However, new functional and non-functional requirements emerged, which we will present throughout this text. We also modified the web dashboard to suit the producers' necessities better and created a mobile application compatible with Android and iOS operating systems.

To build the new version of the artifact, it was necessary to split the development process into stages:

1. We must create an ontological model for dairy cattle and relate the available Compost Barn data.

2. Creating semantic rules to execute inferences over the ontology was necessary. After that, we need to add the machine learning functionality and use the ontology to help the machine learning module process.

3. Present the predictive data to the user.

The second interaction cycle started with the definition of the ontology model (Section 4.5.1) and the SWRL rules that can extract implicit knowledge from the data. Utilizing an ontology as a means of enhancing decision-making in agribusiness involves employing a standardized and structured method of representing and organizing knowledge pertaining to the domain. By doing so, pertinent information can be more readily accessed and utilized while also fostering a sense of consistency and accuracy within the knowledge base.

Ontology is important to the decision support system because it uses machine learning and other techniques to analyze data and make recommendations or predictions. Using an ontology to represent and organize knowledge can make it easier to train and evaluate the system and ensure it makes decisions based on a consistent and comprehensive domain understanding. For each step, it was possible to improve the knowledge that helped enhance the final version of the architecture. We will present below, in more detail, each stage of construction of the second cycle of DSR.

5.4.1 Stages of Case Study 2

To develop the ontology, we used Python language and the OWL2Ready library, which enables us to manipulate data within the ontology. Using the PyMongo library, we queried the data in the database (which was stored in the first cycle), instantiated the ontology, and stored it in an OWL file. This way, it would only be necessary to update the ontology as new data was added to the database.

Once the processed data was instantiated in the ontology, we developed SWRL rules. As explained earlier in Section 4.7.1, SWRL is a language for expressing rules in an ontology model. Rules in SWRL are written as if-then statements and can be used to represent knowledge about a particular domain or subject, such as dairy cattle management.

These are just a few examples of rules that can be expressed in an ontology model for dairy cattle using SWRL. The specific rules included in the model will depend on the needs and goals of the model, as well as the knowledge and experience of the model creators. Figures 10 and 11 show the SWRL rules.

With the ontology model and inferences completed, we proceeded to the development stage of the machine learning module. Developing such a module for dairy cattle is a complex task that requires expertise in machine learning techniques, software engineering, and dairy management. Training machine learning models to recognize patterns and relationships in large amounts of data is a computationally intensive process that demands sophisticated algorithms and software tools. To acquire the necessary knowledge and develop the module, we conducted technical interviews with professionals from Embrapa Gado de Leite, who provided us with technical data on animal management and dairy production.

To improve the architecture and provide information for decision-making, we combined data pre-processing, model training, evaluation, and optimization. Since machine learning models need regular maintenance and updates with new data, we created an auto-training function capable of updating trained models with new data. However, the input data must always have a consistent pattern; otherwise, slight adjustments to the data are necessary.

We used the Python language and the Scikit-learn library to train and execute intelligent models, as per the objectives defined, which were to forecast individual milk production as a function of animal weight and forecast general milk production on the farm as a function of diet. To achieve these objectives, we organized the data, performed filtering and standardization, and trained different models. We evaluated the accuracy of each model and selected the most accurate for each objective. Afterward, we tested the accuracy of the trained models by crossing the result with a set of real data.

Integrating the trained models with the ontology was done through an integration tier that played a vital role in the communication between the intelligence tier (ontology and machine learning) and other parts of the architecture (processing, database, visualization). We developed an API using Flask library to communicate the intelligence tier with the integration tier, enabling data from the integration tier to arrive through web requests, be processed by the intelligence tier, and integrated back into the architecture. The architecture allows low coupling, allowing for easy integration with other intelligent techniques in the future. For instance, a farm selling dairy products could use Natural Language Processing to verify product acceptance on social networks and develop commercial strategies to improve sales.

Although the ontology's inference processing and machine learning models can work independently, combining both approaches provide complete decision support in precision farming. Using ontology to structure and classify data collected by sensors and other monitoring equipment and applying machine learning algorithms to analyze and interpret that data enables producers and researchers to gain a deeper understanding of their animals and production. By defining the various data types collected on the farm and specifying their relationships using an ontology, machine learning algorithms can more accurately identify patterns and trends in data and make more informed predictions and recommendations.

We can use our ontology to identify the best input data and provide consistent and accurate predictions that align with the producer's reality. For example, let's consider milk production as the total milk produced over a month minus the milk discarded from sick cows. We can use the ontology to extract data from only healthy cows and predict total milk production, disregarding the sick cows that are not included in the final sum. Furthermore, ontology allows extracting data derived from semantic relationships and using them to identify events related to the farm environment. For example, temperature and humidity sensors may stop working or be damaged. Still, through the relationship between the external temperature data from the INMET API and the sensor data found on the farm, we can generate a complete dataset and use it to train intelligent algorithms that predict farm-specific weather conditions. Next, we will present the data used for the second case study (CS2).

5.4.2 Conducting Case Study 2

To assess whether the new artifact, which now contains an intelligence tier, can offer complete analyzes and thus support decision-making on the farm, we evaluated the accuracy of the intelligent algorithms and their predictive results. Initially, we chose to look at individual milk production as there are several advantages to predict milk production for a single cow. We also performed the prediction of the milk production of the farm based on the diet of the animals.

Predictive models can help producers optimize their operations, such as allocating resources more efficiently and prioritizing the most productive cows, optimizing the feed and care they provide, leading to higher milk production and greater profits. Moreover, decision-making is improved when predictive models can provide producers with valuable information, such as identifying low-performing cows (production) and improving their productivity. This fact would also lead to early warnings, as predictive models can help producers identify potential issues early on, allowing them to take timely action to prevent or mitigate the impact of those issues. For example, a model that predicts the milk production of individual cows can help producers identify cows at risk of developing health problems, allowing them to take preventive measures to keep them healthy.

To perform the individual production analysis based on the available data, it was necessary to integrate three different datasets, herd, scale, and milking. The herd data contains the animal's data, such as its identifier (earring), date of birth, and batch. The scale data refer to the animal's weighing, and finally, the milking data are milk production data. We generated a dataset containing the relationship between weight and milk production by crossing the weighing dates with the milking dates and identifying the animal through its earring in the herd. Figure 26 shows a clipping of the data used in the training dataset.

We collected data on milk production and cow weight to predict milk production based on cow weight. We pre-processed the data to clean the data to remove any missing or invalid values and normalized the set so that all data was on the same scale. We split the data into a training set and a test set. The training set was used to build the model, and the test set to evaluate model performance. We tested models like Random Forest, Neural Network, kNN, and Adaboots. A complete list of available algorithms is in the official Scikit-learning documentation. With the training of each algorithm, we evaluate the models with the test data. We use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE) to quantify forecast error. The mean absolute error represents the

Figure 26 - Fragment of weight dataset.

Brinco	Peso	Prod. Leite
1069	768	36.3
1428	650	37.8
2195	659	38.8
2414	647	23.4
2592	493	25.3
3030	606	37.2
3038	671	32.4
3063	685	36.3
3122	639	34.5

Source: Prepared by the author (2023)

mean of the absolute difference between the actual and predicted values in the data set. It measures the mean of the residuals in the data set. The mean squared error represents the difference between the original and predicted values in the data set. It measures the variance of the residuals. Root mean square error (RMSE) is used for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance. Finally, The metric R2, also known as R-squared or coefficient of determination, denotes the percentage of data variance explained by the model. The results range from 0 to 1 and are typically expressed in percentage terms, i.e., ranging from 0% to 100%. A higher R2 value indicates a more explanatory model concerning the predicted data. Therefore, we adopted the Mean Absolute Error (MAE) and the Mean Square Error (MSE), as they are metrics adopted in intelligent systems to measure the difference between predicted results and actual data evaluations (Wang and Lu, 2018).

Figure 27 illustrates the training results of various algorithms, including the Extra Tree Regressor, which showed the lowest MAE and MSE and was thus selected for use in the architecture. The Extra Tree Regressor is a meta-estimator that fits several random decision trees (also known as extra trees) on various subsamples of the data and uses the average to improve predictive accuracy and control overfitting.

In the milk prediction based on weight, we observed an MAE of 6.46, indicating that predictions generally have an error margin of 6.46L. For instance, if the algorithm predicts that an animal will produce 26L each month, upon comparing the collected data with the predicted data, that animal could have produced anywhere between 20L and 32L, meaning that there is a margin of error approximately 6L in either direction.

Figure 27 - Error data for training predict milk based on weight dataset.

Algorithm	MSE	MAE	R2	RMSE
Random Forest	69.50008693932122	6.503444770754309	0.1372757343567752	8.336671214538883
Neural Network	94.2504797759458	7.648233781085124	-0.1699579026745115	9.7082686291607
KNN KNeighbors	88.39745937961595	7.215952732644017	-0.09730270257919194	9.401992309059604
KNN Radius Neighbors	77.73230111574121	7.025485031707734	0.03508692794318935	8.816592375500935
AdaBoost	79.67775704313276	7.1172245134234755	0.010937432450248474	8.92623980425872
Bagging Regressor with SVR	78.96599352655033	7.118563476944927	0.019772754594396647	8.886281197809932
Extra Trees Regressor	**68.99721764546248**	**6.4693334203616555**	0.14351799334326498	8.306456383167403
Gradient Boosting	74.16333739876119	6.8534829026789765	0.0793894855000904	8.611813827455931
Hist Gradient Boosting	75.85523141930933	6.92239651356447	0.058387525780271066	8.709490881751318
Epsilon-Support Vector	78.68442217466269	7.110448586245102	0.023267979542755723	8.870424013239878

Source: Prepared by the author (2023)

To predict the farm's milk production based on animal feed, it was necessary to extract the kilograms consumed of each food component in the animals' diet and calculate the monthly milk production. Diet plays a significant role in milk production for cows. They need a balanced diet with the right combination of nutrients such as protein, energy, minerals, and vitamins to produce milk efficiently. If a cow's diet is deficient in any of these nutrients, it can reduce milk production. For example, a protein deficiency can lead to reduced milk production and poor quality. Animals also need sufficient energy, minerals, and vitamins in their diet to produce quality milk. Lack of these nutrients can lead to reduced milk production and poor quality. In summary, a cow's diet significantly affects her milk production. Providing cows with a balanced diet with the proper nutrients is essential to support milk production.

The challenge of making this prediction is the dietary variation the animals may undergo throughout the year due to climate variations, food availability, and even the animal's age. However, it was possible to assemble a dataset with enough data to train the algorithms. Figure 28 shows a fragment of the dataset used to train the models; it is possible to see that the Concentrate and Food Supplement component is filled with zero, as they were unavailable, and cotton was also unavailable on 07/28/2022 and 07/29/2022.

Figure 29 shows the result of training some algorithms. For example, we highlight the Ada Boost that presented lower MAE and lowers MSE and, therefore, was selected to

Figure 28 - Fragment of feed dataset.

date	Algodão	CONCENTRADO	COMPLEMENTO	Farinha de Milho	Farinha de Soja	Feno	Silagem de milho	LEITE
2021-07-02T06:24:00	58	0	0	120	136	34	978	2826
2021-07-04T07:18:00	54	0	0	114	130	32	936	2793
2021-07-05T07:36:00	54	0	0	114	130	32	936	2939
2021-07-06T07:25:00	54	0	0	114	130	32	936	2864
2021-07-03T07:36:00	54	0	0	114	130	32	936	3043
2021-07-07T07:32:00	52	0	0	110	126	30	904	2879
2021-07-08T07:15:00	52	0	0	110	126	30	904	2700
2021-07-28T07:40:00	**0**	0	0	98	122	30	966	2864
2021-07-29T07:06:00	**0**	0	0	110	138	34	1098	2453
2021-07-02T07:19:00	54	0	0	116	112	30	1156	2826
2021-07-03T08:14:00	54	0	0	116	112	30	1156	3043
2021-07-04T07:58:00	54	0	0	116	112	30	1156	2793
2021-07-07T06:34:00	100	0	0	206	186	52	1238	2879
2021-07-09T06:39:00	102	0	0	212	192	52	1278	2695
2021-07-15T06:38:00	100	0	0	204	184	52	1232	2892
2021-07-29T06:07:00	**0**	0	0	188	224	56	1492	2453

Source: Prepared by the author (2023)

act in the architecture. An AdaBoost regressor is a meta-estimator that starts by fitting a regressor to the original dataset and then fitting additional copies of the regressor to the same dataset, but where the instances' weights are adjusted according to the error of the current prediction. As such, subsequent regressors focus on hard cases.

An IoT sensor can give accurate readings if properly calibrated or maintained. Generally, IoT sensors are subject to harsh conditions, and there are several reasons why an IoT sensor may not measure temperature and humidity accurately: Over time, the accuracy of an IoT sensor may vary due to changes in its internal components or factors external factors such as temperature and humidity. Other electronic devices or signals in the environment can interfere with the operation of an IoT sensor. For example, if the sensor's batteries are not replaced regularly, this can affect the sensor's accuracy. Also, in some cases, an IoT sensor can be faulty due to a manufacturing error, which can cause it to give wrong readings, such as 0 or NULL. We use ontology to fill in these gaps to mitigate these situations and data losses.

After training the intelligent models and selecting the most suitable model to make individual and general milk production predictions, we instantiated data in the ontology. The reasoner was executed considering the SWRL rules. We developed SWRL rules to help estimate temperature and humidity, as shown in Figure 30. We used an external source, INMET API, to obtain external temperature and humidity measurements in Coronel Pacheco. As a result, we can get a pattern by comparing the API data (external) with the sensor data (internal) and using an SWRL rule to fill in the missing days.

The inference in the ontology also helped to define who are healthy and sick animals. At this point, we could filter the data considering sick animals and generate only a dataset

Figure 29 - Error data for training predict milk based on diet dataset.

Algorithm	MSE	MAE	R2	RMSE
Random Forest	27492.709913084716	132.89477835254817	0.058366246733246085	165.8092576217767
Neural Network	317190.09730729746	451.3662403876782	-9.863858192617341	563.1963221713166
KNN	29121.948232448376	136.88943952802362	0.0025643342088547127	170.65154037525818
AdaBoost	**27375.001744315967**	**130.77905901544744**	0.062397787643482916	165.4539263490473
BaggingRegressor	29672.468311993587	138.27000530733673	-0.016291147494829517	172.25698334753685
Extra Trees	27902.83166456203	133.65778647269207	0.04431945086042033	167.04140703598623
Gradient Boosting	27116.331154997508	132.46214275658917	0.07125733472522544	164.67037121169523
Hist Gradient Boosting	28687.39020841685	134.50081426278044	0.017448079917248505	169.37352274903196
Epsilon-Support Vector	29108.312600679576	136.98336572731688	0.00303135878232208	170.61158401667683

Source: Prepared by the author (2023)

Figure 30 - SWRL rules for infer sensor data.

```
Measure(?m) ^ external_measure_value(?m, ?v) ^
    swrlb:lessThan(?v, 16.0) ^
    swrlb:subtract(?r, ?v, 1.98) -> internal_measure_value(?m, ?r)

Measure(?m) ^ external_measure_value(?m, ?v) ^
    swrlb:greaterThan(?v, 88) ^ swrlb:lessThan(?v, 89.0) ^
    swrlb:add(?r, ?v, 8.64) -> internal_measure_value(?m, ?r)

Measure(?m) ^ internal_measure_value(?m, ?v) ^
    swrlb:greaterThan(?v, 95) -> is_alert(?m, 1)

Measure(?m) ^ internal_measure_value(?m, ?v) ^
    swrlb:greaterThan(?v, 23.5) ^
    swrlb:lessThan(?v, 45) -> is_alert(?m, 1)
```

Source: Prepared by the author (2023)

containing data from healthy animals (Figure 31 shows the rules used to classify animals as sick or over-productive). This filter helps to improve the dataset by adding only relevant data; otherwise, training the ML algorithms could score lower. The previous results took into consideration only the healthy animals that were filtered through the ontology.

To demonstrate that our architecture can combine ontologies and machine learning for even greater insights, we developed a feature that predicts a farm's milk production based on animal weight for the month. Heavier cows in good body condition produce more milk than underweight or thin cows. This issue is because cows need a certain amount of body fat and muscle mass to produce milk efficiently. Although other factors such as genetics, age, diet, and management practices can also affect a cow's milk production, we performed this analysis based on available farm data.

Figure 31 - SWRL rules for inferring sick and healthy cows.

```
Cow(?x) ^ is_mastitis(?x, ?m) ^ swrlb:equal(?m, 1) -> SickCow(?x)

Cow(?c) ^ DairyMilk(?d) ^ cow_milked(?d, ?cm) ^
    dairy_value(?d, ?v) ^ swrlb:greaterThan(?v, 30) ^
    is_mastitis(?c, ?m) ^ swrlb:equal(?m, 0) ^
    sameAs(?c, ?cm) -> SuperCow(?c)
```

Source: Prepared by the author (2023)

The first step is to identify and remove infected animals from the herd using SWRL rules that define sick animals, overproducers, and "normal"animals (those that are not sick and produce less than 30L of milk on average). We then took weight data from each animal group and used machine learning to predict their milk production. Finally, we add up all forecasted production to give the producer an idea of how much milk he can produce in the next month.

The previously chosen ML algorithm processed this new, improved dataset (containing further information and relationships between the data, coming from the inference processing in the ontology). In this study, the Extra Tree Regressor had a more accurate result for milk prediction based on animal weight, although, for other datasets, new tests need to be performed to choose the best model. We train and run tests with several algorithms with different parameters and choose the best based on error metrics.

5.4.3 Analyzing Case Study 2

After conducting Case Study 1, we gained important insights into using the architecture to support decision-making, which led us to conduct Case Study 2. Despite the data processing and visualization, producers felt the need for deeper analysis, such as estimating the farm's milk production for the next month and analyzing expected milk production based on animal diets. We also noticed that sensors occasionally fail, and the internal conditions of the Compost Barn are unknown. Based on scientific knowledge, we developed a new version of the architecture (CS2) that now includes an intelligent tier containing machine learning techniques and ontology to meet the need for more specific analyses. In this new version, the architecture's integration tier collected and stored data to generate a dataset containing weight and milk production data, sorted by animal and month. By processing different ML algorithms, the architecture stored the results to select the algorithm with the best accuracy. The algorithm with a low absolute mean error, considered satisfactory by the farm producers, was selected, and the trained model was made available.

Both functional requirements, such as FR003, in which the architecture must be capable of performing semantic analyses and predictions; and FR006, which requires data to be pre-processed for intelligent processing, were added. They needed to be addressed in e-Livestock. We also added sustainability-related quality requirements (NFR005) and improved the system's usability requirement (NFR006). It was necessary to review three important requirements: extensibility, flexibility, and scalability.

The architecture's sustainability (NFR005) was verified through the self-training of intelligent models, which can be triggered automatically when new datasets arrive, allowing the insights offered by the system to remain consistent and up-to-date. This requirement becomes necessary since milk production on a farm can vary throughout the year due to various factors. Some of the most common factors that can affect milk production include changes in the diet and feeding habits of animals, changes in climate and environmental conditions, changes in animal health and well-being, and changes in management practices used by producers. For example, milk production on a farm may be higher in the spring and summer months when animals have access to specific diets due to food availability. In contrast, there is a more limited diet in the winter since commonly used foods such as hay and silage are stored, and when the weather is colder it is less conducive to milk production. By keeping machine learning models trained with the most recent data set, producers can work to optimize milk production based on predictions, ensuring that the architecture always provides more realistic data. Therefore, the sustainability of the architecture is ensured through the continuous flow of data training, guaranteeing that the predictions remain updated. Figure 14 in section 4.7.2 presents the flow and steps of self-training.

We verified whether the architecture also met the requirements of extensibility (NFR002), flexibility (NFR003), and scalability (NFR004). To maintain extensibility, the architecture accommodated new controllers and provided new endpoints for the intelligence module, proving to be consistent and scalable. Regarding flexibility, the architecture allowed the addition of different intelligent models, requiring only the input of data in the expected pattern for its operation. Thus, new algorithms can be easily added without a significant impact on the functioning of the architecture. New predictions can be configured, meeting new demands that producers may have; by adding new sensors and collecting different data, the architecture continues to support producers. Finally, as new intelligent algorithms were added and more resources were demanded, the architecture remained stable and functional throughout training and evaluating the algorithms, demonstrating its potential by meeting the quality attributes defined for this work. Furthermore, it is possible to instantiate this architecture for other domains besides dairy cattle, for example, using it for the poultry domain, where the sensor data and rules differ. To do so, developing an ontology with appropriate semantic relationships and training the algorithms based on the available farm data is only necessary.

As a result, ML algorithms to the architecture improved strategic information delivery by making predictive information available. With the milk production predictions, we can cross the predictions with the actual production, check the results, and analyze if the production was better, worse, or what was expected according to the farm's planning. In addition to this analysis, with the milk forecast, it is also possible to verify if the farm is reaching its production target, suggesting to the producer changes in the composition of the foods, adequacy of the environment, and improvement in production. This strategic information is presented on panels, enabling constant monitoring of the Compost Barn environment via a mobile application.

By adding the intelligence tier, the architecture became more flexible and delivered strategic information in addition to that stored in e-Livestock repositories. New strategic information is critical to improving production decision-making at the Compost Barn. This e-Livestock support helps, for example, in the decision to increase the supply of nutrients, replace the composting bed, or open the composting windows in case the temperature and humidity increase. With continuous monitoring, ontology processing, and machine learning, the architecture could offer more complete analyzes for producers.

5.4.4 Results of Case Study 2

The field of dairy cattle management is complex and multifaceted, with many variables and factors that can affect the animals' health, productivity, and welfare. This complexity can make it challenging to develop machine learning models that can accurately capture the relationships and patterns in the data and make reliable predictions or recommendations. Additionally, machine learning models for dairy cattle may be required to handle a wide variety of data types and sources, such as sensor data, logs, and other data types, which can increase the complexity of the model development process. To mitigate this difficulty, we use ontology. Combining ontology and machine learning can significantly improve the insights gained from precision livestock systems and enable more data-driven decision-making in livestock operations.

We could infer relations, missing data, and new instances by integrating the ontology model with the ML module. Based on these new relationships, we could classify animals and generate reports. If the animal has mastitis, it is classified as a Sick Cow. If the animal is healthy and produces more than 30L of milk, it is classified as Supercow. Once the animals were classified, we could analyze the number of sick animals per batch and provide the evolution of mastitis at Compost Barn based on the farm's previous data. In addition, it is possible to alert the researchers/farmers about the possibility of a new mastitis epidemic.

As shown in Figure 32, we observed a high humidity level in the Compost Barn

Figure 32 - Animal Dataset.

Month	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Weight	640	624	630	664	671	683	643	711
Milk	-	-	38.8	46	45.1	46	46.5	46
Mastitis	true	true	-	-	-	-	-	-
Umiditty	98	99	88	89	89	85	86	85
Temperature	26.1	28.1	24	22.4	22.6	21.7	22.7	24.8

Source: Prepared by the author (2023)

during September and October, which generated a proliferation of environmental bacteria. These bacteria caused mastitis in some animals, which needed to be medicated. Due to the medications, the decision was to discard the milk. We could also detect the sudden increase in weight of one of the cows and its change to another flock. An analysis of animal weights' evolution observed a peak in one of the batches. Using the inferences provided by the ontological model, we identified that one of the cows in that batch had an insemination event, causing an increase in weight and, later, the decision to migrate to another batch. It was possible to estimate the milk production of this animal and monitor whether the expectation was reached using machine learning algorithms.

It was possible to monitor the farm through graphics from the mobile application. In the e-Livestock application, producers could follow the evolution of production over the months, receive notifications about expected production, analyze the total food consumption of animals by batch, and see health details, such as the weight of each animal. The used metrics were the following: (i) milk production per batch during the year, (ii) the weight of each animal month by month, and (iii) the feeding per kg/batch. Figure 33 presents the dashboard of the e-Livestock Mobile Application.

We could analyze the expectations for a specific month by delivering milk production prediction results. In July, for instance, the overall error was only 6.46 Liters. Each animal could produce 0.92 Liters of milk above or under the expectation during the week. As a result, it could provide the opportunity for better planning of milk production over the months, supporting decisions on the farm. During the development of the graphics, the producers evaluated both the data and the mobile application, carefully assessing the information presented therein.

5.5 CONCLUSION FROM CASE STUDIES

Both case studies assessed the capabilities provided by e-Livestock architecture, which are a) monitoring; b) reasoning; and c) automated actions. From the results obtained

Figure 33 - Dashboard Mobile Application.

Source: Prepared by the author (2023)

in both case studies, we performed a triangulation, i.e., a research method used to increase the validity and reliability of findings in a case study. We performed a few steps to triangulate Case Study 1 (CS1) and Case Study 2 (CS2), which involved collecting data from multiple sources, comparing, and integrating the data, interpreting the findings, and validating these findings to increase the validity and reliability of the results.

In Case Study 1, we verified that the architecture could monitor the farms and use dashboards through a web application. We collected sensor data related to milk production during the year, the weight of each animal by month, and feed per kg. We observed this issue through the data register and the system execution on the dashboard (triangulation). Additionally, during data processing, producers were able to receive notifications about the current production, analyze the animals' total feed consumption per batch, and view health details such as the weight of each animal. With this data, the previous observation was confirmed. When carrying out Case Study 2, we checked the research question (RQ) again, but from the point of view of the intelligent system. We could observe that the system implementing the architecture could monitor the farms as the sensors generated new data. The system collected, processed and sent to the dashboards the necessary information for the producers to decide. As a result, we can confirm that the e-Livestock system can support farm monitoring. Again, we observed this by analyzing the records in the database and observing rural producers running the dashboard (triangulation).

Performing the triangulation to answer the raised RQ, we verified that the architecture could improve the results through the system. In CS1, we could see the milk production for a specific month. When conducting CS2, which used intelligence techniques with ontology and machine learning, we performed a new evaluation with the same data incrementally. From the point of view of an intelligent system, data is always incremented constantly. As a result, it was possible to obtain greater accuracy of results by evolving to an intelligent architecture. Hence, we obtained an error of only 6.46 for July. As in this case study, the previously established average error was 0.92 L of milk above or below the expectation during the week; that is, less than 1 L of error. These data proved to be satisfactory for the producers. Therefore, when using e-Livestock architecture, greater accuracy was obtained, allowing decisions on rural properties to be more accurate. We observed this fact by analyzing the database records and the results presented through the dashboards to rural producers at the time of decisions (triangulation). With the help of e-Livestock architecture, it would be easier to estimate the farm's total production and make accurate decisions.

From the CS2 results, we confront the theoretical conjectures about the use of ML and ontology raised after CS1, and we analyze whether, after this interaction, the artifact can support decision-making with a complete analysis of precision livestock. The area of dairy cattle management is complex and multifaceted, with many variables and factors affecting the animals' health, productivity, and welfare. This complexity can make it challenging to develop ML models that can accurately capture the relationships and patterns in the data and make reliable predictions or recommendations. Furthermore, ML models for dairy cattle may be required to handle a wide variety of data types and sources, such as sensor data, logs, and other types of data, which can increase the complexity of the development process model. To mitigate this difficulty, we use ontology. Combining ontology and machine learning can improve the insights gained with precision livestock systems and enable more data-driven decision-making in livestock operations.

As a result, we could have evidence to answer the raised RQ, **"How can e-Livestock support automated monitoring, reasoning, and actions in smart farms?"**. The e-Livestock supports automated monitoring, reasoning, and actions in smart farms using ontology inferences combined with machine learning abilities.

Data were displayed in plots and analyzed by the researchers. Once a decision was made or an unusual event was detected, it was possible to track the reason and the process generated through the relationships captured by the ontology model. The results of ML algorithms offered a future insight into production planning and animal management. For example, based on the estimated production of the farm, managers can see if it is in line with the objective outlined in the quarter. They can develop action plans to improve production through feeding or disease control if production is far below the target.

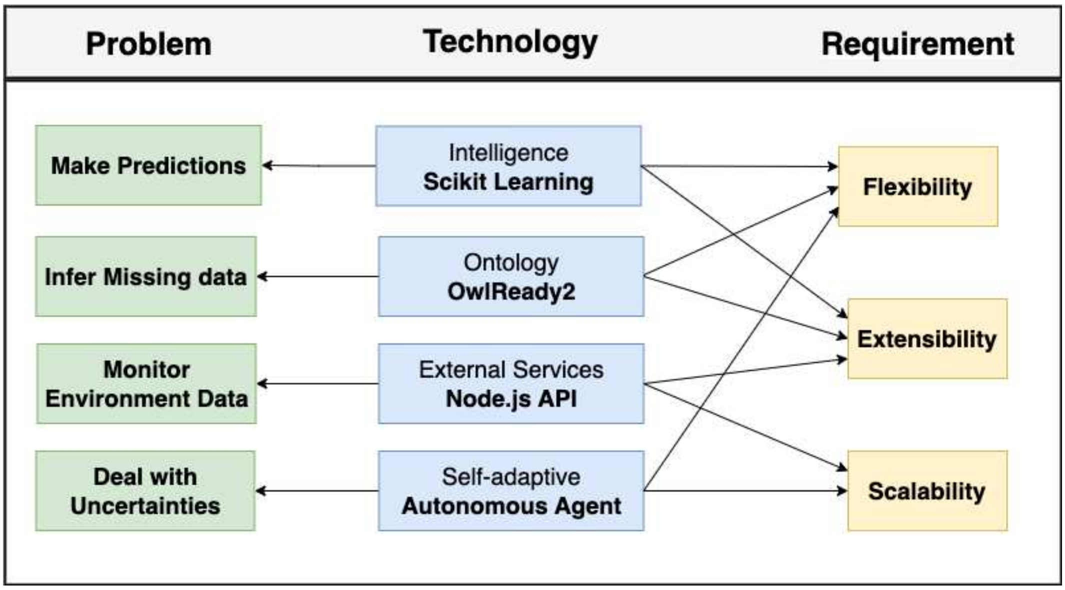
During the execution of both case studies, we could verify that e-Livestock could provide implicit information derived from intelligent data analysis. This information, presented to the researcher/farmer/producer in the dashboard, supports the decision-making process on farms. In addition to this support, we could track the information processing that derived the decision support, providing more confidence in the decision support. We discovered this information by processing inferences over the ontology instances, generating implicit knowledge from new relationships, i.e., processing the information semantically and based on ML algorithms.

The overall aim of the assessment was to observe the influence of the deployed system in the Compost Barn field towards the pursued goals: maximizing animal welfare and increasing productivity, product quality, and sustainability. Evidence brought allowed us to conclude that the proposed architecture can satisfy the mentioned attributes by maximizing animal welfare (making decisions to make the temperature and humidity close to the ideal thresholds), increasing productivity (a direct result of the architectural evolution between CS1 and CS2), product quality (by avoiding selling low-quality milk predicted as contaminated with mastitis) and sustainability (with automatic adjustments in the exhaust fans, turning them on/off, potentially saving energy).

The sensors installed both in the environment and on the animals' necks captured valuable data, which was then analyzed by e-Livestock to provide strategic insights to support decision-making on farms. By leveraging this data, e-Livestock was able to estimate the likelihood of production loss or animal disease, among other critical events. Through careful evaluation of the data collected at the Compost Bar, we were able to obtain compelling evidence that helped us address our research question. However, these results cannot be generalized, and new case studies must be conducted in additional real-world context farms.

Figure 34 presents an overview of the technologies employed and the primary requirements utilized to address the research question. Moreover, it outlines the specific problem that each technology resolves. The Intelligence tier was designed to fulfill the flexibility requirement (NFR003), which is accomplished by dynamically selecting the most appropriate machine learning algorithm. We achieved extensibility requirement (NFR002) through ontology models, which extract the relationship between data and inferences. The architecture can incorporate new controllers and provide new endpoints for the Intelligence tier to access more farm data, while external services assist with extensibility and scalability (NFR004). Finally, the "Autonomous Agent"can trigger alerts or modify the AC based on intelligence module information. Consequently, the Autonomous Agent can search for available devices or operators that need to receive alerts and/or process changes directly on the devices or in the AC operation. As the sensors collect new data, the autonomous agent continuously monitors it to process it, thereby supporting the management of uncertainties.

Figure 34 - Relationship Between Technologies and Requirements.



Source: Prepared by the author (2023)

5.6 THREATS TO VALIDITY

This section discusses threats to validity that can affect or limit the results' validity. The limitations of this research are related to the intelligence tier. We only executed the case studies on the Coronel Pacheco Compost Barn system. Other studies could be conducted to explore the prediction results considering more than one farm. The ontology model was designed for dairy cattle only. All experiments were conducted for cattle inside a barn and did not consider free animals.

Construct validity. During the case studies, the data processed was available to the researchers. All data updates can be tracked and visualized. However, more than intelligent analysis is needed. Considering different contexts, we can use additional data analysis to mitigate this threat. Moreover, e-Livestock supports different ML techniques. However, the dataset and the number of researchers/farmers that analyzed the results can represent a threat. Additional evaluations need to be conducted to reduce this threat. Internal validity: During the conduction of the case studies, the data are from specific sensors from the Compost Barn of Embrapa's experimental farm. The results are still preliminary, and although they indicate a valuable outcome, a more detailed study is needed to present additional findings. However, the features offered by the e-Livestock architecture can pose a threat. In a more complex context, other data analysis techniques need to be used, and, as a result, we must reassess the decision support.

External validity: The case study deals with a dataset associated with a specific Compost Barn production system experiment. We need to conduct evaluations considering other agricultural contexts before generalizing our results. However, it is possible to identify situations where we can obtain similar CS results and the knowledge acquired can be transferred to similar real-world experiments.

Reliability: We presented details of the execution of the studies, but probably some information was probably incomplete. We have made documentation available to ensure the case studies reruns to mitigate this threat.

5.7 FINAL REMARKS OF THE CHAPTER

Considering the difficulties of decision-making in Agriculture 4.0, this work sought to support decisions in this domain through an architectural approach. For that, the e-Livestock architecture was proposed. It performs the collection, processing, storage, and enrichment of data collected by sensors, whether installed on animals or in the environment in which they are found. Subsequently, two case studies were conducted to evaluate the proposed solution. Evidence was presented on the feasibility of instantiating this architecture to support decision-making in the context of precision livestock. Agricultural decisions must be made considering the diversity of information and devices in different contexts. Furthermore, context information is often not used in decision-making due to the complexity of managing a high volume of heterogeneous data. This work presents an architecture that aims to tackle the problems of collecting, processing, and visualizing data to support decision-making. Still, it was possible to support decisions with external information and data from other sources.

ML techniques could improve the use of the IoT system in a precision livestock operation. Machine processing algorithms were used to analyze data collected by sensors and other monitoring equipment and identify patterns and trends that may not be immediately apparent to humans. This analysis can help producers and researchers decide how to manage their animals and improve the overall efficiency and sustainability of the farm. On the other hand, ontologies are often used in the context of artificial intelligence and semantic web technologies that can be used to help computers better understand and interpret data. In precision livestock, ontology was used to help organize and classify data collected by sensors.

Our results are relevant since they address problems related to world food production. Our approach has the potential to be replicated in food production research institutions all over the world, besides being a contribution to scientific and livestock technological solutions. We also intend to reinforce and prioritize quality attributes, such as flexibility, extensibility, and scalability. New semantic rules can also be defined to support data enrichment in the ontology and their integration with other domain-specific ontologies to increase the capacity of knowledge extraction. Finally, it would be useful to conduct new experiments in other livestock subdomains to evaluate the support offered by e-Livestock architecture in different application subdomains.

We checked that the architecture met the extensibility, flexibility, and scalability requirements. Aimed to fulfill extensibility, the architecture could accommodate new controllers and provide new endpoints for the intelligence module, proving consistent and scalable. Regarding flexibility, the architecture allowed the addition of different intelligent models, requiring only the input of data in the expected pattern for its operation. Hence, we could easily add new algorithms without a significant impact on the functioning of the architecture. Furthermore, as we demanded new intelligent algorithms and more resources, we observed that the architecture remained stable and functional throughout the training and evaluation process of the algorithms, showing its potential to meet the elicited quality attributes.

e-Livestock is an architecture of an ecosystem platform that offers services to support decision-making in agribusiness. By evaluating the architecture of an agricultural research corporation, we evidenced that we could monitor the health and well-being of the animals. This issue was possible by using an intelligent architecture capable of enriching data to support decisions through the services offered by the platform. As a result, different partners using e-Livestock can interact and collaborate accordingly, innovating in a highly competitive market. From this solution, they can also align decisions and change strategies and relationships with their external stakeholders to create value and new opportunities for agribusiness. So, using an intelligent architecture goes beyond the solutions proposed by individual organizations and rural properties that do not aim at attracting partners to an innovation ecosystem (Bosch, 2016) (Bosch and Olsson, 2018).

6 CONCLUSION

This work lays out a theoretical framework of key concepts about the Internet of Things (IoT), the semantic web with ontology, and intelligent decision support systems. The work begins by introducing the domain of decision support systems in agriculture 4.0 and identifying the challenges related to leveraging sensor data for decision-making in precision livestock. After that, the work delves into the concepts of ontology and computational intelligence employed in the proposed solution.

To answer the established research question: **"How can e-Livestock support automated monitoring, reasoning and actions in smart farms?"** Given the difficulties associated with decision-making in Agriculture 4.0, the work seeks to address this issue by proposing an architectural approach. An architecture of a software-based production system for milk production, so-named e-Livestock, was developed and deployed in a Compost Barn environment of a real farm in Brazil. To evaluate the effectiveness of the proposed architecture, a case study was conducted, and evidence was presented on its feasibility in supporting decision-making in the realm of precision livestock. The proposed solution, the e-Livestock architecture, collects, processes, stores, and enriches data obtained through sensors, which may be installed on animals or in their immediate environment.

As a contribution of this work, we have published the systematic mapping (Gomes et al., 2021), the first version of the architecture (Gomes et al., 2021), the ontology model (Gomes et al., 2021), the second cycle (Gomes et al., 2023), and (Gomes et al., 2022). In addition to its contributions to decision support, the proposed solution may benefit research groups working on decision-making platforms. Therefore, the work concludes by highlighting the value of this work to a broader audience.

- This study presents a systematic literature review that identifies and categorizes the main works in Agriculture 4.0. As a contribution, this review offers opportunities for further research by providing a state-of-the-art overview of decision support systems. By advancing knowledge in this field, we highlight integration techniques and opportunities utilized in previous studies, demonstrating the importance of using IoT sensor data to support decision-making. As a result, complex decisions related to agribusiness can be enriched with information from other systems used on farms. Additional topics could be explored for future research, especially regarding integrating IoT devices and systems for agribusiness. Furthermore, research on data provenance should be developed, considering the context of IoT data to enrich and improve decision-making in agriculture.

- This study also presents a conceptual architecture for handling IoT data in rural environments. This architecture contributes to the scientific community by providing a solution that considers semantics and artificial intelligence to enrich data for decision-making in this domain

- In addition, this study developed an ontology capable of modeling and extracting implicit knowledge about dairy cattle. This ontology is accessible via a web service and enables interoperability with external platforms.

- The integration of the proposed architecture with other platforms, such as INMET, increased decision support by providing external information to the farm. Furthermore, this integration generated knowledge about using APIs and common data models for integrating systems, opening opportunities for additional integrations.

- Finally, this study also developed an architecture considering a tier of intelligence combining semantic inferences and machine learning predictions. To enrich data and provide better decision support on farms, the study presents the visualization of predictive results and farm information on a mobile device.

Decisions in agriculture need to be made considering the diversity of information and devices present in different contexts. Furthermore, context information is often not used in decision-making due to the complexity of managing a high volume of heterogeneous data. This work presented an architecture that aimed to tackle the problems of collecting, processing, and visualizing data to support decision-making. Still, supporting decisions with external information and data from other sources was possible. Our results are relevant since they addressed problems related to world food production. Our approach has the potential to be replicated in food production research institutions over the world, besides being a contribution to scientific and livestock technological solutions. We also intend to invest in reinforcing and prioritizing quality attributes, such as data provenance, interoperability, and reliability (as highlighted by Fernandes et al. (2021); Valle et al. (2021); Ferreira et al. (2021)). New rules can also be defined to support data enrichment in the ontology and their integration with other domain-specific ontologies to increase the capacity of knowledge extraction. Finally, it would be useful to conduct new experiments in other livestock subdomains to evaluate the support offered by e-Livestock architecture in different application subdomains.

For future work, we intend to generate other instances of e-Livestock architecture and associate them with a software ecosystem and explore aspects of collaboration, communication, and integration between farms to support decisions in these instances. Combining and processing additional data sources and sensors also can lead to more accurate results, reduce costs, and maintain agribusiness sustainability. Furthermore, it is essential to focus on improving the training process, when it comes to machine learning. This can be achieved by repeating the training of the same algorithm multiple times to verify the results. Additionally, it is crucial to develop a clear and well-defined pipeline for the maintenance and deployment of the model in production.

REFERÊNCIAS

Symeonaki E., Arvanitis K., Piromalis D. **A Context-Aware Middleware Cloud Approach for Integrating Precision Farming Facilities into the IoT toward Agriculture 4.0**. Applied Sciences.10(3), 2020, 813; https://doi.org/10.3390/app10030813.

Farooq, M. S., Riaz, S., Abid, A., Abid, K., & Naeem, M. A. **A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming**. IEEE Access 2019, 7, 156237-156271. doi: 10.1109/ACCESS.2019.2949703.

Graciano-Neto, V. V., Kassab, M., Lopes, V., Oliveira, R., & Bulcão-Neto, R. **The State of IoT for Agribusiness in Brazil**. Computer, 2022, 55(12), p. 140-144.

Kitchenham B. **Procedures for performing systematic review**. 2004, Vol. 33, Keele University, Keele, UK.

Zhai, Z., Martínez, F. J., Beltran, V., & Martínez, L. N. **Decision Support Systems for Agriculture 4.0: Survey and Challenges**. Computers and Electronics in Agriculture 2020, 170, 105256. doi: 10.1016/j.compag.2020.105256.

Villa-Henriksen, A., Edwards, G. T., Pesonen, L. A., Green, O., & Sørensen, C. A. G. **Internet of Things in Arable Farming: Implementation, Applications, Challenges and Potential**. Biosystems Engineering 2020, 191, 60-84. doi: 10.1016/j.biosystemseng.2019.12.013.

Bahlo, C., Dahlhaus, P., Thompson, H., & Trotter, M. **The Role of Interoperable Data Standards in Precision Livestock Farming in Extensive Livestock Systems: A Review**. Computers and Electronics in Agriculture 2019, 156, 459-466. doi: 10.1016/j.compag.2018.12.007.

Newlands, N., Ghahari, A., Gel, Y. R., Lyubchich, V., & Mahdi, T. **Deep Learning for Improved Agricultural Risk Management**, Analytics And Ai For Industry - Specific Applications 2019.

Chavan, G. and Momin, B.. **An integrated approach for weather forecasting over Internet of Things: A brief review**. 2017, In international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC), p. 83-88, IEEE.

Gualdi, F., & Cordella, A. **Artificial Intelligence and Decision-Making: The Question of Accountability**. Proceedings of the 54th Hawaii International Conference on System Sciences, 2021.

Lakshmi, V., & Corbett, J. **How Artificial Intelligence Improves Agricultural Productivity and Sustainability: A Global Thematic Analysis**. Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

MilkPoint. **Compost barn vs free stall: diferenças de ocorrência de mastite e conforto**. 2016. Disponível em: www.milkpoint.com.br/colunas/marco-veiga-dos-santos/compost-barn-vs-free-stall-diferencas-de-ocorrencia-de-mastite-e-conforto. Acesso em: 10 set. 2020.

Louie, B., Mork, P., Martin-Sanchez, F., Halevy, A., & Tarczy-Hornoch, P. **Data integration and genomic medicine**, 2007, Journal of Biomedical Informatics. https://doi.org/10.1016/j.jbi.2006.02.007

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., & Hoffman, M. M. **Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities**, 2019, Information Fusion. https://doi.org/10.1016/j.inffus.2018.09.012

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. and Dean, M. **SWRL: A semantic web rule language combining OWL and RuleML**, 2004, W3C Member submission, 21(79), p. 1-31.

Belciug, S., & Gorunescu, F. **Intelligent Decision Support Systems-A Journey to Smarter Healthcare** p. 130-137, 2020, Berlin and Heidelberg: Springer International Publishing.

Sprague, R. **A framework for the development of decision support systems**. 1980. MIS Quarterly.4(4), p. 1-15.

Pietersma, D., Lacroix, R. and Wade, K.M. **A framework for the development of computerized management and control systems for use in dairy farming**, 1998. Journal of dairy science, 81(11), p. 2962-2972.

McGuinness, D. L., & Van Harmelen, F. **OWL web ontology language overview**. W3C recommendation 2004, 10(10).

O'Connor, M. J., Shankar, R. D., Nyulas, C., Tu, S. W., & Das, A. K. **Developing a Web-Based Application using OWL and SWRL**. In AAAI spring symposium: AI meets business rules and process management, 2008, p. 93-98.

Weyns, D. **An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective**. 2020. John Wiley & Sons.

Gruber, T.R. **Toward principles for the design of ontologies used for knowledge sharing?**. International journal of human-computer studies, 1995, 43(5-6), p. 907-928.

Hevener A., **A Three Cycle View of Design Science Research**. 2007. Scandinavian Journal of Information Systems. 2007, 19(2), p. 87-92.

Wang, W., & Lu, Y. **Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model**. In IOP conference series: materials science and engineering, Vol. 324, No. 1, 2018, p. 012049, IOP Publishing.

Pimentel, M., Filippo, D., & Santoro, F. M. **Design Science Research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação**, 2019, Metodologia de Pesquisa em Informática na Educação: Concepção da Pesquisa, Porto Alegre: SBC.

da Silva Amorim, S., McGregor, J. D., de Almeida, E. S., & von Flach G. Chavez, C. **Flexibility in ecosystem architectures**. In Proceedings of the 2014 European Conference on Software Architecture Workshops, 2014, p. 1-6.

Amorim, S. D. S., De Almeida, E. S., & McGregor, J. D.. **Extensibility in ecosystem architectures: an initial study**. In Proceedings of the 2013 International Workshop on Ecosystem Architectures, 2013, p. 11-15.

da Silva Amorim, S., de Almeida, E. S., & McGregor, J. D. (2014, April). **Scalability of ecosystem architectures**. In 2014 IEEE/IFIP Conference on Software Architecture, 2014, p. 49-52, IEEE.

Yin, R. K. **Estudo de Caso: Planejamento e métodos**. 2015. 5. ed., Porto Alegre: Bookman.

Budgen, D., Turner, M., Brereton P., Kitchenham, B. **Using mapping studies in software engineering**, in Proceedings of PPIG, 2008.

Helfer, G. A., Barbosa, J. L., Costa, A. B. D., Martini, B. G., Santos, R. D. **A model for productivity and soil fertility prediction oriented to ubiquitous agriculture**, In Proceedings of the 25th Brazillian Symposium on Multimedia and the Web, 2019, p. 489–492, doi: 10.1145/3323503.3360637.

Karthick, G. S., Sridhar, M., Pankajavalli, P. B. **Internet of things in animal healthcare (IoTAH): review of recent advancements in architecture, sensing technologies and real-time monitoring**. SN Computer Science, 2020, 1(5).

Kitchenham, B., Charters S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Keele University and University of Durham, 2007, EBSE Technical Report Version 2.3.

Mourão E., Kalinowski M., Murta L., Mendes E., Wohlin C.**Investigating the Use of a Hybrid Search Strategy for Systematic Reviews**, ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2017, p. 193-198, doi: 10.1109/ESEM.2017.30.

Uschold, M. and Gruninger, M. **Ontologies: Principles, methods and applications**, The knowledge engineering review, 1996, 11(2), p. 93-136.

Chen, J., Hu, P., Jimenez-Ruiz, E., Holter, O.M., Antonyrajah, D. and Horrocks, I. **Owl2vec*: Embedding of owl ontologies**. Machine Learning, 110(7), 2021, p. 1813-1845.

Runeson, Per, and Martin Höst. **Guidelines for conducting and reporting case study research in software engineering**, Empirical software engineering 14, 2009, p. 131-164.

Vassiliadis, P., Simitsis, A. and Skiadopoulos, S. **Conceptual modeling for ETL processes**, In Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP, 2002, p. 14-21.

Bosch, J. and Olsson, H.H. **Ecosystem traps and where to find them**. Journal of Software: Evolution and Process, 30(11), 208, p. e1961.

Bosch, J. **Tutorial summary for speed, data and ecosystems: The future of software engineering**. In 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), 2016, p. 254-254, IEEE.

Fernandes, J., Graciano Neto, V.V. and Santos, R.P.D., **An Approach Based on Conceptual Modeling to Understand Factors that Influence Interoperability in Systems-of-Information Systems**, In XX Brazilian Symposium on Software Quality, 2021, p. 1-10.

Valle, P.H.D., Garcés, L. and Nakagawa, E.Y. **Architectural strategies for interoperability of software-intensive systems: practitioners' perspective**, In Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, p. 1399-1408.

Ferreira, F.H., Nakagawa, E.Y. and dos Santos, R.P. **Reliability in software-intensive systems: challenges, solutions, and future perspectives**, In 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2021, p. 54-61, IEEE.

Deacon, J., **Model-view-controller (mvc) architecture**. Online, 2009, http://www.jdl.co.uk/briefings/MVC. pdf, 28.

Yang, L., Yang, S.H. and Plotnick, L., **How the internet of things technology enhances emergency response operations**, 2013, Technological Forecasting and Social Change, 80(9), p. 1854-1867.

Simon, H. A., **The Science of the Artificial**, Cambridge, 1996, 3rd ed, Mass.: MIT Press.

Cole, R., Purao, S., Rossi, M. and Sein, M., **Being proactive: where action research meets design research**, ICIS 2005 proceedings, p. 27.

Cole, R., Purao, S., Rossi, M. and Sein, M., **Action research is similar to design science**, 2007, Quality & quantity, 41, p. 37-54.

Gomes, J.S., David, J.M.N., Braga, R., Arbex, W., Barbosa, B., Gomes, W.L. and Fonseca, L.M.G., **Sistemas de suporte à decisão nas fazendas inteligentes: um mapeamento sistemático**, 2021, In Anais do XIII Congresso Brasileiro de Agroinformática, p. 301-310. SBC.

Gomes, J., Lopes, V.C., Neto, V.V.G., De Oliveira, R.F., Kassab, M., David, J.M.N., Braga, R. and Arbex, W., **Deriving experiments from E-SECO software ecosystem in the technology transfer process for the livestock domain**, 2022, In Proceedings of the 10th IEEE/ACM International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems, p. 1-8.

Gomes, J.S., David, J.M.N., Braga, R., Arbex, W., Barbosa, B., Gomes, W.L. and Fonseca, L.M.G., **e-LivestockProv: An Architecture based on Provenance to Manage Traceability in Precision Livestock Farming**, 2021, In Anais do I Workshop de Práticas de Ciência Aberta para Engenharia de Software, SBC, p. 43-48.

Gomes, J.S., David, J.M.N., Braga, R., Ströele, V., Arbex, W., Barbosa, B., Gomes, W.L. and Fonseca, L.M.G., **Architecture for Decision Support in Precision Livestock Farming** 2021, In Anais do XV Brazilian e-Science Workshop, SBC, p. 41-48.

Jonas Gomes, Izaque Esteves, Valdemar Vicente Graciano Neto, José Maria N. David, Regina Braga, Wagner Arbex, Mohamad Kassab, Roberto Felício de Oliveira, **A scientific software ecosystem architecture for the livestock domain**, 2023, Information and Software Technology, Volume 160, 107240, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2023.107240.

Symeonaki E., Arvanitis K., Piromalis D. **A Context-Aware Middleware Cloud Approach for Integrating Precision Farming Facilities into the IoT toward Agriculture 4.0**. Applied Sciences, 10(3), 2020, https://doi.org/10.3390/app10030813.

Kakamoukas, G., Sariciannidis, P., Livanos, G., Zervakis, M., Ramnalis, D., Polychronos, V., Karamitsou, T., Folinas, A. and Tsitsiokas, N., **A multi-collective, IoT-enabled, adaptive smart farming architecture**, In IEEE International Conference on Imaging Systems and Techniques (IST), 2019, p. 1-6, IEEE.

Kamilaris, A., Gao, F., Prenafeta-Boldu, F. X., Ali, M. I. **Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications**, IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016, p. 442-447, doi: 10.1109/WF-IoT.2016.7845467.

Rezk, N. G., Hemdan, E. E., Attia, A., El-Sayed, A., Elrashidy, M.A. **An efficient IoT based smart farming system using machine learning algorithms**, Multimedia Tools and Applications, v.80, 2021, p. 773-797.

Sowmya B. J., Krishna Chaitanya S., Seema S., K.G. Srinivasa, **Data Analytic Techniques for Developing Decision Support System on Agrometeorological Parameters for Farmers**, International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), IGI Global, v.14(2), 2020, p. 92-107, doi: 10.4018/IJCINI.2020040106.

Visconti P., de Fazio R., Velázquez R., Del-Valle-Soto C., Giannoccaro N. I. **Development of Sensors-Based Agri-Food Traceability System Remotely Managed by a Software Platform for Optimized Farm Management**, Sensors. 20(13), 2020, doi: 10.3390/s20133632.

Mikhaylenko, I. M., Timoshin, V. N. **Expert strategic management systems in precision farming**, In Journal of Physics: Conference Series. IOP Publishing, 2019, doi: 10.1088/1742-6596/1419/1/012030.

Rekha, P., Rangan, V. P., Ramesh, M. V., Nibi, K. V. **High yield groundnut agronomy: An IoT based precision farming framework**, IEEE Global Humanitarian Technology Conference (GHTC), 2017, p. 1-5, doi: 10.1109/GHTC.2017.8239287.

Borisenko, I., Gangur, M.. Kobylko, A., Romanov, D., Smirnova, O., **Information support system for management decisions in the agricultural sector**, International Scientific Conference on Agricultural Machinery Industry, v. 403, 2019, Issue 1, doi: 10.1088/1755-1315/403/1/012083.

Dabre, K. R., Lopes, H. R., D'monte, S. S. **Intelligent Decision Support System for Smart Agriculture**, International Conference on Smart City and Emerging Technology. 2018, p. 1-6, doi: 10.1109/ICSCET.2018.8537275.

Yusianto, R. R., Marimin, Suprihatin, Hardjomidjojo, H. **Intelligent Spatial Decision Support System Concept in the Potato Agro-Industry Supply Chain**, International Conference on Computer Science and Its Application in Agriculture (ICOSICA), 2020, p. 1-7, doi: 10.1109/ICOSICA49951.2020.9243233.

Loret, N., Affinito, A., Bonanomi, G. **Introducing Evja-Rugged Intelligent Support System for precision farming**, ACTA IMEKO, 9(2), 2020, p. 83 – 88.

Kale, A. P., Sonavane, S. P. **IoT based Smart Farming: Feature subset selection for optimized high-dimensional data using improved GA based approach for ELM**, Computers and Electronics in Agriculture, v. 161, 2019, p. 225-232. doi: 10.1016/j.compag.2018.04.027

Zhang X., Zhang J., Li L., Zhang Y., Yang G. **Monitoring Citrus Soil Moisture and Nutrients Using an IoT Based System**, Sensors (Basel). 17(3), 2017, p. 447. doi:10.3390/s17030447.

Karim, F., Karim, F., Ali frihida, **Monitoring system using web of things in precision agriculture**, Procedia Computer Science, v.110, 2017, p. 402-409. doi: 10.1016/j.procs.2017.06.083.

Tripathy, P. K., Tripathy, A. K., Agarwal, A., Mohanty, S. P. (2021) **MyGreen: An IoT-Enabled Smart Greenhouse for Sustainable Agriculture**, IEEE Consumer Electronics Magazine. v. 10, no. 4, 2021, p. 57-62, doi: 10.1109/MCE.2021.3055930.

Kamath, R., Balachandra, M., Prabhu, S. **Raspberry Pi as Visual Sensor Nodes in Precision Agriculture: A Study**, In IEEE Access, v. 7, 2019, p. 45110-45122, doi: 10.1109/ACCESS.2019.2908846.

Rotondi, D., Straniero, L., Saltarella, M., Balducci, F., Impedovo, D., Pirlo, G. **Semantics for Wastewater Reuse in Agriculture**, IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, p. 598-603, doi: 10.1109/SMC.2019.8913949.

Suakanto, S., Engel, V. J. L., Hutagalung, M., Angela, D. **Sensor networks data acquisition and task management for decision support of smart farming**, International Conference on Information Technology Systems and Innovation (ICITSI). 2016, p. 1-5, doi: 10.1109/ICITSI.2016.7858233.

Sakthi, U., Rose, J. D. **Smart Agricultural Knowledge Discovery System using IoT Technology and Fog Computing**, Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, p. 48-53, doi: 10.1109/ICSSIT48917.2020.9214102.

Balan T., Dumitru C., Dudnik G., Alessi E., Lesecq S., Correvon M., Passaniti F., Licciardello A. **Smart Multi-Sensor Platform for Analytics and Social Decision Support in Agriculture**, Sensors. 20(15), 2020, p. 4127, doi: 10.3390/s20154127.

Khanna, A., Kaur, S. **Wireless Sensor and Actuator Network (s) and its significant impact on Agricultural domain**, Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2020, p. 384-389, doi: 10.1109/PDGC50313.2020.9315822.

Vazquez, J. P. G., Torres, R. S., Perez, D. B. P., Demarigny, Y., Soldat, V., Gemelas, L., & Bersimis, F. G. **Scientometric analysis of the application of artificial intelligence in agriculture**, Journal of Scientometric Research, 10(1), 2021, p. 55-62.

DAYIOĞLU, M. A., & Turker, U. **Digital Transformation for Sustainable Future-Agriculture 4.0: A review**, Journal of Agricultural Sciences, 27(4), 2021, p. 373-399.

Anbananthen, K. S. M., Subbiah, S., Chelliah, D., Sivakumar, P., Somasundaram, V., Velshankar, K. H., & Khan, M. A. **An intelligent decision support system for crop yield prediction using hybrid machine learning algorithms**, F1000Research, 10, 2021.

Dayalini, S., Sathana, M., Navodya, P. R., Weerakkodi, R. W. A. I. M. N., Jayakody, A., & Gamage, N. **Agro-Mate: A Virtual Assister to Maximize Crop Yield in Agriculture Sector**, In TENCON 2021-2021 IEEE Region 10 Conference (TENCON), 2021, p. 387-392, IEEE.

El Hachimi, C., Belaqziz, S., Khabba, S. and Chehbouni, A., **Towards precision agriculture in Morocco: A machine learning approach for recommending crops and forecasting weather**, International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), 2021, p. 88-95, IEEE.

Fawzi, H., Mostafa, S. A., Ahmed, D., Alduais, N., Mohammed, M. A., & Elhoseny, M. **TOQO: A new Tillage Operations Quality Optimization model based on parallel and dynamic Decision Support System**, Journal of Cleaner Production, 316, 2021, p. 128263.

Ting, L., Khan, M., Sharma, A., & Ansari, M. D. **A secure framework for IoT-based smart climate agriculture system: Toward blockchain and edge computing**, Journal of Intelligent Systems, 31(1), 2022, p. 221-236.

Sarma, K. K., Das, K. K., Mishra, V., Bhuiya, S., & Kaplun, D. **Learning Aided System for Agriculture Monitoring Designed Using Image Processing and IoT-CNN**. IEEE Access, 10, 2022, p. 41525-41536.

Sudha, M. K., Manorama, M., & Aditi, T. **Smart Agricultural Decision Support Systems for Predicting Soil Nutrition Value Using IoT and Ridge Regression**, AGRIS on-line Papers in Economics and Informatics, 14(665-2022-513), 2022, p. 95-106.

Sakthi, U., & DafniRose, J. **Blockchain-Enabled Smart Agricultural Knowledge Discovery System using Edge Computing**, Procedia Computer Science, 2022, p. 73-82.

Elijah, O., Rahim, S. K., Abioye, E. A., Salihu, Y. O., & Oremeyi, A. A. **Decision Support Platform for Production of Chili using IoT, Cloud Computing, and Machine Learning Approach**, In 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), 2022, p. 1-5, IEEE.

Catalano, C., Paiano, L., Calabrese, F., Cataldo, M., Mancarella, L., & Tommasi, F. **Anomaly detection in smart agriculture systems**, Computers in Industry, 143, 2022, p. 103750.