UNISINOS

Programa de Pós-Graduação em

Computação Aplicada

Doutorado Acadêmico

Jorge Arthur Schneider Aranda

Freya: An Event Prediction Model for Power Distribution Networks

São Leopoldo, 2024

UNIVERSITY OF VALE DO RIO DOS SINOS — UNISINOS
ACADEMIC UNIT OF RESEARCH AND GRADUATE STUDIES
APPLIED COMPUTING GRADUATE PROGRAM
ACADEMIC DOCTORATES

JORGE ARTHUR SCHNEIDER ARANDA

FREYA: AN EVENT PREDICTION MODEL FOR POWER DISTRIBUTION NETWORKS

São Leopoldo
2024

Jorge Arthur Schneider Aranda

FREYA: AN EVENT PREDICTION MODEL FOR POWER DISTRIBUTION NETWORKS

Thesis proposal presented as a partial requirement to obtain the Doctor's degree from the Postgraduate Program in Applied Computation of the Unisinos University.

Advisor:
Prof. Dr. Jorge Luis Victória Barbosa

Co-advisor:
Prof. Dr. Paulo Ricardo da Silva Pereira

São Leopoldo
2024

A662f Aranda, Jorge Arthur Schneider.
Freya : an event prediction model for power distribution networks / by Jorge Arthur Schneider Aranda. – 2024.
216 f. : il. ; 30 cm.

Thesis (doctor's degree) — Universidade do Vale do Rio dos Sinos, Postgraduate Program in Applied Computation, São Leopoldo, RS, 2024.
Advisor: Dr. Jorge Luis Victória Barbosa.
Co-advisor: Dr. Paulo Ricardo da Silva Pereira.

1. Smart grid. 2. Machine learning. 3. Ubiquitous computing. 4. Ontology. 5. Power distribution. I. Title.

UDC: 004.85:621.311

Cataloging in Publication (CIP):
Librarian Alessandro Dietrich - CRB 10/2338

Moderação na defesa da verdade é um serviço prestado à mentira.

— Olavo de Carvalho

ACKNOWLEDGEMENTS

Agradeço a todos que me ajudaram em minha trajetória acadêmica e profissional: Colegas, amigos e principalmente a minha Família. A minha esposa Cristiane e a minha mãe Iris que sempre estiveram comigo nos momentos bons e ruins desta jornada. Agradeço ao professor e orientador Jorge Barbosa, que sempre me incentivou a escrever e publicar artigos acadêmicos de alta qualidade em periódicos de alto impacto no meio cientifico, por seu profissionalismo e paciência. Agradeço também meu coorientador Professor Dr. Paulo Ricardo da Silva Pereira pela dedicação ao me auxiliar na condução deste trabalho. Agradeço aos colegas do Mobilab pelos conselhos, conversas e ensinamentos, em especial aos colegas Ricardo dos Santos Costa e Vitor Werner de Vargas, Bruno Martini e Robson Lima. Agradeço o apoio do CNPq / Brasil (Conselho Nacional de Desenvolvimento Científico e Tecnológico - http://www.cnpq.br), CAPES / Brasil (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - http://www.capes.gov.br), CEEE Equatorial (https://ceee.equatorialenergia.com.br/) e Certaja Energia (https://www.certaja.com.br/energia/).

“*Without education, we are in a horrible and deadly danger of taking educated people seriously.*”.
(G.K. Chesterton)

RESUMO

No contexto das Smart Grids (SGs), as concessionárias de energia gerenciam extensos volumes de dados para monitorar e otimizar as redes de distribuição. Os principais parâmetros incluem níveis de tensão, corrente e potência, bem como indicadores de falha, como correntes de curto-circuito e quedas de tensão. Detectar e prever falhas técnicas é fundamental para prevenir faltas de energia que afetam consumidores residenciais e industriais. A estimativa de event, o processo de previsão das condições futuras do SG, é essencial para identificar potenciais falhas técnicas. Os SGs apresentam equipamentos distribuídos hierarquicamente por grandes áreas geográficas, influenciando a rede global com base na sua importância hierárquica e nos contextos históricos individuais. A comunicação contínua entre esses dispositivos e os centros de monitoramento é vital para a operação eficaz do SG. A integração de conceitos como Edge Computing (EC), Internet das Coisas (IoT) e Machine Learning (ML) melhora a estimativa de evento e a eficiência operacional em SGs. Esta tese apresenta o modelo Freya, uma estrutura computacional inteligente projetada para estimativa de eventos em SGs, com foco na distribuição de energia. A contribuição científica do modelo Freya reside na estimativa de eventos tanto ao nível do equipamento como da rede, com especial ênfase na importância hierárquica e na influência do atual contexto. A análise comparativa mostra que Freya aborda exclusivamente três aspectos estratégicos: (1) operação de equipamentos remotos, (2) influência do contexto nos SGs e (3) importância hierárquica dentro da rede. Esses aspectos servem como insumos para modelagem preditiva. A detecção de eventos no Freya consiste em três etapas. Inicialmente, os modelos ML são aplicados a dispositivos SG individuais. Posteriormente, um modelo de ML empilhado consolida essas previsões no nível do dispositivo para prever o evento geral da rede – finalmente, ocorre o processo de inferências através do OntoFreya, a ontologia proposta nesta tese. OntoFreya classifica eventos de rede e equipamentos em conformidade com regulamentos e padrões regulatórios de concessionárias de energia, permitindo manobras proativas para mitigar possíveis problemas. A validação do modelo usa dados do mundo real de alimentadores de distribuição, reguladores de tensão, religadores e vários cenários aplicados, demonstrando a capacidade do modelo Freya. O modelo Freya de redes de distribuição obteve uma precisão de 99,73%, recall de 99,75% e F1-Score de 99,73%, em comparação modelos comumente usados nesse tipo de tarefa alcançaram uma precisão de 83,36%, recall de 82,91% e F1-Score de 83,36%, demonstrando a superioridade do modelo Freya em termos de métricas na detecção de eventos.

Palavras-chave: Smart Grid. Aprendizado de Máquina. Computação Ubíqua. Ontologia.

ABSTRACT

In the Smart Grids (SGs) context, energy utilities manage extensive data volumes to monitor and optimize distribution networks. Key parameters include voltage, current, and power levels, as well as fault indicators like short-circuit currents and voltage sags. Detecting and predicting technical failures is for preventing power shortages that affect residential and industrial consumers. The process of predicting future conditions of the SG, is essential for identifying potential technical failures. SGs feature hierarchically distributed equipment across large geographical areas, influencing the overall network based on their hierarchical importance and inevice-level predictions to forecast the network's overall state—finally, the process of inferences through OntoFreya, the ontology proposed in this thesis. OntoFreya classifiedividual historical contexts. Continuous communication between these devices and monitoring centers is vital for effective SG operation. Integrating concepts such as Edge Computing (EC), the Internet of Things (IoT), and Machine Learning (ML) enhances event prediction and operational efficiency in SGs.This thesis introduces the Freya model, an intelligent computational framework designed for event prediction in SGs, focusing on energy distribution. Freya's scientific contribution lies in event prediction at both the equipment and network levels. Comparative analysis shows that Freya uniquely addresses three aspects: (1) operation of remote equipment, (2) context-aware on SGs, and (3) hierarchical importance within the network. These aspects serve as inputs for predictive modeling.Event prediction in Freya consists of three steps. Initially, ML models are applied to individual SG devices. Subsequently, a stacked ML model consolidates these ds network and equipment events in compliance with energy utility regulations and regulatory standards, enabling proactive maneuvers to mitigate potential issues. The model's validation uses real-world data from distribution feeders, voltage regulators, reclosers, and various applied scenarios, demonstrating the capability of the Freya model. The Freya model for distribution networks achieved an accuracy of 99.73%, recall of 99.75%, and F1-Score of 99.73%, compared to commonly used models in this type of task, which reached an accuracy of 83.36%, recall of 82.91%, and F1-Score of 83.36%, demonstrating the superiority of the Freya model in terms of event prediction metrics.

Keywords: Smart Grid. Machine Learning. Ubiquitous Computing. Ontology.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	Artificial Intelligence
ANEEL	National Agency of Electric Energy
ANN	Artificial Neural Network
BEMS	Building Energy Management Systems
CNN	Convolutional Neural Network
DMS	Distributed Management System
DTR	Decision Tree Regression
EC	Edge Computing
FQ	Focused Question
GQ	General Question
IC	Inclusion Criteria
IoT	Internet of Things
LSTM	Long Short-Term Memory
ML	Machine Learning
MLSTM	Multiplicative Long Short-Term Memory
MVC	Model View Controller
NN	Neural Network
PMU	Phase Measurement Units
PSAL	Power System Automation Language
PU	Per-unit System
RBF	Radial Basis Function
RC	Rejection Criteria
RF	Random Forest
RMSE	Root Mean Square Erro
SG	Smart Grid
SPA	Single Page App
SQ	Statistical Question
SS	Snowball Sampling

LIST OF SYMBOLS

°C	Degrees Celsius
V	Voltage (Volts)
I	Current (Amperes)
P	Power (Watts)
Q	Reactive Power (Volt-Amperes Reactive)
ω	Angular Frequency (Radians per Second)
f	Frequency (Hertz)
R	Resistance (Ohms)
X	Reactance (Ohms)
Z	Impedance (Ohms)
θ	Phase Angle (Degrees or Radians)
kW	Kilowatts
kWh	Kilowatt-hours
ϕ	Power Factor
H	Henry (Inductance)
F	Farad (Capacitance)
Δ	Delta (Change in a quantity)
λ	Wavelength (Meters)
τ	Time Constant (Seconds)
ϵ	Permittivity (Farads per Meter)
μ	Permeability (Henries per Meter)

CONTENTS

1 INTRODUCTION

Energy has been a basic human need in all available forms, enabling sustenance and development. Given this, humankind has always sought energy sources to supply this demand. In this context, electric energy is one of the most viable solutions. Due to the versatility of production, transformation into other energy sources, and use, electric energy has become a strategic resource for the socio-economic development of any region or nation. After decades of development in the generation, transmission, and distribution, the actual focus falls on the reliability and power quality of the electric energy(LEE; YUAN; WANG, 2022).

The Smart Grid (SG) advent allowed more assertive methods for electricity generation and distribution (ZHENG et al., 2021). Intelligent controls and data analysis enable the detection of deviations and increase the system's quality. New developments focused on data from electrical distribution systems and Artificial Intelligence techniques allowing the prediction of electrical metrics and problem detection regarding an electrical grid (NTI et al., 2021). Sensors in power distribution equipment allow monitoring of specific conditions in the SG. Predicting these conditions and triggering actions for a possible adverse condition is an example of an SG intelligent control (TIWARI et al., 2022).

1.1 Motivation

The Freya model represents an approach to addressing the challenges faced in power distribution networks. In these networks, various equipment or entities operate under different states, which refer to the current operational contexts of the equipment. These states can lead to events, which are sudden changes applied to these states. An event can transform a bad state into a good state, or conversely, it can change a good state into a bad state. For instance, an event might be triggered when the data from an entity changes abruptly, such as a shift of one or two standard deviations from the norm, indicating a change in the operational status that requires attention. Events can thus be beneficial or detrimental depending on the context and the nature of the change they bring about (YADAV; PAL; SAINI, 2023).

Electricity shortages have far-reaching impacts on humanity, affecting social and economic aspects of life (KHAN et al., 2021). Problems in power distribution networks contribute to these shortages, directly disrupting productive societies. Therefore, conducting a study to mitigate energy problems and optimize power distribution is essential. This study utilizes the concept of Smart Grids (SG) to collect data, allowing for comprehensive analysis and optimization of power distribution.

The advancement of technologies aimed at SG systems has enhanced decision-making processes for remote-controlled equipment, producing vast amounts of data (ZAINAB et al., 2021a). However, the data generated by these systems or their surrounding environments require sophisticated interpretation strategies to improve system reliability.

SG equipment context histories play a role in enhancing event prediction capabilities within the grid. Context histories refer to the chronological record of operational states and environmental factors related to specific pieces of equipment. These histories include data such as voltage levels, current, power output, temperature, and external conditions like weather. Maintaining detailed context histories makes it possible to identify patterns and trends that precede events (HAUER et al., 2021).

For example, a voltage regulator might show a pattern of minor fluctuations before a fault occurs. By analyzing the context history, predictive models can learn these patterns and trigger preemptive alerts before a fault impacts the grid. Similarly, the context history of a recloser might reveal specific conditions under which it frequently operates, helping to predict future operational events under similar conditions. This historical data enables more accurate and timely predictions, thus preventing outages and improving grid reliability (ZAMAN et al., 2024).

Context histories also support adaptive learning in Machine Learning models for event prediction. As new data is continuously collected, the models can be retrained to recognize new patterns and anomalies, enhancing their predictive accuracy over time. Including contextual information, such as environmental conditions, ensures that the models are not just reacting to changes but anticipating them based on comprehensive historical data. This proactive approach to event prediction and management is essential for maintaining the stability and efficiency of Smart Grids (ALHATHLOUL; MISHRA; KHAN, 2024).

To effectively map and manage these factors, ubiquitous computing emerges as a promising solution. Ubiquitous computing emphasizes context awareness, which helps handle the information related to entities within the power distribution network (BARBOSA, 2015; DUAN et al., 2023). Contextual information includes voltage, current, power, temperature, and humidity data pertinent to equipment like voltage regulators and reclosers. These contexts have specific attributes such as identity (unique identification), status, date, and time, all of which contribute to establishing a chronological order of events. This chronological order forms what is known as context histories (RENTZ; HECKLER; BARBOSA, 2023; LIMA et al., 2022).

In power distribution networks, equipment states can lead to events. For example, a sudden change in voltage or current by one or two standard deviations from the norm can signify a potential issue that needs to be addressed. Predicting these events is strategic because they can lead to electricity shortages, which in turn can reduce economic activity and create challenges for energy companies(MUQEET et al., 2023).

The Freya model is designed to predict these events, thereby helping to prevent energy shortages and enhance the overall reliability of power distribution networks. By integrating concepts from Edge Computing (EC), the Internet of Things (IoT), and Machine Learning (ML), Freya provides a framework for state estimation in SGs. The model considers the hierarchical importance of equipment and their historical contexts, enabling more accurate and reliable predictions of events. In summary, the Freya model offers a comprehensive solution to the challenges faced by power distribution networks by leveraging advanced technologies to predict and manage events. This capability is essential for maintaining power distribution's reliability and efficiency, ultimately supporting social and economic stability (SVENSSON et al., 2023).

1.2 Research Question and Scientific Contribution

SG systems, in general, can work in wide-ranging areas across cities and sometimes even states (BUTT; ZULQARNAIN; BUTT, 2021). Information systems need to support SGs regardless of network reach. Systems that operate in a large-scale area use resources of a concept called edge computing (EC). EC allows processing information and reducing network latency far from a server or an operation center (CAO et al., 2020). Context awareness would make the system aware of the different ambients in this operation area. Context awareness and EC can be a way to mitigate the problems faced by the systems that support SG.

Analyzing the SG system's data helps maintain a grid's reliability and power quality. Communication problems can cause monetary losses or even accidents. Another essential factor refers to the hierarchy in an SG once each equipment influences the next one. Architecturally similar to the Internet, the Smart Grid is hierarchical and has clear demarcation points. Power utilities perform generation, and interstate power distribution, equivalent to the backbone of an internet service provider (WU et al., 2022). Other essential factors have mutual influences between the different operating zones of equipment with the same functions, such as voltage control, reclosers, and capacitor banks, within the same network. The analysis of these data can contribute to the definition of the zones of influence of each equipment and even detect or recommend a hierarchical operation between them.

SGs are dependent on communication for their correct functioning. In case of loss of information due to failure of communication with SG equipments, problems in energy distribution may occur (PAL; SHANKAR, 2022). These problems would not be detected by a central and could cause problems in the subsequent equipments of the network. Some equipment may be responsible for connecting a network to other networks, thus having greater hierarchical importance than other equipment.

The Freya model's intelligence lies in its ability to predict and detect events within the Smart Grid (SG) network. Freya aims to identify associations between the contexts of SG entities and their electrical metrics. The Freya model's scientific contribution involves detecting events at the network's edges and across the entire network. It considers the hierarchical importance of each piece of equipment based on its connectivity and role in supplying power to consumers. Equipment with greater connectivity and consumer responsibility is prioritized in the event prediction process.

Additionally, most SG systems in power distribution networks operate through centralized approaches, relying on a central server to process data and make decisions. This centralization can create bottlenecks and latency issues, particularly as the network complexity increases. Decentralized methods, while less common, are gaining attention for their potential to enhance scalability and efficiency by distributing the computational load across the network. Recent advancements suggest that integrating edge computing with SG could address these centralization challenges effectively (NIU et al., 2023; SATYANARAYANAN, 2017).

The research question guiding this thesis is: *"How can a computational model be developed to evaluate monitoring data in a Smart Grid to predict network events, considering the operational hierarchy among different pieces of equipment?"* This question reflects the core aim of the Freya model, which is to enhance the reliability and performance of power distribution networks through advanced predictive analytics and hierarchical event management.

1.3 Objectives

This thesis aims to create a computational model called Freya for event prediction's in Smart Grids. In order to achieve this objective, the following specific objectives are:

- Perform a literature review of computing techniques that support Smart Grids;
- Create an edge-computing component to perform event prediction in power distribution, according to the equipment context histories and at the edge of the Smart Grid;
- Propose a model for event prediction based on the energy flow and context of equipment within a power distribution Smart Grid;
- Build an ontology for power metrics classification according to the event of the equipment on the edge of the grid;
- Evaluate the Freya model through operational scenarios.

1.4 Methodology

A review of the bibliography on the SG information systems theme helped to understand the gaps in how researchers have approached this domain. An initial analysis verified that SG systems use the Internet of Things (IoT) resources, Data Analysis, Edge Computing, and Context Awareness. The bibliography review consisted of a systematic mapping.

The systematic mapping study conducted a comprehensive literature review of research investigating how IoT, EC, data analysis, and context awareness can aid in implementing SG systems (JAMES; RANDALL; HADDAWAY, 2016; COOPER, 2016; CHRISTOU; PARMAXI; ZAPHIRIS, 2024). The primary goal of this review was to identify evidence and trends in collections of literary works related to the model proposed in this thesis. Based on the guidelines of Pinciroli, Justo, and Forradellas (2020), the systematic mapping applied the following steps: Research questions, Research process, and Criteria for filtering results.

This literature review employed a systematic mapping methodology encompassing 15,081 works related to four SG technologies. After the filtering process, 37 works remained. The mapping verified the most used trends and technologies and analyzed data from SG. The study answered one general question (GQ), six focused questions (FQ), and two statistical questions (SQ). Finally, from these 37 works, 21 were selected to compare with the Freya model due to their objectives and characteristics as related works. Finally, five works were added after a new round of article searches to update the literature review, totaling 26 articles.

A final selection of related works and an analysis resulted in a set of gaps to address in this thesis. After identifying the gaps left by the related works, it was necessary to develop an alternative based on a computational model to event prediction based on the context histories of SG. Before developing the model, identifying possible usage scenarios helps develop the services more adequately. With the scenarios identified, a computational model design fulfills the identified gaps in related works. The same scenarios help to evaluate the proposed model.

Power distribution data should be collected on each piece of equipment. This collection considers the power metrics of the SG equipment and context data that influences the SG power metrics. Then, the data goes through a pre-treatment to allow event predictions of power metrics from this equipment. All this data(current, voltage, weather, period of the day) results in a dataset. Another approach addressed in the dataset and added as a feature is the period of the day; since an event could have different values in the morning or at night, this feature is included in the dataset to mitigate this issue. Still, to predict events, an algorithm to detect sudden changes in this data (features) shows when a sudden change occurs based on the difference between the previous and the following collections. When this difference between a collection of features is more than two standard deviations, the target column "event"receives the value ’true,’ which is where an event occurred; otherwise, the column receives false (DUEÑAS et al., 2024).

This approach labels the dataset and indicates when an event occurred. With all of this set, the equipment has a trained model based on its context histories. Once trained, it is ready to predict events. After that, one event prediction occurs. Since each piece of equipment in the SG can have different environments and datasets, it has his own event prediction model. Then, if a prediction detects an event, it triggers the start of predictions, considering the current energy flow and the hierarchy of each equipment in this flow within the distribution networks.

After an event prediction in a single piece of equipment, the nearest equipment (based on the hierarchical energy flow) performs a stacked event prediction. This process involves the previous and subsequent equipment transferring trained models to the next one, enhancing their ability to identify event patterns more effectively. This method, known as transfer learning, allows each piece of equipment to leverage the knowledge gained by its neighbors. The transfer learning process halts once the initiating equipment detects that the event has concluded, indicated by the power metrics returning to within two standard deviations. To avoid multiple events with the same equipment, the equipment in the event receives a flag showing that the equipment is unavailable for another event (LIU et al., 2024).

These predictions are sent to the orchestrator, who serves as the centralized component of the Freya model and manages all the equipment. The orchestrator stores the predictions and uses them to perform a network prediction up to 4 steps ahead with the probability of events occurring in each piece of equipment involved in the current event. This data is then input into an ontology called OntoFreya. OntoFreya performs inferences based on the energy company's rules to identify possible root causes of the event. All event information is stored in a database for further analysis, ensuring the system can continually improve its predictive capabilities and provide insights for future events.

In order to complete this study, the support of the CEEE Special Equipment project, which has a partnership with CEEE Equatorial Power Utility [1] and Certaja Energy Company [2] was essential. These companies provide data and parameters related to a distribution network that helped to analyze and develop the model proposed in this thesis.

The evaluation of the Freya model utilized operational data. Based on this data and input from the energy companies, it was possible to create two regular operation scenarios with events involving three pieces of equipment for each scenario. Additionally, one scenario simulates regular operations with an event involving five pieces of equipment. Furthermore, two more scenarios represent temporary energy flow hierarchies, each involving three different pieces of equipment. These temporary energy flow scenarios occur when some equipment is under maintenance or unavailable, necessitating flow transfer to a non-regular path. This comprehensive evaluation ensures that the Freya model can handle various operational conditions and predict events across different scenarios.

1.5 Thesis Organization

This thesis is divided into seven chapters, including this introductory. The second chapter addresses the basic concepts used in this thesis, the third chapter raises the literature review carried out as well as related works. The fourth and fifth chapters address the proposed model and a proposed ontology OntoFreya, respectively. Chapter six presents the evaluation of the model. Finally, chapter seven presents the final considerations of the thesis.

[1] https://ceee.equatorialenergia.com.br/
[2] https://www.certaja.com.br/energia/

2 BACKGROUND

This chapter describes the concepts and foundations used to design this thesis. Section 2.1 presents the intelligent features of electrical grids. Section 2.2 introduces the concepts regarding the ubiquity of computing systems. Section 2.3 exposes the benefits of systems being aware of the context. Section 2.4 approaches EC technologies, IoT, Multiagent systems, and distributed analysis data. Finally, Section 2.5 explains the concept of Stacking Ensemble

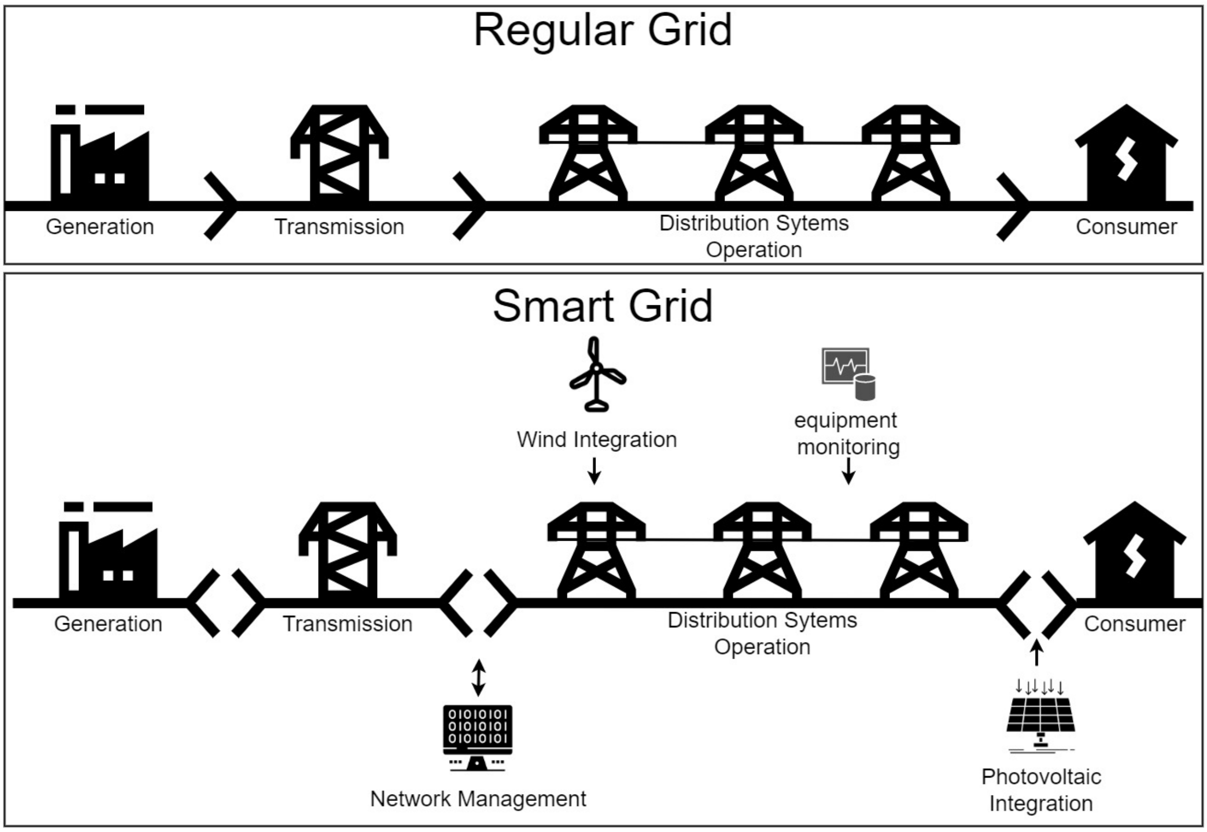
2.1 Smart Grids

The search for increasingly robust, efficient, and integrated systems is one of the most responsible for technological development. Similarly, when electric power systems are under discussion, SGs are a relevant topic. The regular energy grids are becoming obsolete due to the incessantly growing demand for energy and its infrastructure limitations (XU et al., 2022). SGs, also known as intelligent energy systems and networks, holds the most promise for the energy sector in the present and near future. Through the broad vision that SGs provide and act, Figure 1 illustrates the differences between an SG and a regular grid. On the regular grid, energy is generated, transmitted, and distributed to consumers through a distribution system. In SG after generation, the transmission has network management, allowing communication with specific points on the grid. In the distribution system, it is possible to monitor the equipment involved and integrate types of renewable energy such as wind. Finally, the consumer can also integrate photovoltaic energy.

Smart grids propose an integration of the traditional electricity grid with sensors, automated field devices, smart meters, communication technologies, and information technology (BHATTACHARYA et al., 2022). This proposal of automation, production, transmission, distribution, and use of electric energy stands out for optimizing all the most necessary characteristics of these structures while still helping to overcome certain limitations. The main benefits of SGs stand out:

- More efficient transmission of electricity;
- Cost-effective management and operations services;
- Greater control and supervision over the grid infrastructure;
- Greater integration with other energy sources;
- Helps to reduce energy demand;
- Consumers become more conscious and active under energy usage;
- Increased security.

Figure 1: Smart Grid and Regular Grid Comparison.



Source: Adapted from Bhattacharya et al. (2022)

With the SGs, consumers have more quality in electricity transmission and more accessibility to the services offered by a power utility. The SG systems have real-time monitoring from the initial to the final stage of SG infrastructure. The integration of SGs systems and analysis of this system's data leads to statistical and predictive analyses that help optimize energy balance and reduce technical failures (JAIN; BHULLAR, 2022).

Unlike traditional power grids, an essential difference compared to SGs is the ability to communicate in two ways, making it possible to produce power, transmit, and receive information back between all connection points in the grid. In the traditional grid, there is no feedback for the information sent along the path between power generation and transmission, not even between the power distribution station and the end user. However, two-way communication occurs in an SG and integrates with other energy sources and communication technologies (REMIGIO-CARMONA et al., 2022).

The communication technologies used in SGs can help power utilities and energy distributors remain stable and provide quality service. The power utilities comply with the National Electric Energy Agency (ANEEL) regulations in Brazil and define the threshold values of the electric metrics. Thus, decreasing maximum values over the years, making the goals for quality control even more rigid and continuity of electricity supply, demanding from all power utilities a continuous improvement in their service provision, operational efficiency, and cost control

(CARVALHO et al., 2020).

The decentralization of distribution, market competitiveness, and the increasing collection of regulatory bodies in the electricity sector have provided changes, causing utilities to develop new, more efficient, and safer methods for analyzing, planning, and operating electrical energy systems. These changes force electrical companies to invest in new technologies that seek faults in the distribution network, thus ensuring a quality service (XU et al., 2022).

2.2 Ubiquitous Computing

According to Weiser (1991), ubiquitous computing consists of computing devices distributed in an environment and communicating with each other, making the computer familiar and used daily. Satyanarayana (2001) considered Weiser's vision as an evolution of research focused on distributed systems and mobile computing. Thus, ubiquitous computing is an environment with devices or systems that adapt to different environments by doing so transparently.

Weiser's vision has driven the evolution of technology to a great extent. Ubiquitous computing has changed a lot over the years and has proven to be a field of reference in technology, helping to develop new revolutions such as the internet of things (VRITTI et al., 2024). Perhaps ubiquitous computing is a field of particular interest because it blends well with many application domains. The SG domain is an area that increasingly needs intelligent and efficient solutions to deal with energy reliability.

For Barbosa, Filippetto, Lima (2020), application areas such as electrical engineering, commerce, education, and games have much to gain from ubiquitous computing. One is that adopting ubiquitous computing techniques can generate impacts similar to those generated with the advent of the internet in the most diverse application areas.

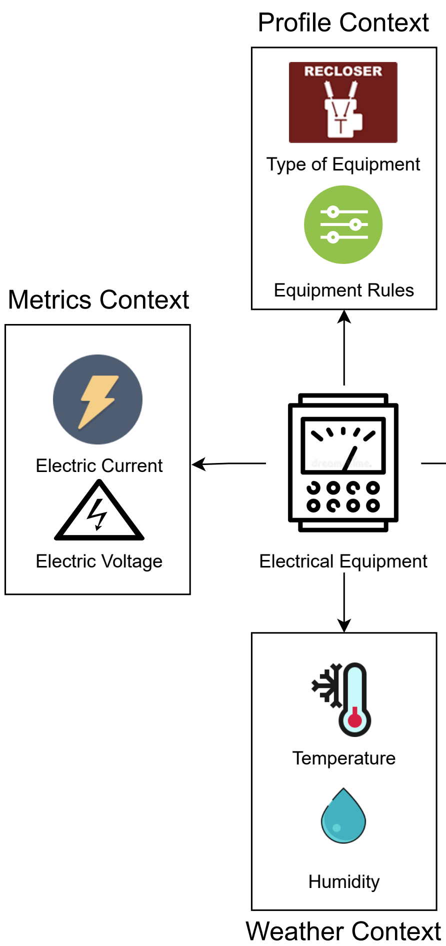
Ubiquitous computing has become more present in people's daily lives. This popularization occurred due to the low cost of microelectronic devices and increased wireless communication connections. The ability to collect data on the current situation of the environment and the subsequent analysis of this data to adapt its functioning is called context-sensitivity, an essential concept for ubiquitous computing (THAM; VERHULSDONCK, 2023).

2.3 Context and Context Histories

In ubiquitous computing, understanding an entity's environment is one of the essential pillars for the operation of ubiquitous applications (DEY; ABOWD; SALBER, 2001). The context refers to any system-relevant information. Examples of context can be the time when a particular event occurred or even the profile of an entity. In this way, context-sensitive computer systems can use this information to perceive and react to changes in the environment where the entity is situated, adapting and improving the system (MARTINI et al., 2021).

Figure 2 shows the possibilities of information present in the context of an entity. This information may include the metrics context, such as electric current or voltage. The profile context regards an entity's specific information, such as its type or the equipment's operating rules. The Daily Context includes information about the entity's status at a specific time and date, such as the inlet/outlet of energy transmitted by that equipment. Finally, the weather context where the entity is inserted, such as temperature or relative humidity. All this information, if correlated, can allow a better understanding of an entity condition at that moment. The storage of this information over a period forms histories called context histories (HECKLER et al., 2022).

Figure 2: Entity information in a context



Source: Prepared by the author.

2.4 Smart Grids Distributed Systems

Smart Grid systems generally work in a distributed way. The SG systems work this way due to the equipment being geographically distributed. One of the concepts to be implemented in SG systems is Edge/Fog computing to support this distributed operation. The application of event detection techniques in a distributed environment is an example of one of the possibilities of SG. With that in mind, the operation of SG is similar in some points to that of smart environments.

According to Hajjaji et al. (2021), a smart environment is the convergence of computing areas, ubiquitous computing, Artificial Intelligence (AI), and Internet of Things (IoT). This environment is discreet, interconnected, adaptable, dynamic, integrated, and smart (WADI et al., 2024). Smart environments have the advantage of the equipment used in the development, which is generally low cost and easy to handle, thus being easy to exist in large numbers. Verifying the entity context location and entity interactions with the environment is crucial. Smart environments can have different sizes, a concept addressed when considering large environments is Fog/Edge Computing.

Samann, Abdulazeez, Askar (2021) defined Fog Computing as a layered model that enables ubiquitous access to scalable and shared computing resources. The Fog layer consists of context-aware virtual or physical fog nodes that serve applications and services, considering latency limitations. A fog system can work through clusters.

For Laghari, Jumani, Laghari (2021), the growing concern about problems related to massive real-time data processing and bandwidth limits led to the birth of Fog Computing, which works intensively, but not exclusively, along with edge. Fog Computing is a paradigm that realizes distributed computing, networking, and storage services, serving as an extension of Cloud Computing connected to edge devices.

Fog Computing's primary functions are filtering and aggregating data for Cloud data centers. It also applies intelligence to the end devices. Fog differs from Cloud by being highly interconnected with end devices (IoT), enabling geographic distribution and supporting mobility.

Fog Computing differs from Edge Computing because of the usage of public internet to allow the connection of local systems online, allowing management, communication, and control, directly with IoT devices or through a Fog server. Fog computing takes place close to the edge of the network, integrating location awareness, low latency, and quality of service into streaming and real-time applications (BOUQUET et al., 2024).

The diffusion of IoT devices has created many intelligent environments (SOURI et al., 2022). These devices are data generators, so vast amounts of data generation occurs at the network's edge, and knowledge extraction should happen from this. Cloud might seem like the most convenient solution for IoT analytics, with high volume, speed, and heterogeneity. However, transmitting all data to the Cloud comes up against limited bandwidth, or high latency (KSERAWI; AL-MARRI; MALLUHI, 2022).

Applications with communication restrictions require distributed analysis algorithms that work directly on the devices. This distributed analysis generates the data, such as sensors and embedded devices, or forwards to previous layers, like the Fog Computing layer, and at the edge of the network (LAGHARI; JUMANI; LAGHARI, 2021). In scenarios where the analyzes calculated on this data can be relevant only for a short period and in specific locations, there is no need for extensive data movements, avoiding wasted bandwidth. This approach targets specific scenarios but could be extended to Smart Grids (KSERAWI; AL-MARRI; MALLUHI, 2022).

Still taking into account the distributed nature of the SG, intelligent agents might perform actions in all system layers. In order to decentralize the control of the network, agents are suitable options to detect deviations and take actions that involve more than one layer at the same time (OLATUNDE et al., 2024). Multiagent systems facilitate observing behaviors through agents present in a systems architecture. In this structure, an organization allows a better approximation of the context domain.

Agents are computer systems in an environment capable of being autonomous in their attitudes, aiming to achieve goals (SHOBOLE; WADI, 2021). Agents can also be defined as components of a system, which perform simultaneous activities, and can reason and adapt to an environment. Saxena, Farag, El-Taweel (2021) state that the emergence of an organization is visible, where the agent model observes the interactions between the system elements.

Agent technology has seen growth in all fields recently, especially SGs (ZHENG et al., 2021). Agent-oriented software engineering is a phenomenon used in the production of distributed systems. The special features of the agent, such as intelligence and autonomy, reduce operating costs and perform automatic functions in some systems, such as SG, which implements the multiagent technology. The use of new communication technologies, distributed systems, and intelligent agents is also considered a new phenomenon in electrical engineering (ZAINAB et al., 2021a).

2.5 Stacking Ensemble

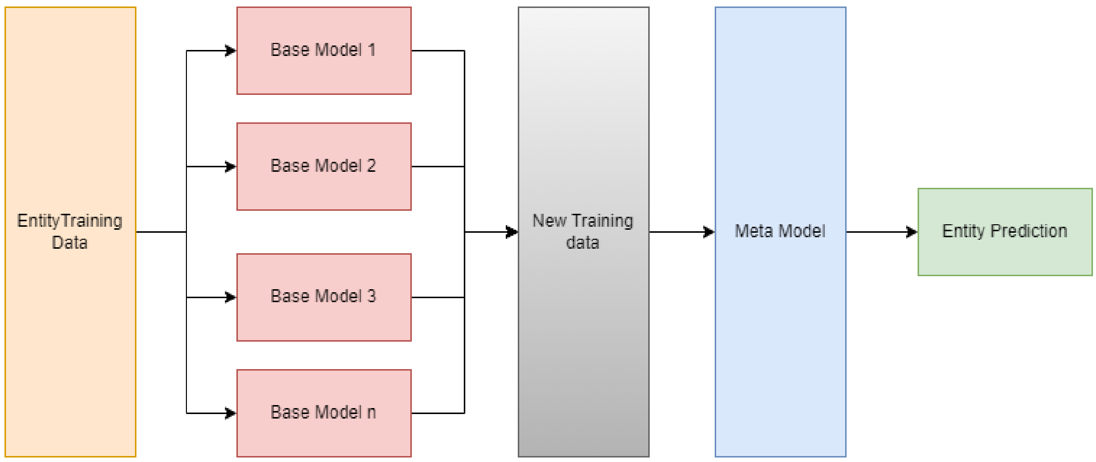
Stacking ensemble displayed in Figure 3, also known as stacked generalization, is a ML technique that combines multiple predictive models to improve the accuracy of predictions. The primary purpose of a stacking ensemble is to capitalize on the strengths and minimize the weaknesses of individual models, thereby creating a predictive system (RAJADURAI; GANDHI, 2022).

The technique of a stacking ensemble involves multiple layers of models. The first layer consists of a diverse set of base models, each trained on the same dataset but using different algorithms. These base models can include a variety of ML techniques such as decision trees, support vector machines, neural networks, and more. The key is that each model brings a unique perspective to the data, capturing different patterns and relationships (ABDELLATIF et al., 2024).

After training the base model, a second layer uses the base model prediction as inputs, the second layer model is also called as meta-model or blender. The role of this meta-model is learning how to combine the predictions of the base models to make a final prediction. The meta-model understands which base models perform well in certain conditions and how their predictions can be optimally blended (LAZZARINI; TIANFIELD; CHARISSIS, 2023).

The stacking ensemble approach is efficient because it not only leverages the predictive

Figure 3: Stacking Ensemble



Source: Prepared by the author.

power of each model but also learns how to integrate these predictions best. This integration can lead to a improvement in prediction accuracy compared to using any single model alone. It is instrumental in complex problems where no single model can capture all the nuances of the data (AGGARWAL et al., 2023).

This work introduces an innovative approach to event detection in power distribution networks through a multi-layer stacked ensemble. This method is distinct in its structure and function, designed to harness both the individual patterns of each entity within the network and the overarching patterns of the network as a whole.

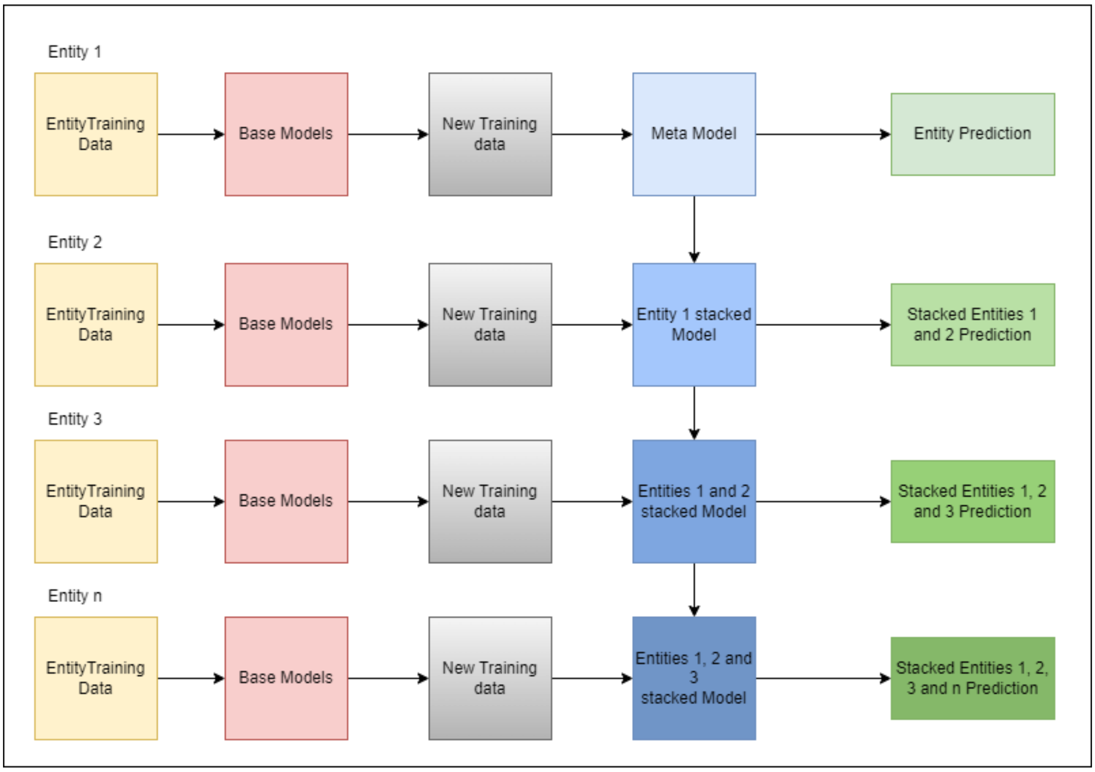
In this approach, each entity within the power distribution network retains its own set of base models. These base models are tailored to the specific characteristics and patterns of the respective entity, ensuring that the unique aspects of each entity are accurately captured and analyzed. This local level of analysis allows for the detection and understanding of events specific to individual entities.

The novel aspect of this approach lies in integrating these base models into the broader network context. Instead of simply aggregating the predictions of the base models, this paper proposes updating the meta-model with a complete stacked model from a previous entity in the network. This process means that each meta-model knows its patterns and the patterns of the previous entity in the network. Figure 4 shows this approach.

Once the meta model concludes the training process a local prediction can be performed. In parallel this current meta model is sent from entity 1 to entity 2 aggregating knowledge from both entities and allowing to send it again to the next entities in the network (entities 3 and n). Since each entity only needs to know the following entity in the network, the flow of stacking models can increase in multiple types of entities.

This process repeats across the network, achieving a multi-layer stacking effect, where each entity is one of the layers of the ML model. Each subsequent meta-model becomes increasingly sophisticated, integrating the knowledge and patterns from all previous entities. This

Figure 4: Proposed Multi-Layer Stacking Ensemble



Source: Prepared by the author.

progressive accumulation of insights allows for a comprehensive understanding of the network, capturing the specific patterns of individual entities and the complex interdependencies within the network. The result is a ML algorithm that maintains the granularity of local data analysis while benefiting from the broader context and insights of the entire network. This approach ensures that the nuances and specifics of individual entities are not lost, yet the collective knowledge and patterns of the network enhance the overall predictive power.

2.6 Chapter Considerations

This chapter introduced the basic concepts and terms used in this research. The first section of this chapter addressed the issue of SG. The second section approaches ubiquitous computing. The third section of the chapter addressed the context and the context histories since the information from the entity is relevant to event detections in the SG. Then issues related to Smart Grids Distributed Systems were addressed. The next chapter addresses the bibliographic mapping of works related to the proposed model, addressing research questions, search methodology, mapping, and gaps observed by the selected works.

3 RELATED WORKS

This chapter presents the methodology, results, and a discussion of the related works of this thesis. The literature review presented here is also a result of a published systematic mapping article (ARANDA et al., 2022). The systematic mapping study conducted a literature review of research investigating how IoT, EC, data analysis, and context awareness can aid in implementing SG systems (JAMES; RANDALL; HADDAWAY, 2016; COOPER, 2016; CHRISTOU; PARMAXI; ZAPHIRIS, 2024).

The main objective of such a review is to identify evidence and trends in collections of literary works related to the model proposed in this thesis. Based on the guidelines of Pinciroli, Justo, and Forradellas (2020), the systematic mapping applied the following steps: Research questions, Research process, and Criteria for filtering results.

This literature review used a systematic mapping methodology to encompass 15081 works related to four SGs technologies. With 37 works after the filtering process, this review revealed that most papers use one to three approaches, while only two use all four technologies.

The results also indicate that EC has been extensively used in SG solutions, with 22 selected studies. Distinctively, only 9 works use context awareness, which may indicate a path for future developments in SG. The study also allowed the learning of 7 lessons that are presented in this chapter. The mapping verified the most used trends and technologies and analyzed data from SG. The study answered one general questions (GQ), six focused questions (FQ) and two statistical questions (SQ).

For the related works discussion, the selection of 21 from 37 related works compared with the Freya model is based on the study objectives and features that allow comparison. Ultimately, five publications were included after subsequent article searches to ensure the literature review remained current, totalizing 26 articles.

A careful examination and analysis of a range of comparable works revealed a set of gaps that this thesis aims to address. Identifying deficiencies in existing research necessitated creating a new approach using a computational model for predicting events based on the historical background of SG.

Before constructing the model, potential usage scenarios were identified to design the services more effectively. The computational model design addresses the gaps in related publications by considering the identified scenarios and facilitating the assessment of the suggested model.

The remaining section of this chapter is organized as follows. Section 3.1 describes other systematic mappings related to the mapping presented in this chapter, section 3.2 details the research method used in this literature review, section 3.3 presents the systematic mapping results, answering the research questions, section 3.4 discusses the findings, section, section 3.5 presents the updated search the systematic mapping, section 3.6 shows the works related to the proposal of this thesis, finally, section 3.7 provides the chapter considerations.

3.1 Related Mappings

Systematic mapping studies applied on different research areas (GONCALES et al., 2014; VIANNA; BARBOSA, 2017; DIAS; BARBOSA; VIANNA, 2018; DALMINA; BARBOSA; VIANNA, 2019) have also applied this approach (PETERSEN et al., 2008; PETERSEN; VAKKALANKA; KUZNIARZ, 2015; COOPER, 2016). During the search stage, literature review articles of SGs returned by the search string were considered as related works for reviewing the literature within the same area of a domain that this thesis proposes. Ten reviews and surveys address EC, IoT, data analysis, and context-aware concepts applied to SGs (DAVOODY-BENI et al., 2019; ZHEN et al., 2019; SHI et al., 2020; IBRAHIM; DONG; YANG, 2020; CHENG; YU, 2019; MIMI; BEN MAISSA; TAMTAOUI, 2023; EL MOKADEM; BEN MAISSA; EL AKKAOUI, 2023; LAMPROPOULOS, 2023; KRIVOHLAVA; CHREN; ROSSI, 2022; SAKHNINI et al., 2021). Three studies only consider the effects on the SG after an event occurs (DAVOODY-BENI et al., 2019; ZHEN et al., 2019; IBRAHIM; DONG; YANG, 2020). Using artificial intelligence techniques such as Machine Learning (ML), these three studies do not consider papers that analyze data in real-time, partially meeting the requirements for data analysis. Other studies (ZHEN et al., 2019; IBRAHIM; DONG; YANG, 2020; MIMI; BEN MAISSA; TAMTAOUI, 2023; EL MOKADEM; BEN MAISSA; EL AKKAOUI, 2023; LAMPROPOULOS, 2023; KRIVOHLAVA; CHREN; ROSSI, 2022; SAKHNINI et al., 2021) consider EC partially since the review studies do not process nor analyze data at the edges, using EC only to collect and transfer data. Conversely, the present work reviewed papers considering data analysis, IoT, EC, and context-aware techniques applied to SG.

The work of Davody and Beni (2019) focused primarily on IoT and SG, analyzing their advantages, challenges, and practical solutions. The authors address aspects, like big data, expenditure reduction, and system security.

Zhen et al. (2019) summarized the key big data technologies and research ideas and discussed the problems faced by big data technologies in SG. The main problems refer to security and data management in SG, as well as latency in data acquisition.

Shi et al. (2020) published a survey on artificial intelligence techniques applied to SG. The authors concluded that these techniques can predict SG stability. Alternatively, communication problems may occur when evaluating the research models in an SG.

The survey of Ibrahim et al. (2020) demonstrated the increasing interest and expansion in the use of ML techniques in SG. According to the authors, some issues remain open and are worth further research, such as the high-performance data processing, and intelligent decision-making in large-scale complex multi-energy systems, lightweight ML, and EC.

The literature review performed by Cheng and Yu (2019) focuses on introducing and summarizing seven usual ML methods in the field of SG: reinforcement learning, deep learning, transfer learning, parallel learning, hybrid learning, adversarial learning, and ensemble learning.

The study of Mimi et al. (2023) investigated demand-side management in the Smart Grid by optimizing energy-related objectives like electricity costs and peak-to-average energy ratios to prevent large-scale network failures. The analysis of 104 studies out of 684 reveals a dominance of genetic algorithms, insufficient focus on renewable energy, a bias towards residential buildings, and a preference for real-time pricing schemes

El Mokadem et al. (2023) presented a study about Federated Machine Learning (Fed ML), which is a distributed Machine Learning technique that trains a global model using local data from clients without transmitting it, ensuring data confidentiality. This method is strategic for data-sensitive applications like IoT and Smart Grids. The study provides a structured overview of the field, answering specific research questions and offering potential recommendations for future research.

Lampropoulos (2023) showed a bibliometric and mapping study investigating the application of artificial intelligence in Smart Grids and its evolution from 2005 to 2022. The study addresses ten research questions, analyzing 1,926 articles from Scopus and Web of Science. It includes descriptive statistics and annual scientific production and identifies the most influential authors, articles, journals, affiliations, and countries. The findings highlight the role of artificial intelligence in digitalizing the power sector to achieve sustainable development and sustainable development goals, discussing results and suggesting future research directions.

Krivohlava et al. (2022) demonstrated a systematic literature review that examines 30 different faults and failures in Smart Grid (SG) infrastructure, highlighting their causes, impacts, detection techniques, and counter-measures. The study classifies and maps these faults and failures to the Smart Grid Reference Architecture Model (SGAM), providing a reference for practitioners and researchers focused on hardware and software dependability in SGs.

Sakhnini et al. paper (2021) explored the integration of sensors and communication technology in power systems, known as the Smart Grid, which enhances system functionality but also increases vulnerability to cyber-threats. It provides a bibliometric survey of research papers on the security aspects of IoT-aided Smart Grids, claiming to be the first of its kind. The analysis includes classifying journal articles by dates, authorship, and concepts. Additionally, the paper summarizes the types of cyber threats, the proposed security mechanisms, and the existing research gaps in Smart Grid security.

Furthermore, these works (DAVOODY-BENI et al., 2019; ZHEN et al., 2019; SHI et al., 2020; IBRAHIM; DONG; YANG, 2020; CHENG; YU, 2019; MIMI; BEN MAISSA; TAMTAOUI, 2023; EL MOKADEM; BEN MAISSA; EL AKKAOUI, 2023; LAMPROPOULOS, 2023; KRIVOHLAVA; CHREN; ROSSI, 2022; SAKHNINI et al., 2021) have discussed advantages, architectures, applications, and research issues. Table 1 shows a comparison between the related works approach. The comparison of the works in the SG domain considers if the works use the techniques of data analysis, IoT, EC, and Context-Awareness. The differential of the proposed systematic mapping is the literature analysis of works considering at least three of these technologies simultaneously.

Table 1: Comparison between same domain systematic mappings

Reference	Data Analysis	Internet of Things	Edge Computing	Context-Aware
Davody et al.(2019)	Partially	Yes	No	No
Zhen et al. (2019)	Partially	Yes	Partially	No
Shi et al. (2020)	Yes	No	No	Yes
Ibrahim et al. (2020)	Partially	Yes	Partially	No
Cheng et al. (2019)	Yes	No	No	Yes
Mimi et al. (2023)	Yes	No	Yes	No
El Mokadem et al. (2023)	Yes	Yes	No	No
Lampropoulos (2023)	Partially	No	No	Yes
Krivohlava et al. (2022)	Yes	Yes	Partially	No
Sakhnini et al. paper (2021)	Yes	Yes	No	No

Source: Prepared by the author.

3.2 Methodology

This section presents a systematic mapping study methodology for a literature review (BUDGEN et al., 2008; PETERSEN; VAKKALANKA; KUZNIARZ, 2015; COOPER, 2016) of research papers that investigated how IoT, EC, data analysis, and context awareness can support SGs. The main objective of such a review is to identify evidence and trends in the collections of literary works related to a topic of interest, reducing bias when single references are used. Based on the guidelines proposed by (PETERSEN; VAKKALANKA; KUZNIARZ, 2015), the systematic mapping followed the following steps:

- Elaboration of research questions.
- Elaboration of the search process.
- Definition of criteria for filtering results.

3.2.1 Research Questions

The research questions delineated the discovery of papers related to data analysis, IoT, context awareness, and EC applied in SGs. Hence, the study defined one General question (GQ), six Focused Questions (FQ), and two Statistical Questions (SQ). Table 2 presents the questions.

The GQ sought basic information regarding the technologies used in SGs. FQs explore quantitative details of the selected papers, such as the most common data analysis techniques or how many studies use IoT. Finally, SQs aimed to verify the publications' chronological data and type of venue.

Table 2: Research Questions

Reference	Question
General Questions	
GQ1	How data analysis and Internet of Things been used to support Edge Computing on Smart Grids?
Focused Questions	
FQ1	Which are the data analysis techniques applied to Edge-Computing in Smart Grid?
FQ2	Are there studies that consider Contexts, Context Histories and Context Prediction, according to the Dey's definition (DEY; ABOWD; SALBER, 2001)?
FQ3	Which are the adaptation strategies used to improve the data management in Edge-Computing applied to Smart Grid?
FQ4	How has the Internet of Things been used for Edge-Computing in Smart Grids?
FQ5	How the works used Big Data to support Edge-Computing in Smart Grids?
FQ6	How has the Machine Learning prediction's been used to support Edge Computing in Smart Grids?
Statistical Questions	
SQ1	What is the number of publications per type?
SQ2	How Many publications occurred per year?

Source: Prepared by the author.

3.2.2 Research Process

The study defined three steps for the research process: specify the search string, select databases, and find the results. The first step identified the main terms and their synonyms. The terms chosen were "Smart Grid, "data analysis and "edge computing"as primary terms, and "smart energy", "big data", "Machine Learning", "deep learning, "data analytics", "intelligent edge computing", "edge computing", "intelligent edge computing", "internet of thing"and "fog computing"as synonyms as displayed in Table 3.

Table 3: Definition of the Search String

Major Terms	Search Terms
Smart Grid	((Smart Grid OR smart energy) AND
Data Analysis	(data analysis, OR big data OR Machine Learning OR deep learning OR data analytics) AND
Edge Computing	(edge computing OR intelligent edge computing OR edge computing OR intelligent edge computing OR internet of thing OR fog computing))

Source: Prepared by the author.

The search string elaboration consists of the definition of the major terms and their synonyms. After defining the search string, the search process encompassed seven digital libraries: ACM, IEEE Xplore, Scopus, Google Scholar, Springer Link, Science Direct, and Wiley. The selection prioritized electrical and computer science databases, which had previously been used in recent systematic review studies (ARANDA et al., 2019; BAVARESCO et al., 2020).

Research in ACM and IEEE Xplore, Science Direct, and Wiley required the use of an advanced search feature. Google Scholar and Scopus search required a combination of the summary and title fields in the advanced search option. Finally, in Springer Link removing documents categorized as "preview only"and select the search filter titled 'computer science' to obtain results.

3.2.3 Study Filtering

After gathering literary works through the search string, the filtration process sorts the papers related to the research area and removes those that are not. The following rejection criteria (RC) allowed to remove papers: the study must have been published in a journal, conference, or workshop (IC 1); the study must be related to the proposed theme – data analysis, EC, IoT and context awareness in SG (IC 2); the study must be a complete paper (IC 3).

The following inclusion criteria (IC) allowed the filtering of papers: studies published prior to 2010 (RC 1); studies not written in English (RC 2); studies published as dissertations or theses (RC 3); studies which did not have any relation to the research questions (RC 4).

The inclusion and rejection criteria enabled the attainment of the most relevant studies and removed any noise generated in the research. Figure 5 shows the result of the filtering process. The initial filtering of papers consisted of removing impurities that did not comply with the RC 1, 2, and 3. Lastly, the RC 4 enabled the extraction of any residuals through the three-pass method (KESHAV; S., 2016). The first step of the three-pass method comprised four stages: 1) read the title, the abstract, and the introduction of each paper; 2) read the titles of each section and subsection; 3) look at the mathematical equations (if available) to review whether they are consistent with the theoretical grounds presented in the paper; 4) read the conclusions. The second step involved carefully reviewing figures, diagrams, and other illustrations of papers, with specific attention given to figures. Finally, the third step was to read the full text, observing the RC 4.

3.3 Systematic Mapping Results

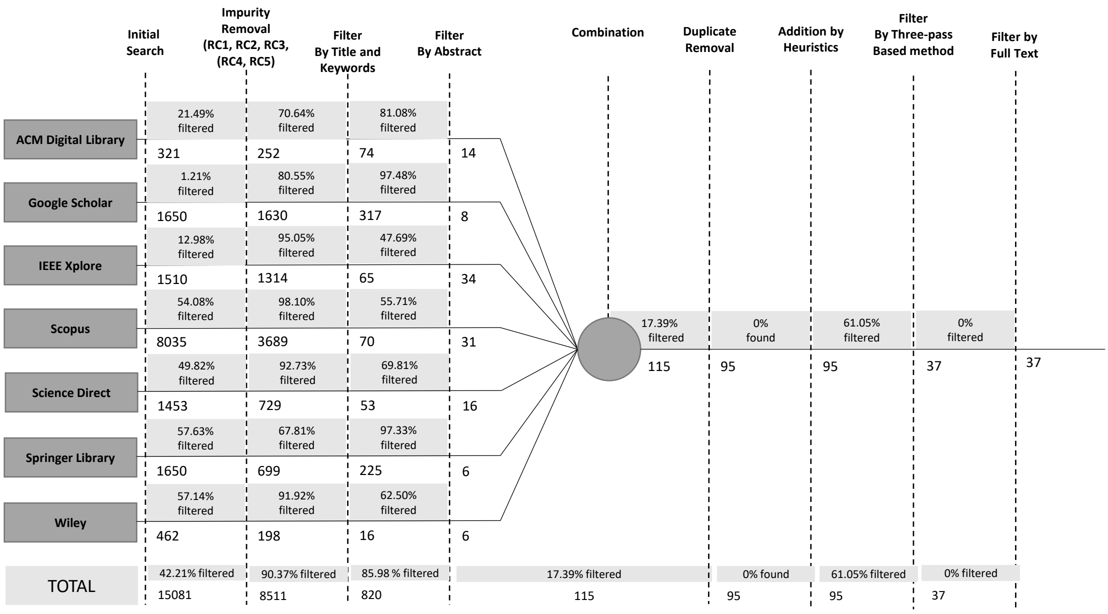
The filtering process resulted in 37 papers. At this stage, the selected papers were analyzed according to their objectives. Table 4 presents papers indicating the paper Id, year of publication, data analysis technique, the use of context awareness, EC, IoT, H-Index, and paper DOI.

3.3.1 *GQ1 - How have data analysis and Internet of Things been used to support Edge Computing in Smart Grids?*

SG systems integrate technologies. These technologies can be used for different purposes, such as forecasting electric power demand and predicting problems in an electric grid.

Zhang et al. (2019) proposed real-time monitoring of residential load schedule. The work

Figure 5: Process of studies filtering



Source: Prepared by the author

uses a deep learning framework through IoT devices to predict load variations throughout the day. Carvalho, Roloff, Navaux (2017) presented an architecture that uses IoT devices to process collected SG data distributively. This architecture considers the network latency and sends data when the network is not busy.

The work of He et al. (2019) developed an ML algorithm based on causal feature selection in order to predict power events in the grid – such as outages. Liu et al. (2014) introduced a big data index to save SG collected data in different types of indexes, reducing the space required to save data. Omara et al. (2018) proposed a framework to transfer data according to the current context. In case of a sensitive event, the framework sets a high priority in the node where the event occurred.

Bin et al. (2019) developed a micro-service that considers changes in the network and uses this information to control company business applications. The authors' simulations show that the micro-service may reduce electrical maintenance costs.

The work proposed by Mhdawi, Al-Raweshidy (2020) applies a neural network (NN) framework with two approaches: firstly, the data is predicted at the edges; secondly, in case of a failure, the EC node computing tasks can be bypassed to another node. Newaz et al. (2014) considered an SG located on a university campus. The SG data is collected, sent to a server, and used to predict future load variations. Huang et al. (2018) presented a framework that monitors and reduces latency in SG networks. Using a data compression algorithm, the authors obtained 85% of reduction in latency.

The work of Raju et al. (2020) consists of an application that can predict the load of an SG in short periods. In order to predict these variations, the study uses different algorithms – such as Radial Basis Function (RBF), Decision Tree Regression (DTR), and random forest (RF). Khaouat, Benhlima (2016) showed an architecture that improves SG data management. Using big data techniques, the architecture improves the information of data analytics systems with a cost-efficiency dashboard.

The research of Tasfi et al. (2017) proposes a deep semi-supervised Convolutional Neural Network (CNN) with confidence sampling for electrical anomaly detection. The solution uses two sub-networks in order to achieve semi-supervised learning. While the first performs reconstruction and uses unlabelled data, the second performs classification with labeled data. Soykan et al. (2019) presented a practical implementation of load forecasting with differential privacy techniques using the Tensorflow Privacy library. The authors show that data privacy guarantee can be achieved to varying degrees with a tolerable degradation in the forecasted values.

Kulkarni et al. (2019) developed a functional unit of the EC node, taking into account constraints – such as costs, customizations, data storage, cybersecurity, and power management. The platform was built, deployed, and applied in distributed SG applications like power quality measurements, automated metering infrastructure, and utility asset monitoring. Mousavi, Stoupis, Saarinen (2018) proposed a decision tree-based methodology for identifying the origin of a general abnormality in SGs through data from a multi-feeder distribution system. The work of Gore, Sawai, Kour (2019) presents an IoT-based SG analyzer that efficiently utilizes the advantages put forward by IoT technology to improve situational awareness of power grids. The mobile-based software tool enables power system experts, including operators, to make decisions based on the current SG condition.

The study of Chen et al. (2019) introduces an EC system for IoT-based SGs to overcome the drawbacks in the current CC paradigm. Additionally, the work implements a privacy protection strategy via EC and data prediction. Vantuch et al. (2018) developed a boosting model to predict short-term and long-term loads. After comparing computational models on three different regression-based criteria, the results revealed that the model outperformed its competitors in most of these comparisons.

Dalcekovic et al. (2017) proposed a general approach to service design considerations based on big data platforms. The method, implemented using Apache HBase, is applied in the context of demand response along with Distributed Management System (DMS) applications for managing SGs. Hasnat et al. (2019) designed a framework that operates providing Distributed event detection (DSE) locally – at the edge nodes. Focusing on Phase Measurement Units (PMUs) as an example of the industrial IoT in SGs, the framework uses an EC platform architecture to enable data analytics for DSE using the PMUs time-series measurements.

Wang et al. (2020) presented a data analysis and application framework for intelligent meter data based on cloud-fog computing and data contextualization. A data contextualization model based on three-dimensional (3D) maturity analysis of industrial power users is proposed to evaluate load characteristics of users from consumption behavior. Xia et al. (2018) developed an algorithm of distributed processing to detect malicious use of energy in SG areas. The authors claim that the work results show that this model allows power grid operators to understand quickly and intuitively the load behavior pattern and power demand of industrial power users.

In order to predict events in SG, Liu et al. (2020) developed a CNN using Long Short-Term Memory (LSTM). The work claims that the method increases the training speed by 61.7%, reduces Root Mean Square Error (RMSE) by 32.9%, and enhances the prediction accuracy by 1.4% compared to similar solutions. Using an MLSTM model, Alazab et al. (2020) considered a NN to predict the stability of an SG. According to the authors, the experimental results prove that the MLSTM approach outperforms the other ML approaches.

The work of Jurado et al. (2017) uses fuzzy reasoning to estimate lost data during data collection improving the accuracy by around 31.5% for different data sets tested by the authors. Mukherjee et al. (2020) implemented standard regression and ML-based architectures for SG load analysis and forecasting. The proposed approach predicts 97% of registers when 73% of training data have missing values.

Junior et al. (2019) presented a low-cost smart meter methodology. According to the authors, the solution has a good potential since it has a low-cost implementation. The work of Rabie et al. (2018) proposes a fog data forecasting approach using EC nodes in an SG. These nodes use big data to increase the system's prediction accuracy. Additionally, Rabie et al. (2019) designed a methodology using outlier rejection to improve the accuracy of big data in SG. The work of Qureshi et al. (2018) also uses big data techniques to improve the performance of the SG network by reducing latency and package loss.

Ahmad et al. (2020) developed an ML model to detect problems in an SG according to the current weather. After comparing with four other ML models, the authors concluded that the proposed solution has better results. Mihailescu, Ossowski, Klusch (2016) provided a computational characterization system in terms of complexity, as well as an empirical analysis against real consumption data sets. Based on the macro-model of the Australian energy market, the results show a performance improvement of about 17%. Chen, Li, Huang (2018) created an anomaly detection monitoring consumption through IoT. The authors used profile similarity analysis to detect a possible fault in the SG network.

Zainab et al. (2021b) developed a technique that improves the speed and the accuracy of different models for short-term prediction of SGs. According to the authors, the random tree algorithm obtained the best results using an SG dataset. The work of Qadir et al. (2021) develops an Artificial Neural Network (ANN) to forecast possible energy generation by solar panels. According to the authors, the prediction applying linear regression has 95% accuracy. The model of Krč et al. (2021) applies CNN to classify power demand for 42 different cities obtaining an average accuracy of 96%. Finally, Aldegheishem et al. (2021) proposed a model that combines support vector machine with CNN to detect possible outliers and electricity theft in SGs.

Table 4: Reviewed Studies

Id	Year	Data Analysis	Context Awareness	EC	IoT	H-Index	Doi
1	2019	Deep Learning	No	Yes	Yes	8	10.1145/3302505.3310069
2	2017	No	Yes	Yes	Yes	21	10.1145/3147234.3148105
3	2019	Random Forest	No	No	No	12	10.1145/3341162.3349333
4	2014	Random Forest	No	No	No	116	10.14778/2733004.2733021
5	2018	Deep Learning	No	No	No	7	10.1145/3265863.3265883
6	2019	No	Yes	Yes	Yes	29	10.1145/3358528.3358576
7	2020	ANN	No	No	No	66	10.1109/JSYST.2019.2921867
8	2014	Similarity Profile Analysis	No	Yes	Yes	16	10.1109/ICTC.2014.6983110
9	2018	Business Analytics	No	Yes	Yes	-	10.1109/icii.2018.00019
10	2020	ANN	No	Yes	Yes	14	10.1109/icosec49089.2020.9215329
11	2016	Feature Selection	No	Yes	Yes	-	10.1109/irsec.2016.7983902
12	2017	CNN	No	No	No	17	10.1109/ithings-greencom-cpscom-smartdata.2017.158
13	2019	Deep Learning	No	No	No	-	10.1109/gcwkshps45667.2019.9024520
14	2019	No	Partially	Yes	Yes	67	10.1109/jiot.2019.2898837
15	2018	Random Forest	No	No	No	-	10.1109/tdc.2018.8440570.
16	2019	Business Analytics	No	Yes	Yes	-	10.1109/i-pact44901.2019.8960098
17	2019	ANN	No	Yes	Yes	86	10.1109/access.2019.2920488
18	2018	Random Forest	No	No	No	14	10.1109/wf-iot.2018.8355123
19	2019	Business Analytics	No	Yes	Yes	-	10.1109/isncc.2017.8072030
20	2019	No	Yes	Yes	Yes	-	10.1109/gcwkshps45667.2019.9024632
21	2020	Similarity Profile Analysis	Yes	Yes	Yes	86	10.1109/ACCESS.2020.2965543
22	2018	Similarity Profile Analysis	No	No	No	86	10.1016/j.cose.2018.05.004
23	2020	CNN	No	Yes	Yes	43	10.1016/j.scs.2020.102363
24	2020	Deep Learning	Yes	Yes	No	86	10.1016/j.asoc.2016.11.040
25	2017	Fuzzy Neural Network	No	No	No	124	10.1016/j.asoc.2016.11.040
26	2020	Deep Learning	No	Yes	Yes	20	10.1016/j.suscom.2019.100356
27	2019	No	Partially	No	Yes	81	10.1016/j.measurement.2019.106890
28	2019	Random Forest	No	Yes	Yes	41	10.1007/s10586-018-2848-x
29	2020	Deep Learning	No	Yes	Yes	41	10.1007/s10586-019-02942-0
30	2019	Business Analytics	No	Yes	Yes	54	10.1007/s11277-018-5936-6
31	2020	Random Forest	Yes	Yes	No	173	10.1016/j.energy.2020.117283
32	2016	Business Analytics	No	No	No	49	10.1111/coin.12093
33	2018	Similarity ProfileAnalysis	No	Yes	Yes	63	10.1002/cpe.4737
34	2021	RandomForest	Yes	No	No	127	10.1109/access.2021.3059730
35	2021	ANN	Yes	No	No	33	10.1016/j.egyr.2021.01.018
36	2021	CNN	No	No	No	85	10.3390/su13052954
37	2021	CNN	No	No	No	127	10.1109/access.2021.3056566

Source: Prepared by the author.

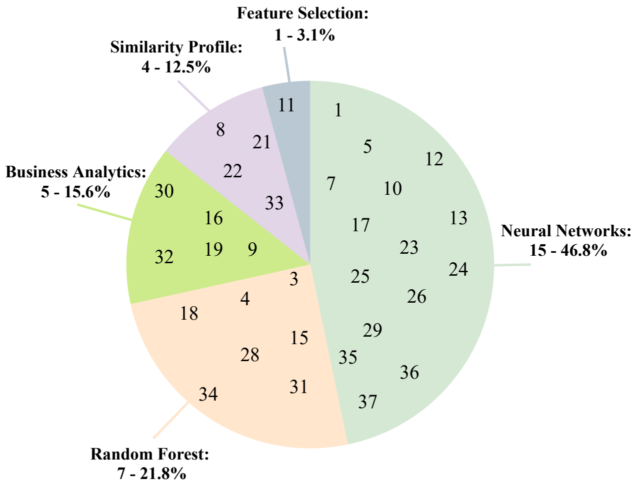
3.3.2 *FQ1 - Which are the data analysis techniques applied to Edge Computing in Smart Grids?*

Figure 6 shows the most used data analysis techniques. Seven works use ML-based approaches with RF (HE et al., 2019; LIU et al., 2014; MOUSAVI; STOUPIS; SAARINEN, 2018; VANTUCH et al., 2018; RABIE et al., 2018; AHMAD et al., 2020; ZAINAB et al., 2021b). These studies use a variation of the RF algorithm to predict long-term loads, events like faults, and even weather conditions that affect the SG.

Fifteen works consider NN for ML-based solutions. Some of these studies focus on event, stability predictions, and short-term load prediction in SG (Zhang et al., 2019; OMARA et al., 2018; MHDAWI; AL-RAWESHIDY, 2020; RAJU et al., 2020; SOYKAN et al., 2019; CHEN et al., 2019; LIU et al., 2020; MUKHERJEE et al., 2020; RABIE et al., 2019; QADIR et al., 2021). Others studies detect anomalies and predict power demand in SG (TASFI et al., 2017; XIA et al., 2018; KRČ et al., 2021; ALDEGHEISHEM et al., 2021). Finally, the last NN study applies fuzzy reasoning to attenuate data loss during collection, and transmission (JURADO et al., 2017).

Five papers use business analytics to send more information to the SG systems' management (HUANG et al., 2018; GORE; SAWAI; KOUR, 2019; DALCEKOVIC et al., 2017; QURESHI et al., 2018; MIHAILESCU; OSSOWSKI; KLUSCH, 2016). Four similarity profile studies identify consumption patterns of residential neighbors or university campuses in order to predict future power demand (NEWAZ et al., 2014; ALAZAB et al., 2020; WANG et al., 2020; CHEN; LI; HUANG, 2018). Finally, one study uses feature selection to provide relevant information to SG management systems (KHAOUAT; BENHLIMA, 2016).

Figure 6: Types of data analysis techniques.



Source: Prepared by the author.

3.3.3 *FQ2 - Are there any studies which consider contexts, context histories and context prediction, according to the Dey's definition (DEY; ABOWD; SALBER, 2001)?*

Six works consider context awareness in data analysis (CARVALHO; ROLOFF; NAVAUX, 2017; BIN et al., 2019; HASNAT et al., 2019; ALAZAB et al., 2020; LIU et al., 2020; AHMAD et al., 2020). Five of them (CARVALHO; ROLOFF; NAVAUX, 2017; BIN et al., 2019; HASNAT et al., 2019; ALAZAB et al., 2020; LIU et al., 2020) use the SG context in real-time, according to voltage measurements. Four studies uses context prediction (DALCEKOVIC et al., 2017; AHMAD et al., 2020; ZAINAB et al., 2021b; QADIR et al., 2021) through weather forecasting in order to predict possible instability in the grid.

The remaining SG works apply CC in the network's architecture. This architecture implies that the intelligence of SG occurs far from the substations. The application of EC in addition to context-aware techniques, may help the SG systems to explore this gap.

3.3.4 *FQ3 - Which are the adaptation strategies used for improving data management in Edge Computing applied to Smart Grids?*

Five works use adaptation strategies (CARVALHO; ROLOFF; NAVAUX, 2017; BIN et al., 2019; KULKARNI et al., 2019; HASNAT et al., 2019; JUNIOR et al., 2019). Three of these works (CARVALHO; ROLOFF; NAVAUX, 2017; HASNAT et al., 2019; JUNIOR et al., 2019) consist of changing the edge node to a different working node in case of a node failure. Two of the works (BIN et al., 2019; KULKARNI et al., 2019) adapt when the edge nodes send data to the cloud. If the edge network latency is too high in these cases, the node waits and tries to send again when the edge network has a better latency.

The use of context-aware techniques can achieve the systems' adaptivity, since systems need to know their possible next events. Context awareness may help in this case.

3.3.5 *FQ4 - How has the Internet of Things been used for Edge Computing in Smart Grids?*

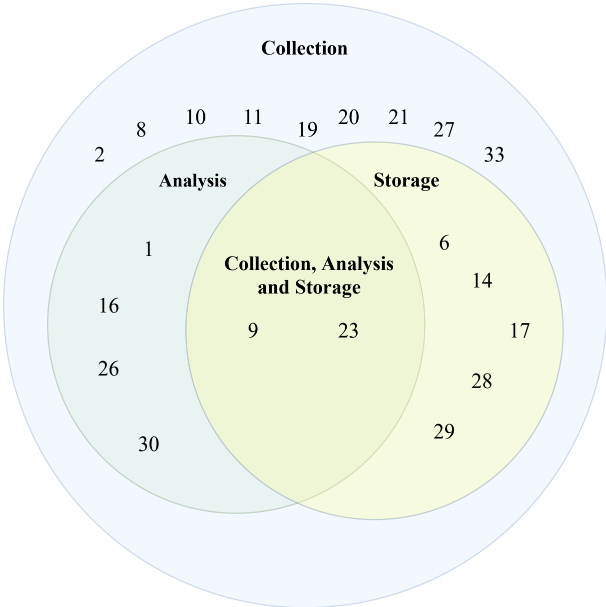
IoT has been used in four different ways by 20 of the reviewed works. The first type considers data collection, storage and analysis in the IoT layer (HUANG et al., 2018; XIA et al., 2018). The second type regards data collection, and data analysis by the IoT layer (Zhang et al., 2019; GORE; SAWAI; KOUR, 2019; MUKHERJEE et al., 2020; QURESHI et al., 2018). The third type contemplates works which only collect data and store (temporarily or not) in the IoT layer (BIN et al., 2019; KULKARNI et al., 2019; CHEN et al., 2019; RABIE et al., 2018, 2019). Finally, the last type considers works which only collect data through the IoT layer (CARVALHO; ROLOFF; NAVAUX, 2017; NEWAZ et al., 2014; RAJU et al., 2020; KHAOUAT; BENHLIMA, 2016; DALCEKOVIC et al., 2017; HASNAT et al., 2019; ALAZAB et al., 2020; JUNIOR et al., 2019; CHEN; LI; HUANG, 2018). Figure 7 shows the different uses of IoT in SGs.

3.3.6 *FQ5 - How have the works used big data for supporting Edge Computing in Smart Grids?*

Nine works use big data techniques in order to clean raw SG data using tools such as Apache Hadoop, MongoDB, Hive, Map Reduce, and Tableau (NEWAZ et al., 2014; HUANG et al., 2018; KHAOUAT; BENHLIMA, 2016; DALCEKOVIC et al., 2017; ALAZAB et al., 2020; WANG et al., 2020; QURESHI et al., 2018; MIHAILESCU; OSSOWSKI; KLUSCH, 2016; CHEN; LI; HUANG, 2018). Additionally, the data is stored in relational or non-relational databases, where the developed solutions use the information in management dashboards, predict energy demand and determine possible anomalies in the SG.

Three works use MongoDB as non-relational databases (NEWAZ et al., 2014; DALCEKOVIC et al., 2017; CHEN; LI; HUANG, 2018). Other three use map-reduce techniques and

Figure 7: Different uses of Internet of Things in Smart Grid studies.



Source: Prepared by the author.

store data in relational databases (GORE; SAWAI; KOUR, 2019; ALAZAB et al., 2020; QURESHI et al., 2018). Two works use Apache Hadoop (HUANG et al., 2018; MIHAILESCU; OSSOWSKI; KLUSCH, 2016) to store SG data in clusters spread across different servers to reduce the processing required to analyze SG data. Finally, one work (WANG et al., 2020) uses business intelligence and Tableau to display SG data from a relational database. Table 5 summarizes the big data tools and types of databases used in the reviewed papers.

Table 5: Big data tools and types of databases

Authors and Paper Id	Big Data Tool/ Big Data Technique	Type of Database
Qureshi et al. (Id 30)	Map Reduce	Relational
Gore et al. (Id 16)	Map Reduce	Relational
Wang et al. (Id 21)	Map Reduce	Relational
Mihailescu et al. (Id 32)	Apache Hadoop	Hybrid
Huang et al. (Id 9)	Apache Hadoop	Hybrid
Newaz et al. (Id 8)	Mongo DB	non-Relational
Dalcekovic (Id 19)	Mongo DB	non-Relational
Chen et al. (Id 17)	Mongo DB	non-Relational
Xia et al. (Id 22)	Tableau	Relational

Source: Prepared by the author.

3.3.7 *FQ6 - How has Machine Learning been used for supporting Edge Computing in Smart Grids?*

Four works use ML, and EC techniques simultaneously (RAJU et al., 2020; MUKHERJEE et al., 2020; RABIE et al., 2018, 2019). Raju et al. (2020) used edge nodes to clean the data. Differently, Mukherjee et al. (2020) applied edge nodes to analyze part of the collected data in order to help the server-side to predict futures loads. Two works of the authors Rabie et al. (2018; 2019) temporarily store collected data in the edge nodes and send them when the SG network has low latency. The work of Krč et al. (2021) uses ML nodes in different cities working as edge nodes of an SG. The nodes are implemented in each city's electrical substations and perform energy demand prediction of the SG.

3.3.8 *SQ1 - What is the number of publications per type?*

Figure 8 shows the selected papers' publication data by type of venue, quantity, year, and digital library. This selection shows that 20 journal publications correspond to 54.05%, 15 conference papers account for 40.54%, and two workshop publications correspond to 5.40% of the studies reviewed in this chapter.

3.3.9 *SQ2 - How many publications occurred per year?*

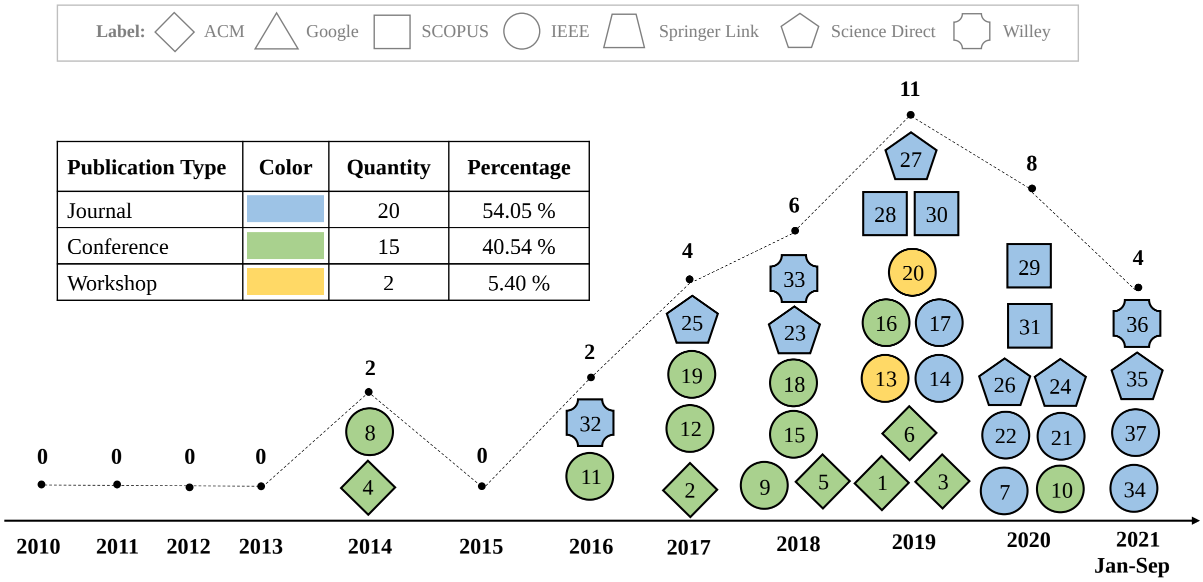
Figure 8 shows the distribution of papers by the year of publication. These papers were analyzed from 2010 up to April 2021 since the study was performed by the end of May 2021. Research in the area has been ongoing since 2014, with an increase in 2017 and a slight variation of papers published after 2019.

3.4 Discussion

The studies reviewed in this chapter include different types of technologies from the SG domain, data analysis, context awareness, IoT and EC. Thirteen studies apply EC, IoT, and data analysis (Zhang et al., 2019; NEWAZ et al., 2014; HUANG et al., 2018; RAJU et al., 2020; KHAOUAT; BENHLIMA, 2016; GORE; SAWAI; KOUR, 2019; CHEN et al., 2019; DALCEKOVIC et al., 2017; MUKHERJEE et al., 2020). These studies consist of a node network that sends data to a server or analyses the data in the EC layer. Five papers employ the IoT, EC, and context awareness techniques (BIN et al., 2019; KULKARNI et al., 2019; HASNAT et al., 2019; JUNIOR et al., 2019; QADIR et al., 2021). These papers consider network adaptation by sending data according to the latency of the SG system or SG and weather contexts to predict events that can affect the SG.

Two papers implement only context-aware EC and data analysis (LIU et al., 2020; AH-

Figure 8: Publications per year by type and digital library.



Source: Prepared by the author.

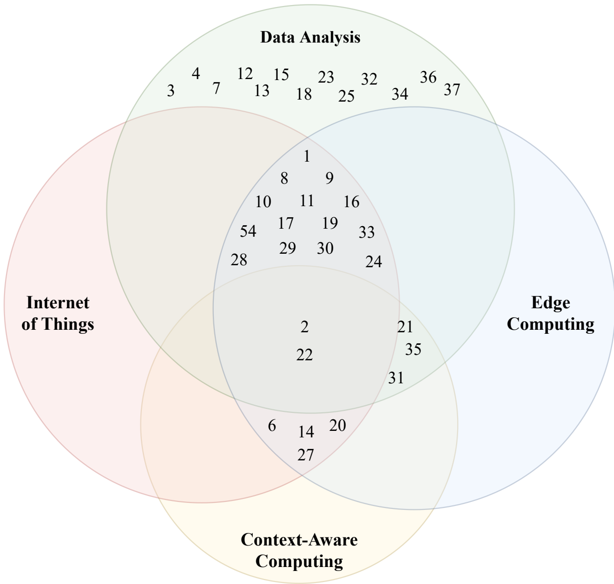
MAD et al., 2020). The aforementioned studies have a processing device at the edge to detect SG problems and distribute the processing in a different SG node. Pure data analysis technologies are used in eleven papers, applying ML techniques with measured and simulated data sets (HE et al., 2019; LIU et al., 2014; OMARA et al., 2018; MHDAWI; AL-RAWESHIDY, 2020; TASFI et al., 2017; SOYKAN et al., 2019; MOUSAVI; STOUPIS; SAARINEN, 2018; VANTUCH et al., 2018; WANG et al., 2020; JURADO et al., 2017; MIHAILESCU; OSSOWSKI; KLUSCH, 2016; KRČ et al., 2021; ALDEGHEISHEM et al., 2021).

Finally, two works applied the four technologies extensions within the SG domain (data analysis, EC, IoT, and context awareness) (CARVALHO; ROLOFF; NAVAUX, 2017; ALAZAB et al., 2020). Figure 9 shows the intersections between these technologies, indicating that the majority of the SG reviewed works use data analysis, EC, and IoT at some level – although context-aware solutions are found in nine of the mapped works (CARVALHO; ROLOFF; NAVAUX, 2017; BIN et al., 2019; VANTUCH et al., 2018; HASNAT et al., 2019; ALAZAB et al., 2020; LIU et al., 2020; AHMAD et al., 2020; ZAINAB et al., 2021b; QADIR et al., 2021). The lack of context-aware researches may denote an opportunity for future works in the area.

IoT and data analysis applied to SGs have more works than EC and context-aware computing. Context-aware solutions have fewer studies, providing potential opportunities, specifically through works that apply weather forecasting to detect grid failures.

Only two works reducing latency between the SG edges and the operation center apply EC techniques (HE et al., 2019; KULKARNI et al., 2019). Two other works applied to SG use Fuzzy NN (JURADO et al., 2017), and feature selection (KHAOUAT; BENHLIMA, 2016), both ML techniques. Further studies of these techniques may point to new opportunities for future works.

Figure 9: Intersection between Smart Grid technologies.



Source: Prepared by the author.

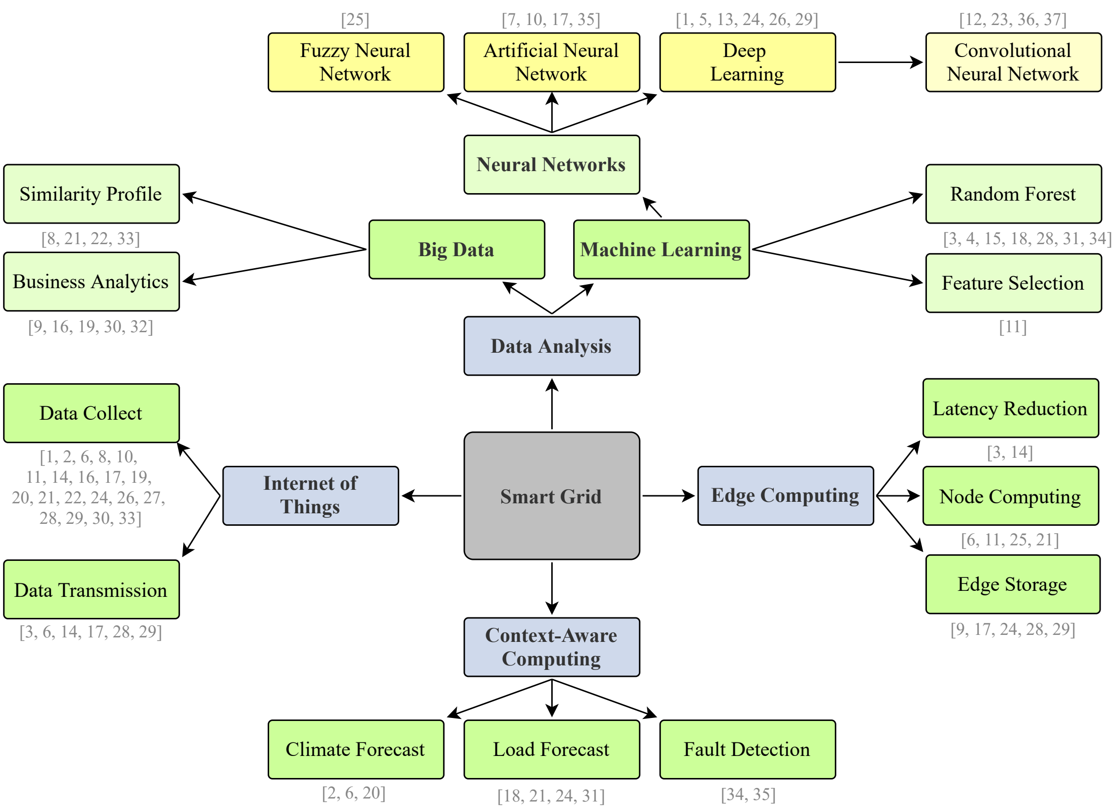
Considering the works analyzed in this chapter, an SG taxonomy can provide an answer to the opening question in the form of a "categorized list of SG concepts". The list contains the reviewed papers' concepts related to SG, splitting them into a set of meaningful categories. The SG elements are distinguished according to different types: data analysis, EC, IoT, and context awareness.

Figure 10 displays the SG taxonomy according to the concepts' relations – with the numbers representing the reference of the selected papers. The figure shows that data analysis techniques are organized into ML (RF, feature selection, NN) and big data (business analytics, similarity profile). The NN class is divided into three other subclasses: deep learning, ANN, and fuzzy NN. The Deep learning class has a subclass called CNN. Most data analysis works are categorized as ML, indicating that artificial intelligence in SG is a well-established concept. The primary use of IoT is data collection followed by data transmission, probably due to the small sensor devices close to the data origin.

Additionally, EC is used for edge storage, node computing, and latency reduction, analyzing the data collected at the origin. This collection and analysis implicate a possible decrease in latency since less data needs to reach a data center for further analysis. Finally, few works consider context-aware computing (climate forecast, load forecast, fault detection) in the SG, which may indicate possible trends in future works.

The systematic mapping of the papers provided insights, detailed solutions, and topics that need future exploration in the SG domain. In addition, lessons learned indicate concepts esta-

Figure 10: Taxonomy of Smart Grid technologies.



Source: Prepared by the author.

blished within the domain of SG. Table 6 presents seven lessons learned, summarizing contributions and observations of this study. Lesson 1 shows that ML and NN are established concepts in the SG domain. Lesson 2 presents an opportunity to decentralize the analyzed data in an SG. Lessons 3 and 4 mention the usage of context histories and context-aware computing in SG. Lesson 5 tells the primary application of EC in SG. Lesson 6 presents that IoT is a concept established in SG. Lesson 7 shows a risk of data loss in SG.

3.5 Systematic Mapping Updated Search

A new search was conducted using the search string presented in Section 3.2 to update the related works. This search covered the period from September 2021 to June 2024. Five articles were found and directly inserted into the related works for comparison with the Freya model. These articles provide new insights and advancements in Machine Learning applications for SG event prediction.

The article of Martinelli, Mercaldo, and Santone (2022), proposed a method for automatically reading digits from dial meters using deep learning, specifically the YOLOv5s model. This method aims to enhance the implementation of Smart Grids by automating meter readings,

Table 6: Lessons learned

Lesson	Description
1	ML is a concept already established in SG applied to 62% of the works. NN correspond to 40% of the filtered works, being the most used ML technique.
2	ML combined with EC helps reducing the amount of analyzed data in the server.
3	Context histories works mainly consider weather forecasts to predict grid instability caused by bad weather.
4	Context-aware computing is only considered in 9 works (24.3%), which may denote an opportunity to explore in the SG domain.
5	EC techniques can reduce the latency between the grid edge and the grid operation center.
6	IoT is a concept already established in SG applied to 56% of the filtered works. The majority of works (54%) use sensors to perform data collection. One work (2%) applies IoT to perform only data transmission.
7	Data loss is a threat considered by five studies (13.5%). Five works propose edge storage in SG to attenuate this threat.

Source: Prepared by the author.

suggesting that it is efficient in SG management.

The work of Deepak Gangadharan et al. (2021), discussed the design and implementation of an IoT system calibrated with Machine Learning for measuring Total Dissolved Solids (TDS). The paper demonstrates its application in a smart campus environment, which can be extended to SGs.

The study of Kavya et al. (2023), focused on refining issues faced by existing power grids, such as unidentified fault detection, prediction of power generation, and utilization at the consumer side. The proposed Machine Learning-based Smart Grid uses cloud-edge computing to control renewable energy power generation, power prediction, fault detection, and utilization. Simulation analysis shows improved performance compared to CNN-based cloud computing.

The paper of Aflhana et al. (2023)focused on adaptive Machine Learning techniques to detect faults in Smart Grids. The authors propose a novel adaptive learning algorithm that continuously updates the model based on real-time data. Experiments conducted on a simulated SG network indicate that the adaptive model outperforms traditional static models in detecting faults and minimizing false positives.

The article of Wang et al. (2023), proposed a unified Machine Learning framework for simultaneously performing electrical load forecasting and unsupervised anomaly detection in real-time. The framework uses a training data generator (TDG) and a look-back optimizer (LBO) to enhance the performance of ML-driven prediction approaches. It operates on raw meter data without requiring input conditioning or additional information. The proposed framework is evaluated using a real-world power consumption dataset, showing superior outcomes compared to alternative methods.

These newly identified articles contribute to the existing body of knowledge by introducing innovative methods and applications of Machine Learning for Smart Grid event prediction. These articles highlight advancements in automated meter reading, IoT system design, adaptive learning algorithms, and real-time anomaly detection frameworks. Finally, this article increased the related works from 21 to 26 papers, where 21 were in the original search, and the last five were added after this update. These 26 related works are the final selection of articles to discuss related works and the Freya model.

3.6 Related Works

The final selection of 26 papers, comprising 21 from the original systematic mapping search and five from the updated search conducted between September 2021 and June 2024, provides a comprehensive foundation for discussing the related works. These papers encompass the features and insights required to compare and evaluate the Freya model within the context of Smart Grid technologies. This curated selection ensures that the discussion is informed by the most relevant and recent research in the field, facilitating a thorough analysis of existing approaches and identifying gaps addressed by the Freya model.

Seven works use ML-based approaches with RF (HE et al., 2019; MOUSAVI; STOUPIS; SAARINEN, 2018; VANTUCH et al., 2018; RABIE et al., 2018; AHMAD et al., 2020; ZAINAB et al., 2021b; GOPARAJU et al., 2021). These studies use a variation of the RF algorithm to predict long-term loads, events like faults, and even weather conditions that affect the SG. Nineteen works consider NN for ML-based solutions. Some of these studies focus on event, stability predictions, and short-term load prediction in SG (Zhang et al., 2019; OMARA et al., 2018; MHDAWI; AL-RAWESHIDY, 2020; RAJU et al., 2020; SOYKAN et al., 2019; CHEN et al., 2019; LIU et al., 2020; MUKHERJEE et al., 2020; RABIE et al., 2019; QADIR et al., 2021; MARTINELLI; MERCALDO; SANTONE, 2022; KAVYA et al., 2023; ALHANAF; BALIK; FARSADI, 2023; WANG; YAO; PAPAEFTHYMIOU, 2023). Other studies detect anomalies and predict power demand in SG (TASFI et al., 2017; XIA et al., 2018; KRČ et al., 2021; AL-DEGHEISHEM et al., 2021). Finally, the last NN study applies fuzzy reasoning to attenuate data loss during collection and transmission (JURADO et al., 2017).

The primary gaps left by the mapped works are summarized below. The extent to which each work addresses these gaps is in the comparison. Works that satisfy the criterion may completely, partially, or not fill the gap. The extent to which each of the 26 works fills the gaps indicates that the assessments consider the criteria of each gap. Table 7 presents the gaps found in the literature.

Gap 1 considers whether the works perform predictions of each entity in the network. None of the works fully fulfill this gap. The works that partially fulfill this gap (HE et al., 2019; MOUSAVI; STOUPIS; SAARINEN, 2018; VANTUCH et al., 2018; RABIE et al., 2018, 2019; ZAINAB et al., 2021b; Zhang et al., 2019; MHDAWI; AL-RAWESHIDY, 2020; RAJU et al.,

Table 7: Gaps of the related works

Gap	Description	Fulfill the Gap	Partially fulfill the Gap	Do not fulfill the Gap
1	The work performs predictions of entities events in the network	-	13	13
2	The study performs network prediction	-	3	23
3	The proposal considers dynamic network layout according to an event	-	1	25
4	The approach transfers entity event pattern to another entity	-	-	26
5	The ML algorithm retrains the model automatically	-	-	26

Source: Prepared by the author.

2020; CHEN et al., 2019; LIU et al., 2020; TASFI et al., 2017; WANG; YAO; PAPAEFTHYMIOU, 2023) consider entity predictions but do not consider a generic model to train and predict different entities of an SG network. The remaining works (AHMAD et al., 2020; OMARA et al., 2018; SOYKAN et al., 2019; MUKHERJEE et al., 2020; XIA et al., 2018; KRČ et al., 2021; ALDEGHEISHEM et al., 2021; JURADO et al., 2017; MARTINELLI; MERCALDO; SANTONE, 2022; GOPARAJU et al., 2021; KAVYA et al., 2023; ALHANAF; BALIK; FARSADI, 2023) do not predict entity events.

Gap 2 considers whether the implementation of each study performs network event prediction. None of the works fulfill this gap. Three works (HE et al., 2019; OMARA et al., 2018; MHDAWI; AL-RAWESHIDY, 2020) partially fulfill it due to performing network-related predictions, such as a global and centralized prediction, but not network events. The remaining works (MOUSAVI; STOUPIS; SAARINEN, 2018; VANTUCH et al., 2018; RABIE et al., 2018, 2019; ZAINAB et al., 2021b; Zhang et al., 2019; RAJU et al., 2020; CHEN et al., 2019; LIU et al., 2020; TASFI et al., 2017; AHMAD et al., 2020; SOYKAN et al., 2019; MUKHERJEE et al., 2020; XIA et al., 2018; QADIR et al., 2021; KRČ et al., 2021; ALDEGHEISHEM et al., 2021; JURADO et al., 2017; MARTINELLI; MERCALDO; SANTONE, 2022; GOPARAJU et al., 2021; KAVYA et al., 2023; ALHANAF; BALIK; FARSADI, 2023; WANG; YAO; PAPAEFTHYMIOU, 2023) do not perform network event prediction.

Gap 3 checks if the work considers dynamic network layout according to an event. All of the work still needs to fill this gap. One of the works (OMARA et al., 2018) partially fulfills the gap by changing the network layout based on a static approach. The remaining works (HE et al., 2019; MOUSAVI; STOUPIS; SAARINEN, 2018; VANTUCH et al., 2018; RABIE et al., 2018, 2019; ZAINAB et al., 2021b; Zhang et al., 2019; MHDAWI; AL-RAWESHIDY, 2020; RAJU et al., 2020; CHEN et al., 2019; LIU et al., 2020; TASFI et al., 2017; AHMAD et al., 2020; SOYKAN et al., 2019; MUKHERJEE et al., 2020; XIA et al., 2018; QADIR et al., 2021; KRČ et al., 2021; ALDEGHEISHEM et al., 2021; JURADO et al., 2017; MARTINELLI; MERCALDO; SANTONE, 2022; GOPARAJU et al., 2021; KAVYA et al., 2023; ALHANAF; BALIK; FARSADI, 2023; WANG; YAO; PAPAEFTHYMIOU, 2023) do not consider this feature.

Gap 4 verifies if an entity could send detected patterns as knowledge to another entity. This process of sending patterns is also known as transfer learning (GHORBANALI; SOHRABI, 2024). Sending patterns could help the entity's model detect events they are not trained to. None of the works consider this approach.

Finally, Gap 5 checks if the entity models in the network perform retraining to avoid data drift. Data drift is a concept that occurs when an input of ML models changes over time. This could reduce the model metrics (SAHINER et al., 2023) if not treated. None of the works consider this approach.

Two works support the identification of these gaps in the current literature. The work of Önder et al. (2023) highlights the necessity of entity-specific predictions and the lack of a generic model to train and predict different entities within an SG network. Similarly, the work by Dayananda et al. (2024) underscores the importance of network event prediction, dynamic network layout adaptation, and the potential benefits of transfer learning for event detection. These works collectively affirm the validity of the identified gaps and underscore the need for further research in these areas to enhance the reliability and adaptability of SG systems.

The ML proposed algorithm, a novel approach, fills gap 1 through its application at the edges of the distribution network. Gap 2 is addressed by the ML model using a unique model stacking technique. For gap 3, the ML algorithm considers distribution network layout variations that can dynamically change during operation. In the case of gap 4, the stacking method allows the transferring of knowledge or detected patterns between network entities. Finally, gap 5 meets the criterion by the ML algorithm due to the constant retraining of models carried out in the model stacking method.

Thus, this work's scientific contribution involves detecting events at the network's edges and across the entire network. It considers the hierarchical importance of each piece of equipment based on its connectivity and role in supplying power to consumers. The Freya model introduces a novel approach by employing a hierarchical structure that evaluates equipment's operational energy flow and context histories to improve event detection accuracy. The following section delves into the core concepts of the Machine Learning model, outlining the techniques and algorithms used to address the gaps identified in related works. This comprehensive approach aims to enhance the reliability and performance of Smart Grid systems by providing more accurate predictions and efficient management of network events.

3.7 Chapter Considerations

This chapter presents a comprehensive and novel update on the systematic mapping and analysis of literature in the domain of Smart Grid (SG) technologies. The focus is on data analysis, the Internet of Things (IoT), edge computing (EC), and context awareness. The systematic mapping, initially conducted from 2010 to 2021, has been refreshed to include five new articles published between 2021 and 2024. This update not only enhances the relevance and timeliness of the review but also underscores the dynamic nature of the field.

The new articles identified in the updated search have made contributions to the field, not just in terms of theoretical advancements but also in practical applications. These works have introduced methods for automated meter reading using deep learning, IoT system design for environmental monitoring, Machine Learning-based Smart Grids for home power management, adaptive Machine Learning techniques for fault detection, and real-time electrical load forecasting combined with unsupervised anomaly detection. These additions provide a broader perspective and highlight innovative approaches to overcoming existing challenges in SG management, thereby demonstrating the real-world impact of the research.

The chapter highlights the gaps in the existing literature, identifying areas where previous works fell short. While many studies partially addressed entity event predictions, network event predictions, dynamic network layouts, transfer learning, and automatic model retraining, they still need to fully meet these criteria.

The primary gaps identified and the extent to which each work addressed these gaps were summarized in Table 7. This table illustrated that while some works partially addressed specific gaps, the comprehensive approach of the Freya model uniquely fulfills all five gaps identified in the literature. The systematic mapping revealed that data analysis, IoT, and EC are well-established concepts within the SG domain, with a notable opportunity for further exploration in context-aware computing.

This chapter identified the current state of research, highlighting both established concepts and areas for future exploration. The findings underscored the importance of integrating multiple technologies to enhance the efficiency and reliability of SGs, ultimately contributing to this field. The chapter concluded with a summary of lessons learned, emphasizing the need for continued innovation and adaptation in SG technologies.

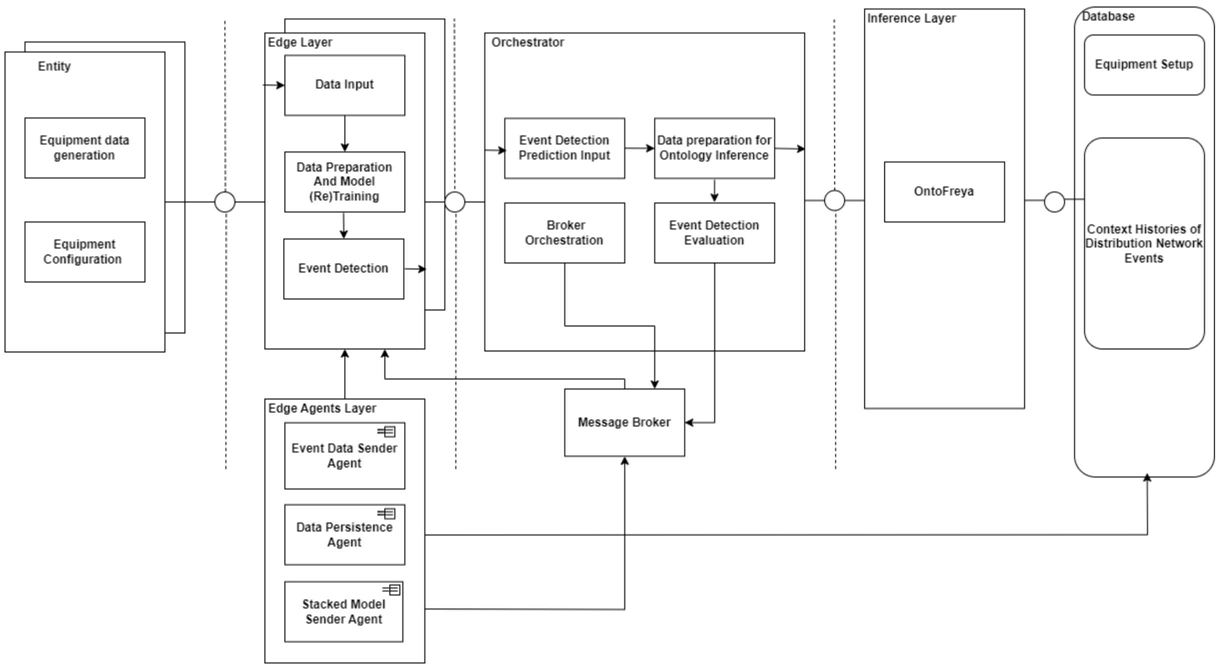
4 FREYA MODEL

This chapter presents the components of the Freya model. Section 4.1 displays the model architecture, and layers of the model. Section 4.2 introduces the Entity Layer. Section 4.3 introduces the Edge Layer. Section 4.4 approaches the Edge Agents Layers, and section 4.5 shows the Orchestrator. Section 4.6 details the Inference Layer. Finally, Section 4.7 presents the chapter considerations.

4.1 Freya Architecture

Figure 11 illustrates the architecture overview of the Freya model, depicting the interaction between layers and components involved in the event detection and prediction process for a power distribution network. The architecture is divided into layers, each with specific roles and responsibilities:

Figure 11: Architecture of Freya Model



Source: Prepared by the author.

Entity Layer:

- **Equipment Data Generation:** This component is responsible for retrieve the data from each equipment within the network.
- **Equipment Configuration:** This component manages the configuration settings of the equipment, ensuring they are set up correctly for data generation and communication.

Edge Layer:

- **Data Input:** It Receives new data inputs from the equipment.
- **Data Preparation and Model (Re)Training:** It Prepares the data for analysis and retrains the predictive models as new data is received.
- **Event Detection:** It Utilizes the trained models to predict events in the network based on the input data.

Edge Agents Layer:

- **Event Data Sender Agent:** Sends detected event data to the orchestrator layer.
- **Data Persistence Agent:** Ensures that the data is persisted for future reference and analysis.
- **Stacked Model Sender:** Sends the stacked models to other entities in the network or to the orchestrator for further processing.

Orchestrator:

- **Event Detection Prediction Input:** Receives the event prediction inputs from the edge layer.
- **Broker Orchestration:** Manages the orchestration of message brokers to ensure efficient communication between entities.
- **Data Preparation for Ontology Inference:** Prepares the data for ontology-based inference.
- **Event Detection Evaluation:** Evaluates the detected events and the metrics of edge models within the network.

Inference Layer:

- **OntoFreya:** An ontology-based component that performs inference on the prepared data to provide insights and predictions about the network's potential events.

Database:

- **Equipment Setup:** Stores the configuration and setup information of the equipment. Also stores the cache and database policies that could be synchronized with entities (YI et al., 2020)
- **Context Histories of Distribution Network Events:** Maintains historical data and context information about past events in the distribution network.

Message Broker:

- **Message Broker:** Facilitates communication between different layers and components by managing the message queues.

The process begins with data generation and configuration in the entity layer, followed by data input and preparation in the edge layer. The edge layer also handles event detection using trained models. Detected events and predictions are sent to the orchestrator through the edge agents layer. The orchestrator manages the event predictions, prepares data for ontology inference, and evaluates the detected events. The inference layer, powered by OntoFreya, provides insights and predictions based on the data.

The database stores all relevant data, including equipment setup and historical events. The message broker ensures smooth communication between all layers and components, supporting the distributed processing and scalability of the Freya model. This architecture ensures a comprehensive, scalable, and efficient approach to event detection and prediction in a power distribution network, leveraging advanced Machine Learning techniques and ontology-based inference.

The following sections delve into the detailed aspects of each layer and component presented in the Freya model. These layers encompass the intricate architecture that allows efficient event prediction within the Smart Grid. The discussion covers the specifics of the Entity Layer, Edge Layer, Edge Agents Layer, Orchestrator, and Inference Layer, providing a comprehensive understanding of how each part functions and contributes to the overall performance and reliability of the power distribution network.

4.2 Entity Layer

The entity layer represents the physical equipment within the Smart Grid infrastructure. This layer encompasses various types of equipment, each with its unique configurations and operational parameters. The primary function of the entity layer is to collect data directly from these pieces of equipment. The data collection includes gathering electrical metrics such as voltage, current, and power levels, which are relevant for monitoring the equipment's performance and detecting potential issues.

In addition to the core operational metrics, the entity layer also integrates sensors and other data sources that contribute to the context histories of the equipment. These sensors can monitor environmental conditions like temperature and humidity, which may affect the equipment's performance. The context histories provide a comprehensive view of the equipment's operational environment over time, helping to identify patterns and anomalies that could indicate impending failures or inefficiencies.

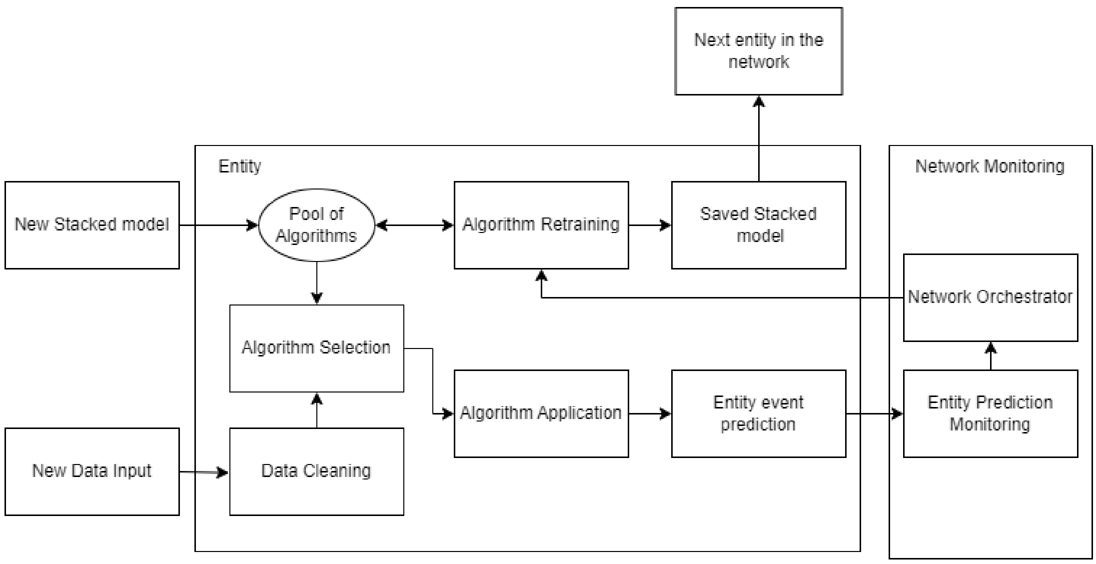
The configuration of each piece of equipment within the entity layer can vary depending on its type and function within the network. For example, a voltage regulator may have different monitoring needs and data collection mechanisms than a recloser. This variability necessitates a flexible data collection and configuration management approach within the entity layer.

The entity layer serves as the foundation for advanced analytics and predictive modeling by collecting and integrating operational metrics and contextual data. It enables the Freya model to leverage detailed, real-time information about the equipment's state and its surrounding environment, facilitating more accurate event detection and prediction. This comprehensive data collection and management approach ensures that the Smart Grid can operate efficiently and respond proactively to potential issues, ultimately enhancing the reliability and stability of the power distribution network.

4.3 Edge Layer

Figure 12 illustrates the workflow within the entity layer of the Freya model. This diagram demonstrates the process from data input to entity event prediction and interaction with the network monitoring system. The workflow can be summarized as follows:

Figure 12: Edge Layer



Source: Prepared by the author.

The process begins with inputting new data to maintain up-to-date predictions and improve the model's accuracy. This new data undergoes a cleaning process to ensure its quality and relevance for the subsequent analysis. From the pool of available algorithms, the most appropriate one is selected based on the current data and context.

An introduction of a new stacked model combines multiple algorithms from previous entities to enhance prediction accuracy. The selected algorithm and the new stacked model undergo retraining using the cleaned data to refine their predictive capabilities. The retrained algorithm is then applied to make predictions based on the current entity data.

The algorithm's application results in predictions of potential events for the entity. The updated and retrained stacked model is saved for future use and continuous improvement. The predictions made for the entity are monitored continuously to ensure timely and accurate event detection. The entity's predictions are input to the network orchestrator, which coordinates the overall network monitoring and management activities. The process repeats for the following entity in the network, ensuring a comprehensive and distributed approach to event prediction and network monitoring.

This workflow highlights the comprehensive and systematic approach taken by the Freya model to ensure accurate and timely event prediction in a power distribution network. Integrating data cleaning, algorithm selection, retraining, and continuous monitoring provides a robust framework for maintaining network reliability and performance.

4.4 Edge Agents Layer

The Freya model employs specialized agents to manage different aspects of data processing, event detection, and communication within the network. Each agent has specific roles and actions that contribute to the overall functionality and efficiency of the model. Figure 13 illustrates the interactions and actions of these agents. Freya agents modeling process used the Prometheus methodology (LARIOUI; BYED, 2021).

The following items and sub-items provide a more detailed explanation of each agent's role and actions within the Freya model. This in-depth exploration helps to clarify the specific responsibilities and interactions of each component, ensuring a comprehensive understanding of their contributions to the overall system.

Event Data Sender Agent:

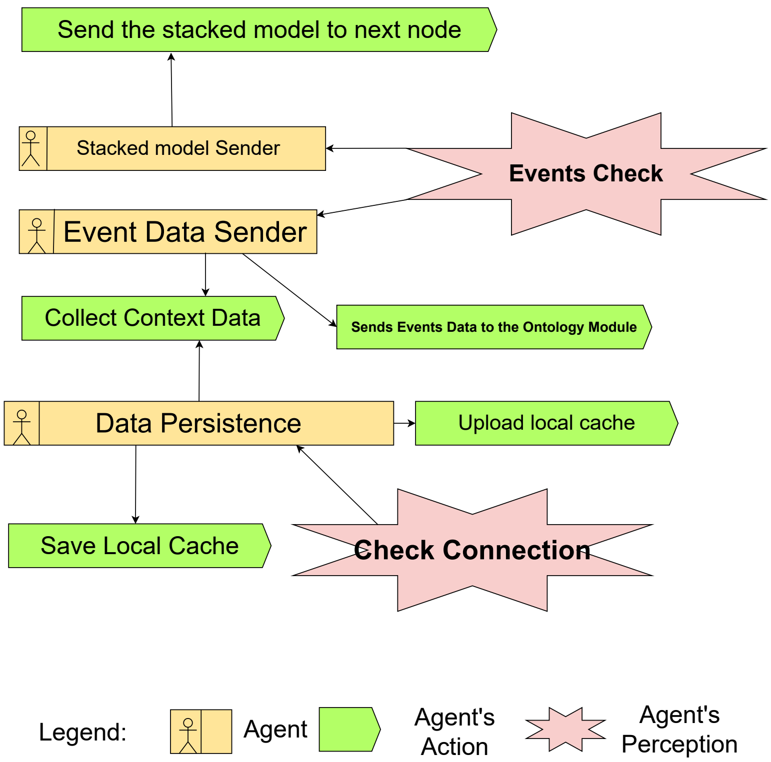
- **Role:** Responsible for sending detected event data to the ontology module for further analysis and inference.
- **Actions:**
  - Collect context data from the entity layer.
  - Send event data to the ontology module for processing.

Stacked Model Sender Agent:

- **Role:** Ensures that the updated and retrained stacked models are communicated to other nodes or entities within the network.
- **Actions:**
  - Send the stacked model to the next node in the network.

Data Persistence Agent:

Figure 13: Multiagents organization



Source: Prepared by the author.

- **Role:** Manages the storage and retrieval of data, ensuring that important information is preserved and can be accessed when needed.
- **Actions:**
  - Save local cache of data to ensure no information is lost.
  - Check connection status to determine if data can be uploaded.
  - Upload local cache to the central database once a stable connection is established.

Events Check Agent:

- **Role:** Monitors the network for events and ensures that all detected events are correctly processed and communicated.
- **Actions:**
  - Continuously check for new events within the network.

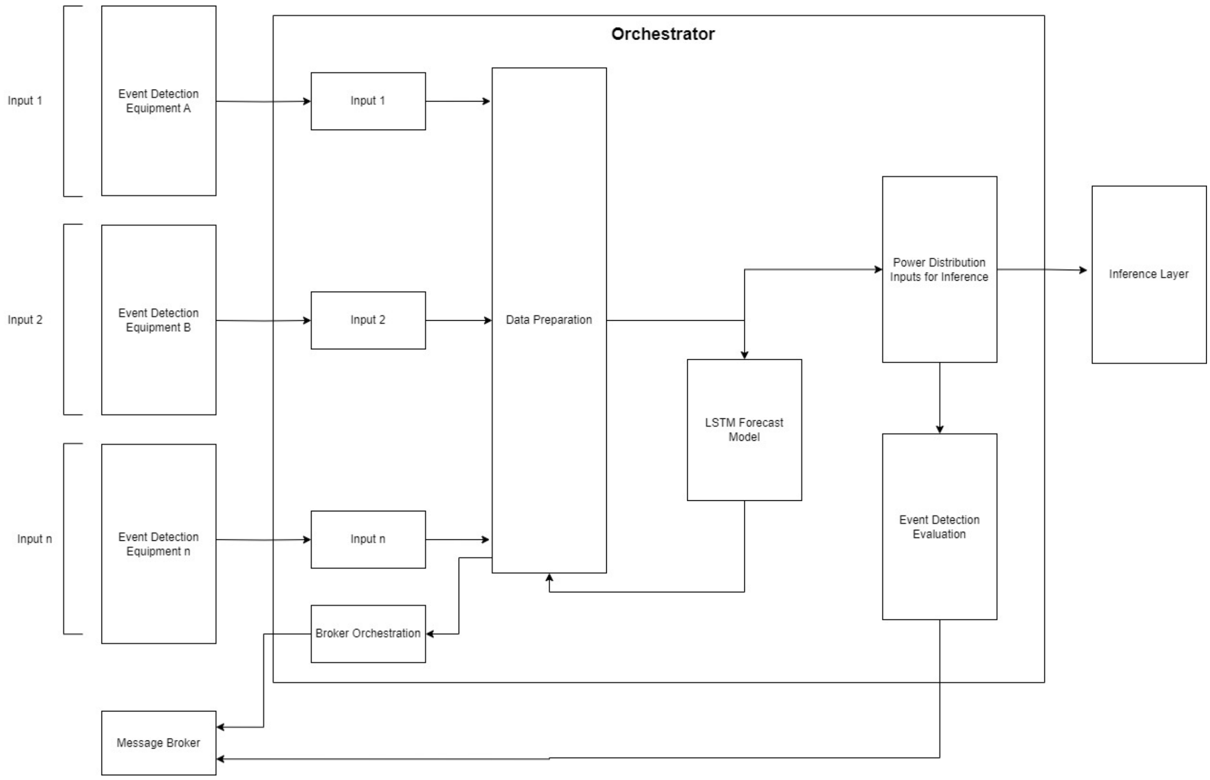
– Verify that events are sent to the ontology module and processed accurately.

These agents work collaboratively to ensure that data is efficiently processed, events are accurately detected and communicated, and the model remains scalable and adaptable to changes in the network. By distributing responsibilities among specialized agents, the Freya model enhances its reliability, performance, and ability to manage complex power distribution networks.

4.5 Orchestrator

Figure 14 illustrates the architecture of the orchestrator within the Freya model. The orchestrator manages event predictions, preparing data, and facilitating real-time and predictive analytics. The key components and their interactions are described as follows:

Figure 14: Orchestrator



Source: Prepared by the Author

The orchestrator receives event detections from pieces of equipment labeled Equipment A and Equipment B to Equipment N. Each input (Input 1, Input 2, up to Input n) represents data received from the event detection processes running on the equipment. Once the data is received, it is structured and prepared for further processing, ensuring it is in the correct format for immediate inference and predictive analytics.

The prepared data is sent to the inference layer (OntoFreya) to perform real-time inference based on the current state of the network. This step provides immediate insights and predictions regarding potential events. In parallel, the current data and previous data sequence classification probabilities are used by an LSTM (Long Short-Term Memory) forecast model to predict possible events in the next timestamps. The forecasted data is then re-prepared and sent for inference, similar to the real-time data. This iterative process, which can be repeated up to four times to forecast four samples in the future, showcases the system's ability to learn and improve its predictive capabilities over time.

When an event is detected, the broker orchestration component updates the publisher-subscriber queues. By default, an event considers one equipment before the entity with the event and one equipment after, totaling three pieces of equipment. However, it is possible to select two, three, or more pieces of equipment based on the requirements. The Event Detection Evaluation component, checks the event classification prediction against what actually happened. If an event reduces the accuracy, this component triggers retraining of the model, ensuring the system's reliability by addressing data drift, a common challenge in Machine Learning systems where data patterns change over time.

This detailed architecture ensures that the Freya model can handle real-time event detection and prediction while maintaining the flexibility to adapt to changing data patterns. The orchestrator's ability to manage data flow, update communication channels, and trigger model retraining ensures that the system remains robust and accurate over time.

4.5.1 Message Broker Service and Queue Orchestration

An aspect of the Freya model is how equipment sends its predictions to the following equipment or the server. This process happens by orchestrating queues in a message broker service.

As illustrated in Algorithm 1 and Figure 15, each piece of equipment publishes its predictions to a specific queue, and another piece of equipment subscribes to this queue to receive the predictions. The logic to maintain all equipment in the network publishing and subscribing is to have the number of queues equal to the number of equipment minus one. This setup ensures that each piece of equipment can communicate its predictions efficiently.

In order to keep a correct scalability level, the prototype increases the number of queues as necessary, accommodating the growing number of equipment in the network. A code example simulating this queue orchestration enlightens the data-sending process in the prototype. The pseudo algorithm derived from the simulation code demonstrates the main idea applied:

The algorithm "Simulate Equipment Updates"begins with the initial data organization (lines 1-2). A list named `data` is initialized, containing entries for each piece of equipment along with their respective publisher and subscriber queues. This list includes equipment such as *REC-001*, *VR-001*, and *AL-001*, each associated with specific queues for publishing and subscribing.

The next step defines the function `SIMULATE_UPDATES` (line 3) to simulate and print

Algorithm 1 Simulate Equipment Updates

Initial Data Organization:

Initialize data list with equipment and their respective publish and subscribe queues.

```
data = [
    {"Equipment": "REC-001", "publisher":
    "queue1", "subscriber": "queue2"},
    {"Equipment": "VR-001", "publisher":
    "queue2", "subscriber": "queue3"},
    {"Equipment": "AL-001", "publisher":
    "queue4", "subscriber": "queue5"}
]
```

function SIMULATE_UPDATES(equipment_list)

Create Mapping:

Create a mapping of equipment to their respective publish and subscribe queues.

```
eq_map = {item["Equipment"]:
{"publisher": item["publisher"],
"subscriber": item["subscriber"]} for item in data}
```

Extract Queue Information:

Extract necessary queue information based on the equipment list.

Let $first\_eq$, $second\_eq$, $third\_eq$ be the elements of $equipment\_list$.

$first\_publish\_to \leftarrow eq_map[first\_eq]["publisher"]$

$second\_subscribe\_to \leftarrow eq_map[second\_eq]["subscriber"]$

$third\_subscribe\_to \leftarrow eq_map[second\_eq]["publisher"]$

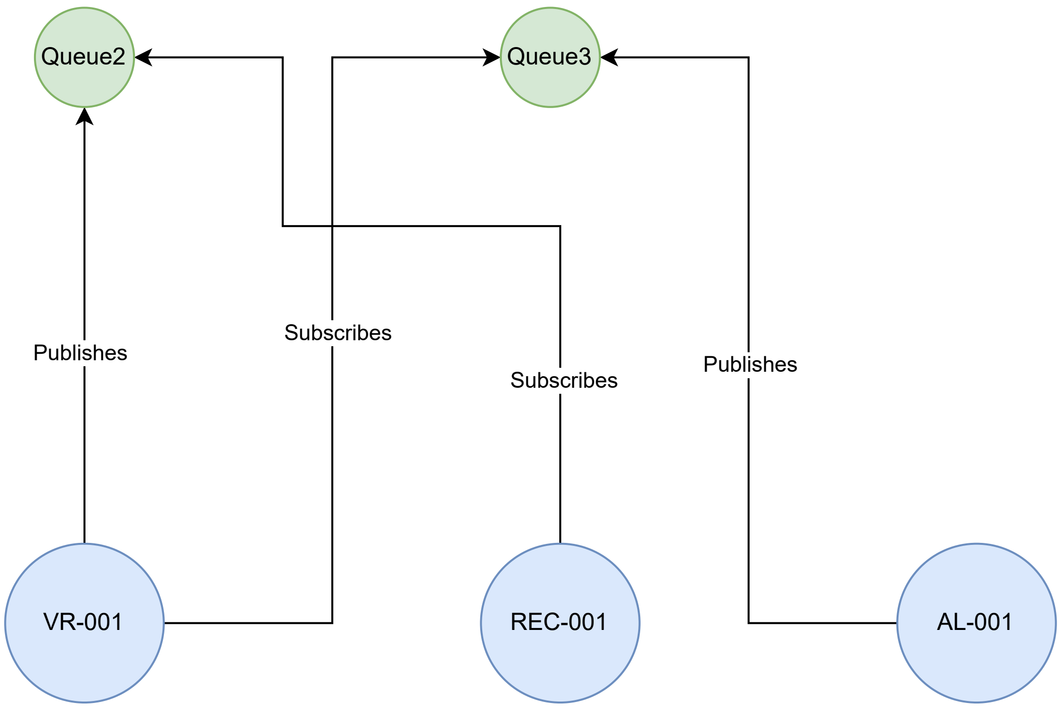
Simulate Updates:

Simulate updates by printing them.

```
print(f"{first_eq} will now publish to
{first_publish_to}, which is {second_eq}'s
current subscribe queue.")
print(f"{third_eq} will now subscribe to
{third_subscribe_to}, which is {second_eq}'s
current publish queue.")
```

end function

Figure 15: Communication between nodes orchestration



Source: Prepared by the author.

changes based on the provided equipment list. The function first maps equipment to their respective publish and subscribe queues (lines 4-5). This mapping is constructed using a dictionary comprehension that iterates over the `data` list, extracting the *publisher* and *subscriber* values for each piece of equipment.

Once the mapping is established, the function extracts the necessary queue information based on the equipment list passed as an argument (lines 6-12). The equipment list, named `equipment_list`, is assumed to contain three elements representing the equipment involved in the simulation. The function assigns these elements to variables `first_eq`, `second_eq`, and `third_eq`. It then retrieves the relevant queue information from the mapping: `first_publish_to` for the queue of the first equipment, `second_subscribe_to` for the subscribing queue of the second equipment, and `third_subscribe_to` for the subscribing queue of the third equipment.

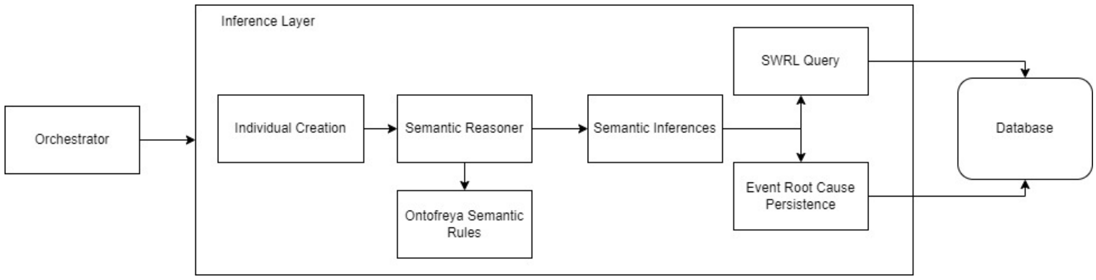
Finally, the function simulates the updates by printing the changes (lines 13-14). The first piece of equipment is set to publish to its designated queue, the current subscribing queue of the second piece of equipment. Simultaneously, the third piece of equipment subscribes to the publishing queue of the second piece of equipment. These prints clearly explain the new configurations resulting from the simulation.

This algorithm effectively demonstrates how equipment updates can be simulated and managed within a network, ensuring that each piece links correctly to its respective queues for publishing and subscribing. This approach ensures that the communication between equipment is handled efficiently and can scale with the increasing number of equipment in the network, maintaining the prototype's scalability.

4.6 Inference Layer

Figure 16 illustrates the architecture of the inference layer within the Freya model. The inference layer applies semantic reasoning to the event detection data from the orchestrator, enabling a more profound analysis and understanding of the network events.

Figure 16: Overview of Inference Layer



Source: Prepared by the author.

The process begins with the orchestrator sending data to the inference layer. The first step in the inference layer is the *Individual Creation* component, where individual instances are created based on the incoming data. These instances represent specific events or conditions detected in the network.

Next, the *Semantic Reasoner* processes the created individuals. This component applies the OntoFreya semantic rules to the data, enabling the generation of *Semantic Inferences*. These inferences provide valuable insights into the relationships and patterns within the data, going beyond simple event detection.

The results from the semantic reasoning process occur in two ways. First, a *SWRL (Semantic Web Rule Language) Query* executes to extract specific information from the inferred data. The results of this query provide detailed insights into the network's state and potential issues.

Second, the usage of data for *Event Root Cause Persistence*. This component identifies the root causes of detected events and ensures that this information is in the Database for future reference and analysis.

Finally, the *Database* component stores the SWRL query results and the root cause information. This Database is a comprehensive repository of historical data and inferred insights, supporting ongoing analysis and decision-making.

In summary, the inference layer of the Freya model leverages semantic reasoning to enhance the understanding of network events. This layer provides deeper insights into the network's behavior by applying semantic rules and generating inferences. It helps identify the root causes of detected events, ensuring that valuable information is preserved and utilized to improve network reliability and performance.

4.7 Machine Learning Algorithms and Predictions in the Freya Model

The Freya model employs Machine Learning algorithms to predict events in power distribution networks. Each piece of equipment within the network performs its predictions independently, selecting the best algorithm from a pool based on accuracy. When an event triggers, a different machine-learning approach is initiated.

Initially, each entity uses a single model for its predictions. This model is selected from a pool of algorithms, always choosing the one with the highest accuracy for the given data. Upon detecting a change in the operational state (an event), the entity sends a message to the orchestrator through a message broker. This message indicates the onset of an event.

Upon receiving the event message, the orchestrator alerts the neighboring entities closest to the originating entity. This notification is strategic for two main reasons:

- *Event Origin Detection*: Determining if the event started at the original entity or if issues influenced it in a previous entity.
- *Network Impact Assessment*: Assessing whether the event affects subsequent entities in the network.

During an event, each involved entity trains its best-performing model and stacks it with the models of its neighboring entities. This stacking process facilitates the sharing of event identification patterns through transfer learning. Transfer learning allows each model to learn from the patterns detected by its neighbors, enhancing the overall predictive accuracy. To prevent model mixing, an entity involved in an event cannot participate in another simultaneous event. This approach ensures that predictions remain relevant and specific to the current event.

The stacked predictions generate probabilities of the event occurring, ranging from 0% to 100%. The orchestrator collects these probabilities, creates a new dataset, and employs a Long Short-Term Memory (LSTM) approach for further analysis. The LSTM model uses these probabilities to predict the likelihood of an event occurring in the following 1 to 4 sample collections.

The LSTM forecasts provide valuable insights that the energy company's operational staff can monitor, enabling proactive measures to mitigate potential issues in the power distribution network.

The algorithm 2 summarizes the complete flow of the Freya model, encompassing the steps involved in event detection, prediction, and handling within a power distribution network. The algorithm integrates the processes of individual equipment prediction, event-triggered stacked learning, and long-term prediction using LSTM.

Algorithm 2 Machine Learning Event Detection and Prediction in Freya Model

```
Initialize Network Orchestrator
while network activity is ongoing do
    for each Entity in the network do
        Collect and prepare data from Entity
        Select best Algorithm from Pool based on accuracy
        Apply selected ML model to data
        Predict Entity State
        if change in state (event detected) then
            Send event message to Orchestrator via message broker
            Orchestrator alerts neighboring entities
            for each Neighboring Entity do
                Train best model with current data
                Stack model with neighboring entities' models
                Share event identification patterns through transfer learning
                Predict probabilities of event occurrence
                Send predictions to Orchestrator
            end for
            Create new dataset with probabilities
            Apply LSTM model for further event prediction
            Monitor LSTM predictions for future event occurrences
            if power metrics return to normal then
                Mark event as resolved
                Stop transfer learning process
            end if
        end if
        Save prediction results
    end for
    Update Network Orchestrator with latest predictions
    Store event information in Database
end while
```

The algorithm outlines the event detection and prediction process within the Freya model *initialize Network Orchestrator* (Line 1): the orchestrator is set up to manage the flow of information and coordinate actions within the network. *Continuous Monitoring* (Line 2): The system continuously monitors network activity, ensuring real-time responses. *Data Collection and Preparation* (Lines 3-4): Each entity in the network collects and prepares data for analysis and selects the most accurate algorithm from a predefined pool. *Prediction of Entity State* (Lines 5-7): The selected Machine Learning model is applied to the data to predict the entity's current state. *Event Prediction* (Lines 8-9): If a change in the operational state is detected, indicating an event, the entity sends an alert to the orchestrator via a message broker. *Neighboring Entities Notification* (Lines 10-11): The orchestrator then notifies neighboring entities to assess the event's broader impact. This approach helps determine whether the event originated from the initial or previous entities and evaluates its effects on subsequent entities. Stacked Learning and *Transfer Learning* (Lines 12-16): Each neighboring entity trains its best model with current data. Models from neighboring entities are stacked, enabling transfer learning to share event detection patterns. This process enhances predictive accuracy by leveraging shared knowledge. The stacked models generate probabilities of the event occurring, ranging from 0% to 100%. These predictions then go to the orchestrator. *LSTM Prediction* (Lines 17-20): The orchestrator creates a new dataset with these probabilities. An LSTM model is applied to predict the likelihood of future events further. The LSTM forecasts provide valuable insights that the energy company's operational staff can monitor, enabling proactive measures. *Monitoring and Resolution* (Lines 21-23): If power metrics return to normal, the event is marked as resolved, and the transfer learning process stops to prevent model mixing. *Result Saving and Orchestrator Update* (Lines 24-29): Prediction results are saved for continuous improvement and historical analysis. The orchestrator updates its records with the latest predictions. All event information at this point is in a database for future reference.

4.8 Chapter Considerations

This chapter provided an in-depth exploration of the Freya model, a comprehensive solution for Smart Grid event prediction and management. The discussion spanned multiple layers and components of the Freya model, showcasing its innovative architecture and capabilities.

The Freya Architecture section detailed the model's overall structure, emphasizing its microservices architecture based on the Model-View-Controller (MVC) standard. This architecture enables modularity and scalability, crucial for handling SG systems' dynamic and distributed nature.

The Edge Layer section highlighted the significance of performing computations at the network's edge. This layer reduces latency and improves real-time decision-making capabilities by processing data close to the source. The edge layer is responsible for initial data collection, preliminary processing, and forwarding relevant information to higher layers for further analysis.

In the Edge Agents Layer, the focus was on the intelligent agents deployed at the edge. These agents adapt and retrain predictive models based on local data, ensuring that the predictions remain accurate despite changes in the environment or network conditions. This adaptability is essential for maintaining the reliability and effectiveness of the Smart Grid.

The Orchestrator section elucidated the pivotal role of the orchestrator in the Freya model. It receives event detections from multiple equipment, structures, and prepares this data for the inference layer. The section also detailed the orchestration of message queues for efficient communication between different network components. The orchestrator's manages the flow of data, ensuring that predictions are made and acted upon in a timely manner, thereby enhancing the efficiency of the Smart Grid management.

Finally, the Inference Layer section described using the OntoFreya ontology to make inferences based on the processed data. This layer integrates the predictions from the edge and orchestrator layers, providing a holistic view of the network's status and potential future events. The inference layer's ability to continuously update and refine its models based on new data is strategic for maintaining high accuracy and reliability in event predictions.

This chapter demonstrated how the Freya model addresses the five gaps identified in the literature review: predicting entity events, performing network predictions, considering dynamic network layouts, transferring event patterns between entities, and automatically retraining models to prevent data drift. The detailed explanation of each component and layer of the Freya model illustrated its robustness and capability to enhance SG management through advanced Machine Learning and edge computing techniques.

5 ONTOFREYA: A POWER DISTRIBUTION ONTOLOGY FOR ELECTRIC METRICS CLASSIFICATION

This chapter presents an Ontology to classification of power metrics called OntoFreya. Therefore, this chapter presents the modeling and the results obtained with the classification of electrical metrics according to the readings of the electrical distribution network equipment. The main contribution of this approach lies in the automatic classification of energy metrics. Usually, specialists perform this task, and the automatic classification reduces the time-consuming activity. Particularly, the current literature does not propose a formal ontology for the power distribution domain. A power distribution ontology can help to represent the domain and map relevant information. In addition, the proposed ontology considers real data and the ambient equipment to perform inferences and classifications.

Power utilities demand large volumes of data used in power distribution networks. Among them are parameters representing possible technical failures, such as network's short circuit current and voltage sag. Specialists find these parameters and detect technical failures. However, this process can become time-consuming. Thus, this chapter proposes an ontology called OntoFreya, which classifies voltage, current, or any electric metric, following the definitions of the regulatory agencies and reducing the time spent on this task. A series of 4402 axioms, 132 classes, and 40 data proprieties comprises OntoFreya. The ontology automatically inferred classifications for four hundred readings from energy samples, validating OntoFreya across three scenarios. The first and second scenarios classified current in amperes, and the third classified voltage in per-unit system (pu). The scenarios showed that OntoFreya automatizes the classification of electric metrics, reducing specialists' time in detecting technical failures in a distribution network.

This chapter has four sections. Section 5.1 describes related works. Section 5.2 presents the ontology modeling process. Section 5.3 shows the results of the ontology validation, and, finally, the last section concludes with the final remarks of this chapter.

5.1 Ontology Related Works

The use of ontologies to support SG systems is an emerging field of research (PRITONI et al., 2021). Hence, this study adopted the Snowball Sampling (SS) technique (LEIGHTON et al., 2021). The SS allowed finding related studies that use ontologies in the SG domain. The SS technique involves reading the references in a paper and searching for works related to the research study. Performing the SS from a recent systematic mapping about SG and data analysis methods (ARANDA et al., 2022) allowed the search for the related works presented in this section.

Salameh et al. (2019) proposed an ontology to resolve the interoperability issues related to SG systems. Since each component of an SG could have syntactic and semantic differences, the authors use the ontology to unify SG components.

Schachinger et al.(2016) developed an ontology to work as a middleware between SGs and Building Energy Management Systems (BEMS). The ontology works as a semantic translator of the SG components to the BEMS, allowing SG data input through management systems.

Zanabria et al. (2019) presented the EMSOnto, an ontology to support SG automation systems. The authors previously created a Power System Automation Language (PSAL) to validate the inferences resulting in SG automation suggestions.

The study of YeeChong et al. (2020) approaches a methodology to create ontologies for SGs. Considering all the domain aspects of the SG, the author's method intend to become a gateway to semantically translating SG information.

The article of de Souza et al. (2024) introduced an agency-based design ontology called OntoAgency, which systematically traces relationships between stakeholders involved in smart building/Smart Grid services. This ontology links 'smartness' to various building operation domains and assigns functionalities to agents managing data flow and decision-making processes. This comprehensive approach offers a realistic and versatile model of inter-agent relationships, enhancing the understanding and management of smart building operations.

The article of Nepsha et al. (2022) discussed the growing use of the ontological approach in providing information support for the automation of technical systems, particularly in designing control systems for smart distributed energy systems. It highlights the challenges in developing these systems, mainly due to the need for a unified information environment for describing the subject area. Additionally, the article presents an ontology update process to ensure it meets the digital platform stakeholders' needs.

The criteria for performing *inferences*, use of *SPARQL* queries, *domain* application, type of *evaluation*, and consideration of *context-aware* data allow the comparison between OntoFreya and related works. All related works focus on SG systems in general, while OntoFreya focuses on energy classification of power distribution networks. Additionally, none of the works uses context histories to make inferences. Usually, the ontologies literature advises extending an existing ontology (LI; ARMIENTO; LAMBRIX, 2019), but in this case, the entities represent different domains, demanding the creation of a new ontology.

Regarding the related studies and the comparison criteria, five use inferences to support applications (SALAMEH; CHBEIR; CAMBLONG, 2019; SCHACHINGER; KASTNER; GAIDA, 2016; ZANABRIA et al., 2019; SOUZA; BADYINA; GOLUBCHIKOV, 2024; NEPSHA et al., 2022). Three studies use SPARQL queries (SALAMEH; CHBEIR; CAMBLONG, 2019; SOUZA; BADYINA; GOLUBCHIKOV, 2024; NEPSHA et al., 2022). All works applied ontologies in generic SG applications, energy generation, SG environments, and SG Automation. Two of the studies evaluate the ontologies with Synthetic data (simulated or manually inserted) (SALAMEH; CHBEIR; CAMBLONG, 2019; SCHACHINGER; KASTNER; GAIDA, 2016) and three uses third-party energy dataset (ZANABRIA et al., 2019; SOUZA; BADYINA; GOLUBCHIKOV, 2024; NEPSHA et al., 2022). Finally, neither of the works considers context data

Table 8: Comparison of Ontology related works

Related Works	**Inferences**	**SPARQL**	**Domain**	**Evaluation**	**Context Awareness**
Salameh et al. (2019)	Yes	Yes	SG Applications	Synthetic Data	No
Schachinger et al. (2016)	Yes	No	Energy Generation	Synthetic Data	No
Zanabria et al. (2019)	Yes	No	SG Automation	Energy Dataset	No
YeeChong et al. (2020)	No	No	SG Applications	No evaluation	No
Nepsha et al. (2022)	Yes	Yes	SG Automation	Energy Dataset	No
de Souza et al. (2024)	Yes	Yes	SG Environments	Energy Dataset	No
OntoFreya	**Yes**	**Yes**	**Power Distribution**	**Real Data**	**Yes**

Source: Prepared by the author.

regarding the SG entities.

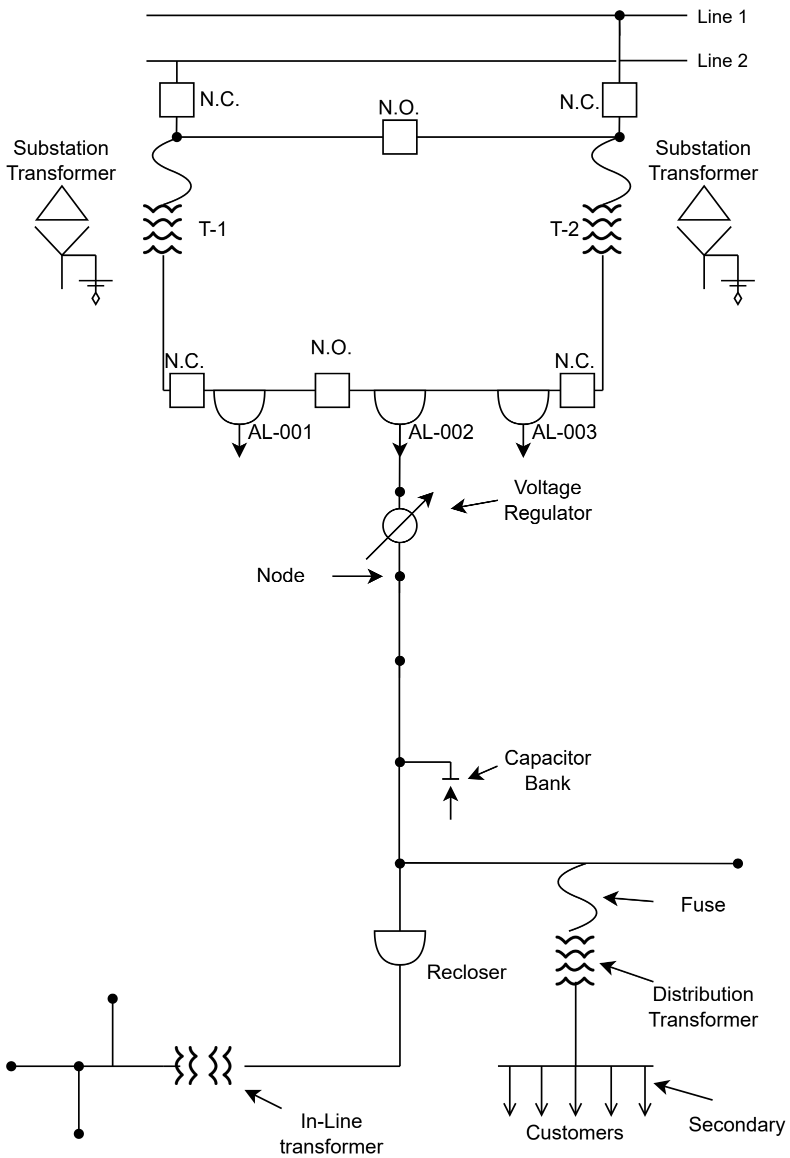
Furthermore, related works did not apply equipment data or real network data for validation. Since ontologies use domain data to reason about entities, this is also a differential of OntoFreya compared to the related works. Table 8 compares the OntoFreya with related works.

5.2 Ontology Modelling

According to Smirnov et al. (2021), the creation of an ontology is an iterative process, and the ontology concepts must be related to a domain. In order to adhere to the power distribution domain, the rules of the ontology follow the definition of regulatory agencies. Since the distribution network analyzed in this chapter is in Brazil, the rules consider the regulations proposed by the Brazilian National Agency of Electric Energy (ANEEL, 2021). Power distribution networks have different types of equipment. These equipments generate data that can be analyzed and estimated. The equipment with data available for analysis are reclosers, voltage regulators, capacitor banks, substation transformers, smart meters or digital meters, distribution transformers, and even distributed generators. Figure 17 shows an example of a distribution network showing the path taken by the distribution lines. First, the energy comes from one of the lines, represented by lines 1 and 2. Then it passes through a substation transformer to reach a feeder represented by the names AL-001, AL-002, and AL-003. The energy then passes through the voltage regulator, as well as through the capacitor bank. Finally, the energy flow can pass through a recloser or keep going to the consumers through a distribution transformer.

ANEEL defines voltage values ranging between *adequate*, *precarious*, and *critic* according to the reference value of the equipment. The voltage rule considers the per-unit system (pu).

Figure 17: Didatic Representation of a distribution network



Source: Adapted from Kersting (2017)

For instance, VR-001 has the referenced value of 7967 in medium voltage, so in this case, the value of 7967 is equal to 1 pu.

On the other hand, REC-001 uses parameters inserted in the supervisory system. According to the load balance estimation of the power utility specialists, this equipment has a pick-up value of 80 amperes. The supervisory system considers a heavy load if the current surpasses 55% of the pick-up value, in this case, 44 amperes. Moreover, the level of loads *Light*, *Medium*, and *heavy* allow performing inferences regarding the recloser. Table 9 shows the rule definition modelled into OntoFreya.

Figure 18 illustrates an overview of the OntoFreya ontology. Due to the size of the ontology,

Table 9: Rules modelled into OntoFreya

Scale	Unit	Rule
Recloser's Current		
Light	Current (% Pick-up)	C <= 35%
Medium	Current (% Pick-up)	C > 35% OR C <= 55%
Heavy	Current (% Pick-up)	C > 55%
Voltage Regulator's Voltage		
Adequate	Voltage (PU)	PU >=0.93 OR PU <= 1.05
Precarious	Voltage (PU)	PU >=0.90 OR PU < 0.93
Critic	Voltage (PU)	PU < 0.90 OR PU > 1.05

Source: Prepared by the author.

the overview only displays the main classes to increase readability. The OntoFreya is available at a GitHub repository [1]. The ontology development considered the types of equipment and entities of a distribution network as classes. Each class has a context, measured electric metrics, and other information that can help to infer the entity classification, like temperature and humidity. The main entities identified in the distribution network and modeled into classes were the following:

- *Voltage Regulator* responsible for preventing voltage sag;
- *Recloser* responsible for avoiding short-curt circuit;
- *Substation Transformer* changes the relationship between the incoming voltage and current and the outgoing voltage and current;
- *Weather Condition* represent the weather of the network in general;
- *Distributed Generation Unit* refers to technologies that generate electricity at or near the consumption, such as solar panels and combined heat and power.
- *Capacitor Bank* corrects a power factor lag or phase shift in an alternating current (AC) power supply;
- *Customer* represents customer relations in the power distribution network.

In OntoFreya, the main entities represent the equipments of the power distribution network. The subclasses of the main entities represent power metrics or status according to the class. Each subclass has the name of its superclass at the beginning of its name. The subclasses of the Voltage regulator are:

- *Voltage Regulator In* the amount of inlet energy in a voltage regulator;
- *Voltage Regulator Out* the amount of outlet energy in a voltage regulator;

[1] https://github.com/jsaranda/OntoFreya

- *Voltage Regulator Tap* changes the voltage ratio of a transformer so that its secondary voltage stays at nominal;
- *Voltage Regulator Power* the power of the energy rate transformed or transferred over time;

  Voltage Regulator Context Data data related to the equipment context.

Recloser subclasses are the following:

- *Recloser Context Data* data related to the equipment context;
- *Recloser Voltage* is the pressure from an electrical circuit's power source that pushes charged electrons (current) through a conducting loop;
- *Recloser Current* Current is the number of charges per unit of time passing through a boundary;
- *Recloser Power* is the power of the energy rate transformed or transferred over time.

Substation Transformer has the following subclasses:

- *Substation Transformer Power* the power of the energy rate transformed or transferred over time;
- *Substation Transformer Context Data* data related to the equipment context;
- *Substation Transformer Voltage* is the pressure from an electrical circuit's power source that pushes charged electrons (current) through a conducting loop;
- *Substation Transformer Tap* changes the voltage ratio of a transformer so that its secondary voltage stays nominal.

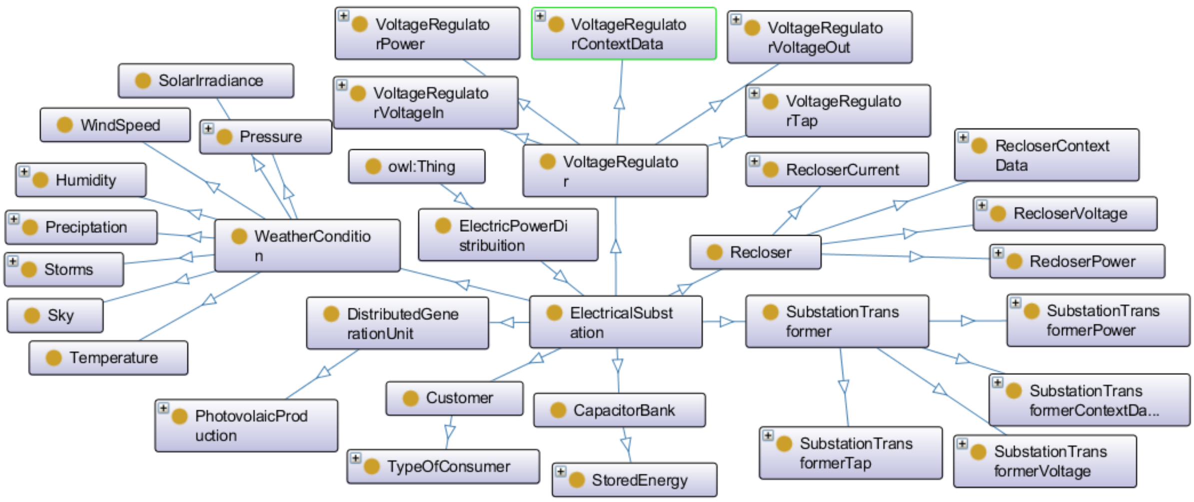
The weather condition class has the following status:

- *Temperature* the external temperature in Celsius degrees;
- *Sky* if the sky is clear or cloudy;
- *Storms* if a storm is occurring;
- *Preciptation* how much is raining;
- *Humidity* the relative humidity;
- *Wind Speed* the speed of the wind;
- *Solar Irradiance* the solar irradiance amount;

- *Pressure* the current pressure.

Finally, the *Capacitor Bank* class has the *Stored Energy* subclass that represents how much energy the capacitors store, the Consumer class has the *Type of Consumer* subclass representing the Customer, residential or industrial, and *Distributed Generation Unit* that has the *Photovoltaic Production* subclass measuring the generation of photovoltaic energy.

Figure 18: Overview of the classes and relationships of the OntoFreya ontology.



Source: Prepared by the author.

The distribution network in the current proposal has an energy sample collected by supervisory systems. Each reading has *In* and *Out* in pu (for the voltage regulator) or % of pick-up current (for the recloser), date of reading, entity name, temperature, and humidity values. The last two, regarding weather, can affect the electric readings and are considered a piece of important context information (RASTOGI et al., 2021). This energy sample presents a hierarchy of entities within a domain and the properties defined by attributes of a type value. In this scenario, OntoFreya can assign result analysis of electrical sample readings from the established rules and automatically infer the classification of this sample.

The OntoFreya ontology was developed in the OWL language using the software Protégé in version 5.5 (MUSEN, 2015). All readings were disjuncts within each class group: *Adequate*, *Precarious*, and *Critic* for voltage and *Light*, *Medium*, and *Heavy* for current. In this way, an energy reading has at least one value for each type.

Table 10 shows OntoFreya metrics extracted from the Protégé software and indicates the expressiveness of the ontology. Axioms are logical expressions that define a concept. Object properties indicate the relationships between instances of two classes, while data properties indicate relationships between class instances and literal data types. Total classes and total subclasses represent the number of elements found in the ontology, while individuals represent the instances populated to perform the inferences and queries of the SPARQL type. In these instances, electrical readings represent the real data from the equipment. The annotation is a free semantic element describing any ontology's feature or axiom.

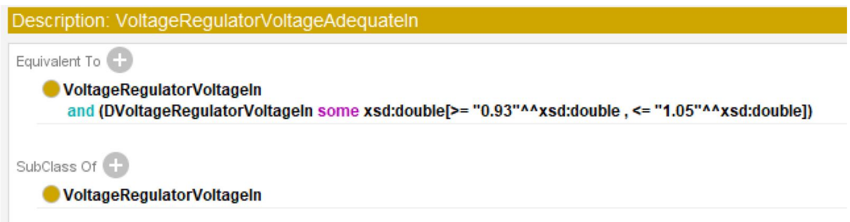
Table 10: OntoFreya Metrics

Métric	Value
Axioms	4442
Logical Axioms	3869
Number of Axioms Declaration	573
Number of Classes	132
Number of subclasses	131
Number of Object Propriety	1
Number of data Propriety	40
Number of individuals	400
Number of annotations	1

Source: Prepared by the author.

In OntoFreya, axioms define classes of electric metrics, converting amperes to a percentage of pick-up value for the recloser and pu for the voltage regulator. Table 9 shows the rules to classify the pu_{in} or pu_{out} into three levels according to the inlet and outlet voltages. For example, Figure 19 shows the logical expression used in the equivalence axiom for that calculation. Similar expressions define the *Critic* and *Precarious* levels, according to the ranges presented in Table 9, in this specific example and for validation purposes.

Figure 19: Logical expression in axiom for voltage regulator inference.

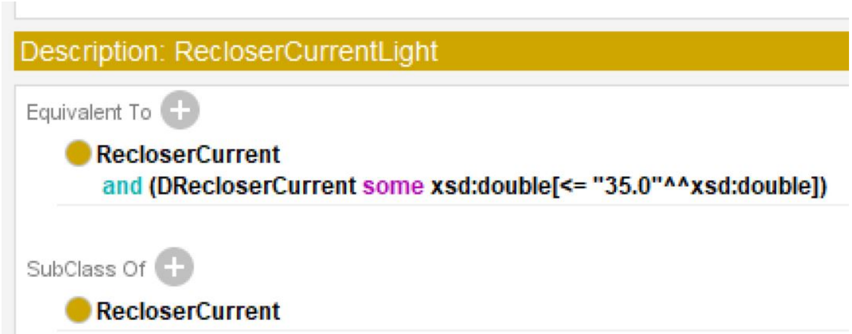


Source: Prepared by the author.

Similarly, the recloser class applies the rule regarding the current reading. Table 9 displays that the recloser rules have different thresholds to infer from energy samples. Figure 20 shows the logical expression of the rule applied to recloser readings at the *Light* level.

OntoFreya also considers the context information of the recloser, the voltage regulator, or other main classes presented in this section. Each main class has the subclass *Context Data*, which contains temperature and humidity logical axioms to *Operational Temperature/Humidity* or *Non Operational Temperature/Humidity*. Figures 21 and 22 exemplify the inferences regarding context information according to the effects on the power distribution network(RASTOGI

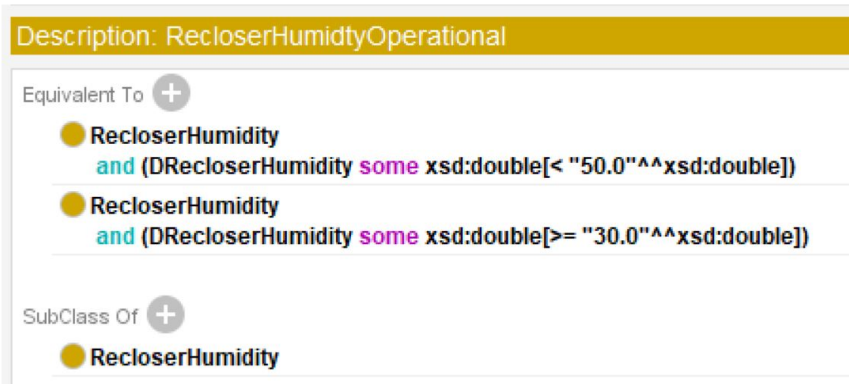
Figure 20: Logical expression in axiom for recloser current inference.



Source: Prepared by the author.

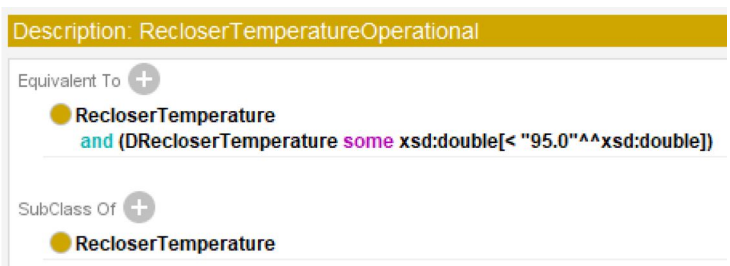
et al., 2021).

Figure 21: Logical expression in axiom for recloser humidity inference.



Source: Prepared by the author.

Figure 22: Logical expression in axiom for recloser temperature inference.



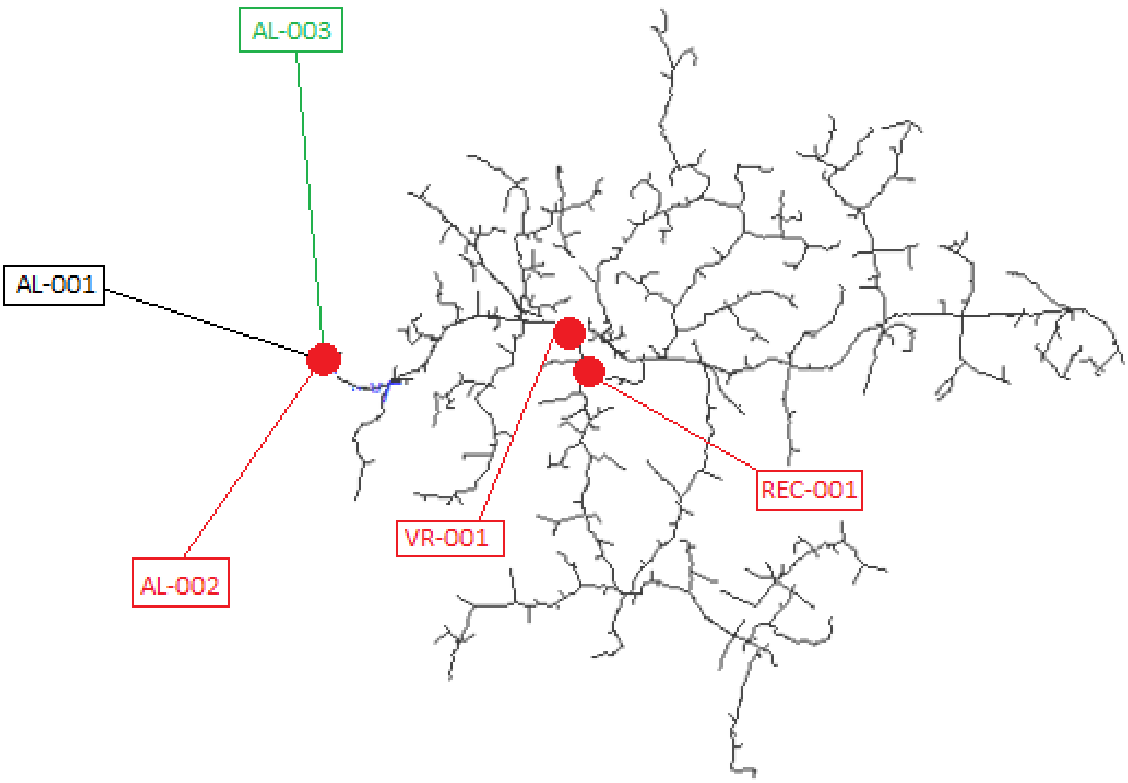
Source: Prepared by the author.

5.3 OntoFreya Evaluation

This thesis evaluates OntoFreya based on scenarios. Scenarios are strategies to evaluate ubiquitous applications and context-aware systems (RENTZ; HECKLER; BARBOSA, 2023). Three scenarios applied in the distribution network presented in Figure 23 were considered in the evaluation. The first and second scenarios consider the recloser data, while the third scenario consider the voltage regulator.

The distribution network has different types of equipments, but this study employed the voltage regulator VR-001 and the recloser REC-001. These two equipment are 10.5 km from the substation that serves three primary feeders - AL-001, AL-002, and AL-003. Figure 23 presents a representation of the power distribution network analyzed in this chapter.

Figure 23: Representation of the analyzed distribution network.



Source: Prepared by the author.

The first scenario comprises the recloser equipment REC-001. The second scenario is similar to the first one, with different parameters. A change in parameters could show other outputs regarding the recloser equipment. Finally, the third scenario evaluates the ontology with the voltage regulator VR-001.The data of the equipments are stored and provided in excel files. Adding the context data (temperature and humidity) from a weather API [2] allows the context

[2]https://www.meteomatics.com/en/weather-data/

inferences.

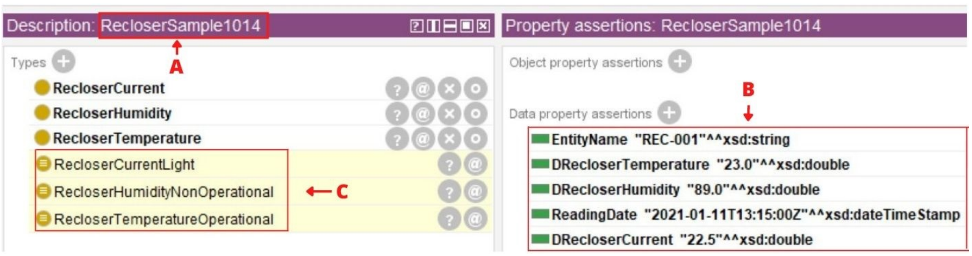
Due to performance limitations on the protégé software, OntoFreya can only create 200 energy samples for each type of equipment, totaling 400 individuals. Finally, a python script adds each individual to the ontology XML file. The following subsections show the results of the inferences in the scenarios mentioned earlier.

5.3.1 Scenarios 1 and 2 - Recloser

The creation of instances called individuals in the protégé software covers the range of load readings classification of a recloser. The evaluation takes place with the automatic reasoning process. Then the logical expressions process each individual with the reasoner HermiT, an ontology reasoner for the OWL language included in the protégé.

Figure 24 shows the result for the energy reading classification of the sample 1014, the letters indicate strategic information in the figure. One of the individual instances of the readings is the *Recloser Sample 1014* (A), which has an entity name of *REC-001*, temperature of *23*degrees celsius, *89%* of humidity, the date of the reading of *11-January-2021 at three o'clock* and a current of *22.5%* of the pick-up value (B). The classes *Recloser Current Light*, *Recloser Humidity Non-Operational*, and *Recloser Temperature Operational* are the results of this individual inference (C).

Figure 24: Recloser inference result for Scenario 1.

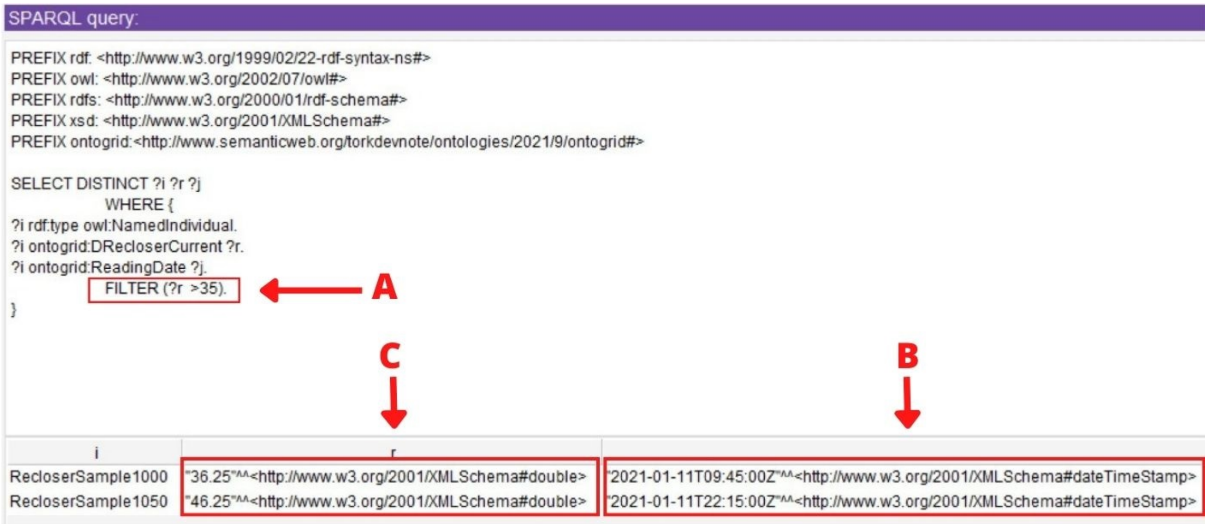


Source: Prepared by the author.

This inference considers the rules presented in Table 9. The *Recloser Current Light* class represents readings with a current below 35% of the pick-up value. The *Recloser Current Medium* represents readings with a current between 35% and 55% of the pick-up value. Finally, the *Recloser Current Heavy* represents the readings with a current over 55% of the pick-up value.

In addition to inferences related to the electric metrics classification, OntoFreya allows domain representation to other languages such as RDF that queries a consult with SPARQL, similar to SQL, but with greater expressiveness. Figure 25 illustrates a typical search for instances classified as *Recloser Current Heavy* or *Recloser Current Medium*. The classification rules were in the query's filter field in this case (A). This energy status happened within along with all the readings, on 11-January-2021, one at 09:45 and another at 22:15 (B). This search example allows the specialists to identify when one of the equipment is with a *Medium* or *Heavy* load (C).

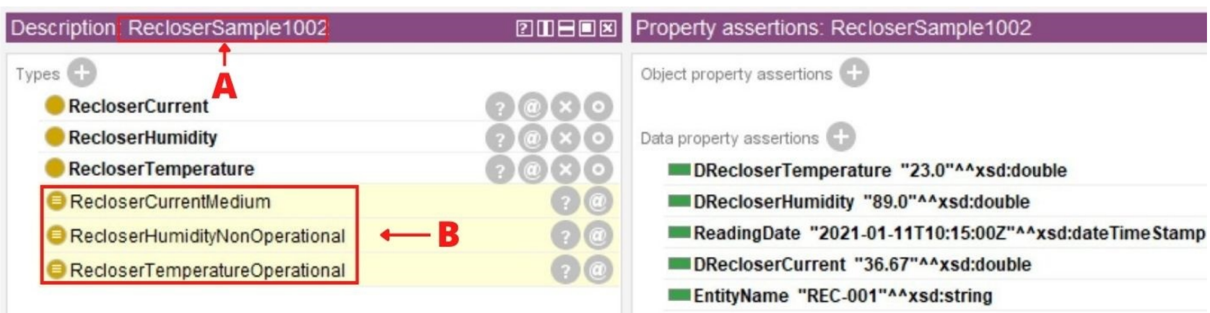
Figure 25: SPARQL query based on inference rule for scenario 1.



Source: Prepared by the author.

Since only two of 200 readings are under *Recloser Current Medium* or *Recloser Current Heavy* classification, most readings are under the *Recloser Current Light* classification in Scenario 1. This result means that a Recloser with a pick-up value of 80 is a good choice for this network. Alternatively, for Scenario 2, the specialists consider a Recloser with a pick-up value of 60 instead of 80. This change means that the 55% of the current threshold decreases and affects every classification. Nonetheless, all rules from Scenario 1 apply to Scenario 2 – except the pick-up value. Figure 26 shows the inferences about sample 1002. In this new round of inferences, the *Recloser Sample 1002* (A) classification changed from *Light* to *Medium* (B).

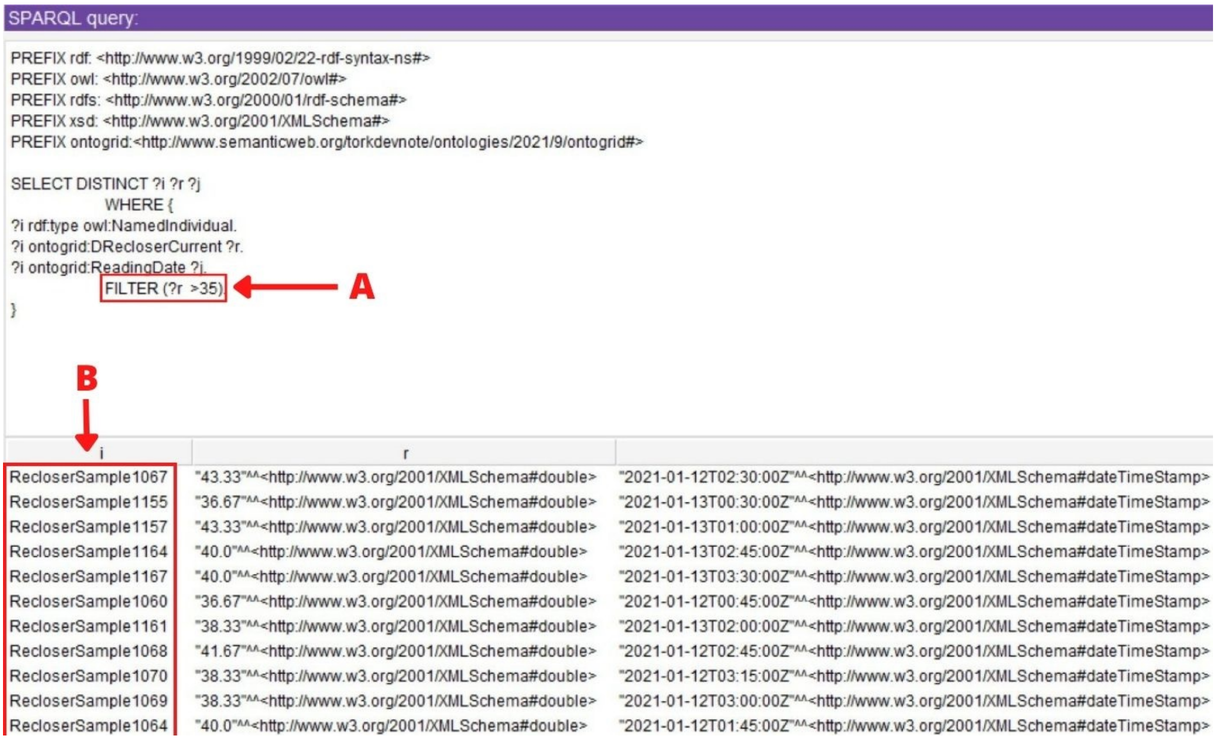
Figure 26: Recloser inference result for Scenario 2.



Source: Prepared by the author.

Figure 27 illustrates a new search for instances classified as *Recloser Current Medium* or *Recloser Current Heavy* (A). Again, the classification rules were in the query's filter field. This time 35 readings match the *Medium* or *Heavy* status (B) compared to only 2 in Scenario 1. This result shows that OntoFreya can help specialists find issues when the network equipments parameters are not well estimated – such as the pick-up value for the recloser. The use of the historical record of equipments as individuals of the ontology allows the representation of the domain's current behaviors.

Figure 27: SPARQL query based on inference rule for scenario 2.




```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontogrid:<http://www.semanticweb.org/torkdevnote/ontologies/2021/9/ontogrid#>

SELECT DISTINCT ?i ?r ?j
            WHERE {
?i rdf:type owl:NamedIndividual.
?i ontogrid:DRecloserCurrent ?r.
?i ontogrid:ReadingDate ?j.
            FILTER (?r >35)
}
```

i	r	
RecloserSample1067	"43.33"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-12T02:30:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1155	"36.67"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-13T00:30:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1157	"43.33"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-13T01:00:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1164	"40.0"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-13T02:45:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1167	"40.0"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-13T03:30:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1060	"36.67"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-12T00:45:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1161	"38.33"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-13T02:00:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1068	"41.67"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-12T02:45:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1070	"38.33"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-12T03:15:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1069	"38.33"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-12T03:00:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>
RecloserSample1064	"40.0"^^<http://www.w3.org/2001/XMLSchema#double>	"2021-01-12T01:45:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTimeStamp>

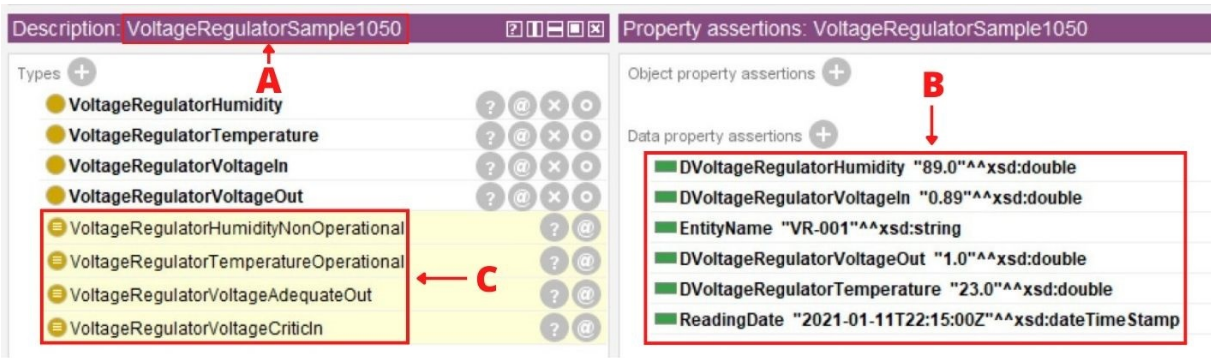
Source: Prepared by the author.

5.3.2 Scenario 3 - Voltage Regulator

Scenario 3 has 200 instances of voltage inlet and outlet readings converted to pu. This difference happens because the voltage regulator has pu_{in} and pu_{out} readings. When the recloser's purpose is to avoid high current load in the network, the voltage regulator receives a *critic* or *precarious* value of voltage in the pu_{in} of the equipment and regulates the voltage giving the network an *adequate* value of pu_{out}.

Figure 28 shows the result for the energy reading classification of the sample 1050. One of the individual readings instances is the *Voltage Regulator Sample 1050* (A) with *89%* of humidity, *Voltage In* of *0.89* pu, an entity name of *VR-001*, *voltage Out* of *1.0* pu, a temperature of *23* degrees celsius, and the date of the reading of *11-January-2021 at 22:15* (B). The classes *Voltage Regulator Humidity Non-Operational*, *Voltage Regulator Temperature Operational*, *Voltage Regulator Voltage Adequate Out*, and *Voltage Regulator Voltage Critic In* are the results of the inference of this individual in Scenario 3 (C).

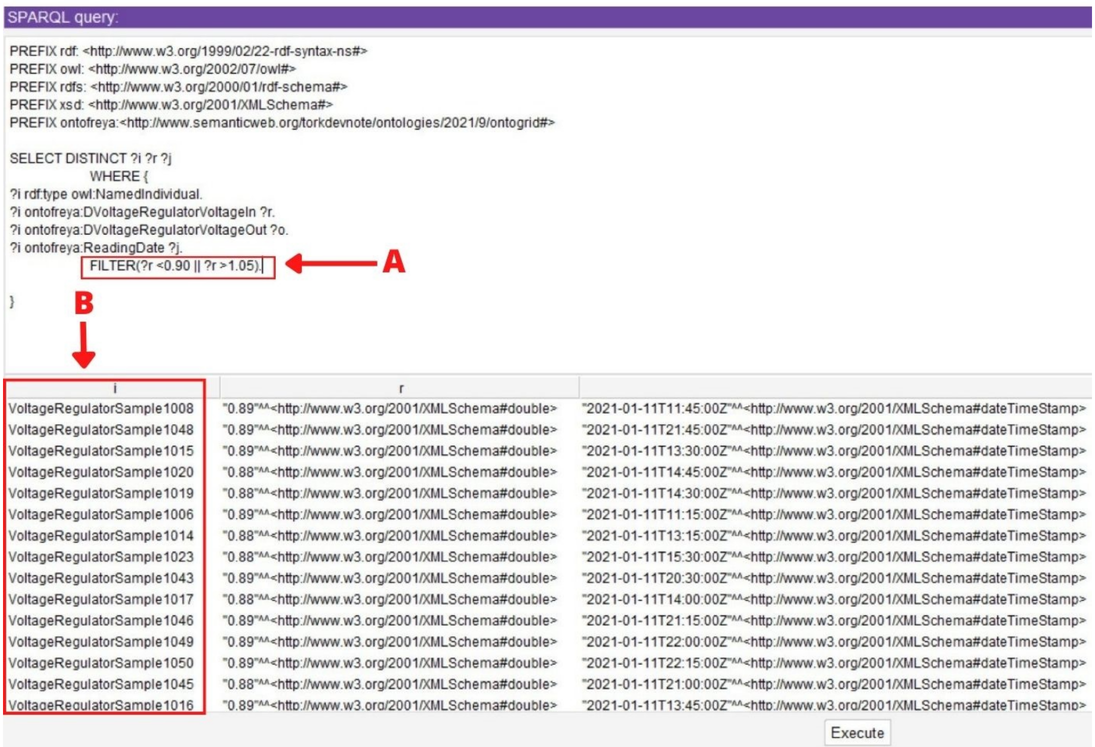
Figure 28: Voltage Regulator inference result for Scenario 3.



Source: Prepared by the author.

Proceeding with the same validation patterns of Scenarios 1 and 2, Figure 29 shows a search for instances classified as *Voltage Regulator Voltage Critic In* (A). The search for a status applying the rules in the query filter identified 25 results (B).

Figure 29: SPARQL query based on inference rule for scenario 3.



Source: Prepared by the author.

5.3.3 Discussion

The advantages of OntoFreya lie in the automatized classification of the power distribution network values. After the insertion of the rules in the ontology, the reasoner quickly performs the inferences and provides the output class results. Another advantage is the proposal of an ontology oriented to the power distribution domain. Specifying a domain allows a better representation of entities, showing possible status or conditions that these entities are. The generic modeling of the OntoFreya ontology allowed changes in the reference value and a new round of inferences, as shown in Scenarios 1 and 2.

One of the differentials of the OntoFreya is the inference of context information. The humidity and temperature of a recloser or a voltage regulator show how this context can affect the energy consumption (SARKODIE; AHMED; OWUSU, 2021). Recent works about ontologies call this type of ontology as Ontology-Aware (SKRETA et al., 2021). This definition means that the ontology can adequately work with context-aware systems.

The most relevant information added to the ontology are the date and hour of an energy reading, the temperature, and the humidity of these entities. The climate is also essential for detecting solar irradiance, as it influences the generation distribution units. The weather information is relevant since the power distribution network is widespread. This way, the contexts could change if they are far from each other. Since there is a specific class for context data, this information type can scale according to the network needs. This context data can also use information from the Internet of Things sensors or devices installed on each equipment. Hence, OntoFreya can work with fog or edge computing systems, a well-known technique used in SGs and distribution networks (KULKARNI et al., 2019).

The scenarios used to validate the ontology show that not only a single instance can present a valid output, but the SPARQL query can also retrieve a batch of readings that match a specific rule or filter. For instance, if an specific equipment is not performing well, specialists can query search for the hour of the day when the equipment presents this behavior. Alternatively, the specialists can query a search with the temperature and humidity to show how these phenomena affect an entity in the domain.

However, OntoFreya currently presents limitations. Unfortunately, running inside the protégé software reduces the number of readings the ontology can infer. One possible solution to this problem is to develop a specific architecture to implement the ontology behavior. A new implementation could improve the performance when using large amounts of data.

Finally, the rules inserted in the ontology follow the guidelines of a regulatory agency or strategic limits from the utility company. The generic nature of OntoFreya allows the use in any network, only converting the network's rules to semantic rules. The search and indications of problems through the inferences can alert the specialists to act proactively in the network equipment. This action can reduce problems that, if not addressed, can generate fines from regulatory agencies.

5.4 Chapter Considerations

OntoFreya classifies and infers data from distribution network readings as current, voltage, and contextual information. The ontology is generic since each network has rules regarding classes, allowing new relationships between the terms as necessary. Through the process of a reasoner, using axioms and rules based on OWL allows a series of precise and automatic inferences and queries based on instances. The data from these instances are from actual electrical energy readings provided by a partner company of this study.

This study confirmed the potential of ontologies in electrical engineering as an efficient tool for energy reading classification, which can later be configured as part of an Information System in a decision support platform. Future works may translate the ontology to another programming language to create a microservice used to support the inference of electrical readings in real-time, providing inferences for an operator or other electrical engineering professional responsible for monitoring a distribution network.

6 EVALUATION ASPECTS

This chapter describes the implementing and evaluating the Freya model. Section 6.1 details the prototype implementation. Section 6.2 introduces the methodology for the model evaluation. Section 6.3 shows the results and section 6.4 presents a discussion about this results. Finally, section 6.5 concludes this chapter.

6.1 Prototype Implementation Aspects

The prototype's development have three stages. The first step involves developing the Edge Layer module and the agents for adapting and retraining the predictive model. The second step is to develop the Orchestrator Layer Module. This module receives the predictions made in the first step as input. The third module implements the Inference Layer, which consists of the OntoFreya ontology, for inference from the predicted values in the first and second steps.

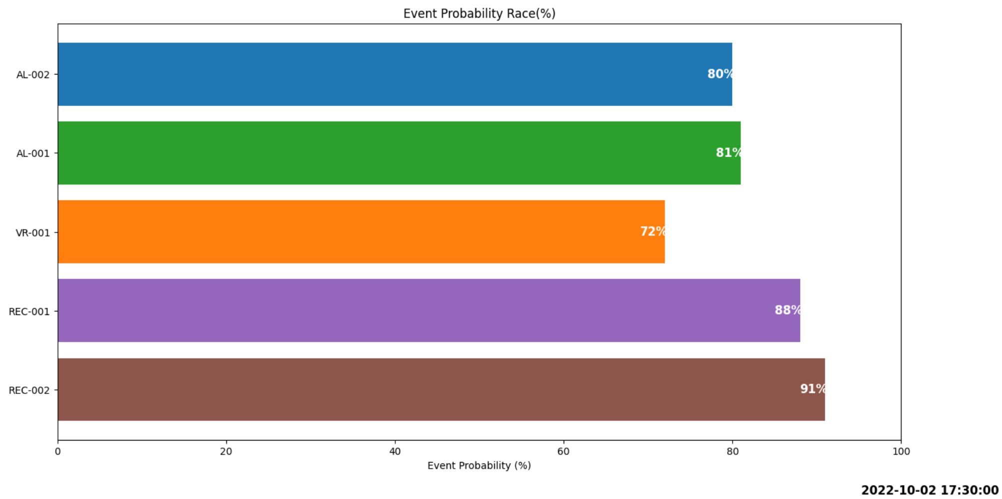
The items proposed in the three steps works through a microservices architecture based on the MVC (Model View Controller) standard. The technologies used in the three modules was developed in Python, utilizing state-of-the-art Machine Learning libraries (tensor flow, scikit learning), Owlready2 (for ontology manipulation), and the Flask framework to build the backend. The implementation employed Javascript, HTML, and CSS for the front end. The third step also considers setting up a NoSql database for data persistence, ensuring a comprehensive and advanced technological approach.

In order to present a summary of all of the prototype output, a UI (User Interface) was developed to present an event probability in the form of a race bar chart. The UI also presents a ranking of event probability based on criticism by equipment/entity. The criticism could be defined manually by the user or changed dynamically, depending on the context of the equipment. For example, if a region with one piece of equipment faces a storm, this equipment could increase its criticism temporarily.

Figure 30 illustrates the event probability ranking for various pieces of equipment in the network, showing the likelihood of each piece of equipment experiencing an event at the next timestamp. The equipments in this image are labeled as AL-001 (feeder), VR-001 (voltage regulator), AL-002 (feeder), REC-002 (recloser), and REC-001 (recloser), with their respective event probabilities displayed as horizontal bars. This chart uses operational data from samples saved in a dataset as input with a simulation predicting the events, in this case, an event that happened on 02-10-2022 at 17-30-00. The prediction at the edge layer detects if the equipment has an event when the equipment gets the sample read.

The prediction at the orchestrator level predicts steps; these steps could vary according to the time between each sample collection; for example, if the sample collection occurs every 5 minutes, four steps ahead predicts 20 minutes. The collection during this evaluation and used by the energy company was 15 minutes. Each event prediction should define the periodicity of sample collection on a case-by-case basis (SHEHU; HARPER, 2023). The study emphasizes the importance of characterizing individual data streams to effectively deploy the most appropriate algorithm for real-time anomaly detection. By leveraging a periodicity detector based on the Lomb-Scargle periodogram, the research demonstrates how efficient identification of periodicity impacts algorithm choice, highlighting the necessity of a tailored approach for each data stream. This methodology ensures optimal performance in detecting anomalies or predicting events, which can vary significantly across different contexts and applications (SHEHU; HARPER, 2023).

Figure 30: Probability of Event on each equipment



Source: Prepared by the author.

In this figure, the probability values rank the equipment, highlighting those most likely to present an event. This ranking allows for quick identification of the equipment at the highest risk, facilitating proactive measures to address potential issues.

All prediction values are collected from the Orchestrator side (server) and can be sorted to generate rankings. This process is crucial for real-time monitoring and decision-making in the network.

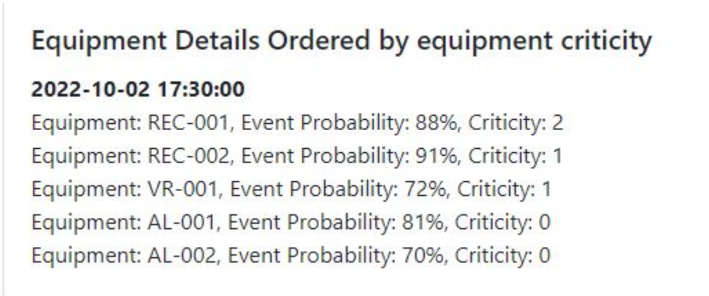
The prototype's capability extends beyond merely generating classification predictions. It utilizes the input probabilities from the classification predictions and feeds them into a neural network with Long Short-Term Memory (LSTM) units. This LSTM network is employed to forecast event probabilities for future timestamps.

By leveraging the LSTM network, the prototype can predict future event probabilities, enabling advanced warning and better preparation for potential network disruptions. This approach combines real-time classification with forecasting, providing a comprehensive network reliability and performance enhancement solution.

The timestamp at the bottom right corner of the figure indicates the specific time at which the event probabilities could occur, ensuring accurate tracking and temporal context for the predictions.

Figure 31 shows a user interface card that displays the event probability for various pieces of equipment, sorted by their criticity level. The timestamp at the top indicates when the event probabilities calculations occur, ensuring accurate tracking and temporal context for the predictions.

Figure 31: Probability of Event on each equipment sorted by criticity.



Source: Prepared by the author.

The criticism level indicates the importance or severity of each piece of equipment. Criticism can be either static or dynamic, calculated based on various contextual aspects, including the equipment's historical performance, current operational status, and weather conditions. This sorting mechanism allows for the prioritization of equipment that requires immediate attention based on the event probability and the criticism level. For instance, REC-001, with a criticism level of 2 and an event probability of 88%, is flagged as critical, requiring attention.

By incorporating static and dynamic factors into the criticism calculation, the system ensures a comprehensive assessment of the equipment's condition. This approach allows for a more accurate and context-aware prioritization, improving the network's reliability and performance. The prototype's ability to dynamically adjust criticism levels based on real-time data and context ensures that the most critical equipment highlights it, enabling timely interventions and proactive maintenance. For a better understanding, this simulation is available at: https://jsaranda.github.io/FreyaUI/

6.2 Evaluation Methodology

The Freya model evaluation is conducted through a system simulation with operational data. The simulations operate through scenarios corresponding to the operation of a power distribution company. These scenarios are based on the Certaja power utility network, demonstrating that the Freya model is adaptable to any distribution network. The evaluation results are derived from the accuracy and precision of the prediction algorithms, with these metrics indicating whether the proposed technology can enhance the tasks performed by the operation of power utilities.

The evaluation process involves the following equipments: *AL-001, AL-002, AL-003, VR-001, REC-001, REC-002, REC-003, REC-004, REC-005, and REC-006*. This selection includes three feeders (*AL-001* to *AL-003*), one voltage regulator (*VR-001*), and six reclosers (*REC-001* to *REC-006*). It is important to note that the names and locations of some of the equipment have been modified or omitted due to company privacy issues.

This thorough evaluation involved a detailed data collection process, with a simulation of the equipment's operation based on data collected over a year. Each dataset for each piece of equipment comprises an average of 40,000 records, ensuring extensive data availability for this evaluation. The following five scenarios represent specific operational patterns of the power distribution network. Here is the explanation of each scenario and the equipment involved:

First Scenario: This scenario considers a regular operation related to the following equipments: *AL-002, REC-005, and REC-003*.

Second Scenario: This scenario considers a different part of the network, with events occurring between: *AL-001, VR-001, and REC-001*.

Third Scenario: This scenario involves more equipment and covers a broader part of the network. The equipment for this events includes: *REC-002, VR-001, REC-006, REC-004, and REC-005*.

Fourth Scenario: This scenario considers an inversion in the regular flow of the network due to a temporary network adjustment involving: *REC-006, VR-001, and REC-002*.

Fifth Scenario: Similar to the fourth scenario, this scenario also considers an inversion in the regular flow of the network, with the sequence of equipment being: *VR-001, REC-003, and REC-005*.

These scenarios are designed to represent different network conditions, such as regular operation (Scenarios 1 and 2), regular operation with an event occurring in the broader set of equipment (Scenario 3), and a temporary network configuration due to maintenance or the unavailability of a regular energy flow (Scenarios 4 and 5). The role of scenarios in understanding and interpreting these events is strategic, as they help understand the pattern of events in various situations. The selected equipment represent parts of the network with events to predict. The events predicted by the simulation are compared to the actual events or sudden data changes mapped by the energy company. This comparison helps validate the accuracy and effectiveness of the Freya model in real-world scenarios.

The methodology and the prototype were developed through the steps described with the intent to validate this work. The implemented prototype and the evaluation methodology answer the research question: *"How can a computational model be developed to evaluate monitoring data in a Smart Grid to predict network events, considering the operational hierarchy among different pieces of equipment?"*

By developing the edge-computing component for event prediction, creating a model for event prediction based on equipment and its context histories, building an ontology for power metrics classification, and evaluating the model through different scenarios, in order to fulfill the thesis objectives:

- Create a computational model called Freya for event prediction's in Smart Grids
- Perform a literature review of computing techniques that support Smart Grids;
- Create an edge-computing component to perform event prediction in power distribution, according to the equipment context histories and at the edge of the Smart Grid;
- Propose a model for event prediction based on the energy flow and context of equipment within a power distribution Smart Grid;
- Build an ontology for power metrics classification according to the event of the equipment on the edge of the grid;
- Evaluate the Freya model through operational scenarios.

Achieving the objectives requires a definition of metrics to evaluate the prediction models. The following sections present tables containing performance metrics for Machine Learning models evaluated on a classification task. Each table presents metrics for understanding the performance of the models. The chosen metrics (Accuracy, Precision, Recall, F1 Score, ROC AUC, Computational Time, and Score) provide an understanding of the evaluation process:

- **Accuracy**:
  - **Definition**: Accuracy is the ratio of correctly predicted instances to the total instances. It is a measure of how often the model makes correct predictions.
  - **Formula**:
$$\text{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{Total Instances}}$$
- **Precision**:
  - **Definition**: Precision is the ratio of correctly predicted positive observations to the total predicted positives. It indicates how many of the predicted positives are positive.
  - **Formula**:
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$
- **Recall**:
  - **Definition**: Recall, also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all observations in the actual class. It measures the ability of the model to identify all relevant instances.

– **Formula**:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

- **F1 Score**:
  – **Definition**: The F1 score is the harmonic mean of precision and recall. It balances precision and recall, especially when the class distribution is imbalanced.
  – **Formula**:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$$

- **ROC AUC (Receiver Operating Characteristic - Area Under Curve)**:
  – **Definition**: The ROC AUC score represents the area under the ROC curve, which plots the true positive rate against the false positive rate at various threshold settings. A higher AUC indicates better model performance when distinguishing between classes.
  – **Interpretation**: Values range from 0 to 1, where 1 indicates perfect classification, and 0.5 indicates no better performance than random guessing.
- **Computational Time**:
  – **Definition**: Computational time refers to the total time required to train the model and make predictions. It is an important metric for evaluating the model's efficiency, especially in real-time applications.
  – **Measurement**: Measured in seconds, this metric provides insights into the computational resources required for each model.
- **Score**:
  – **Definition**: The Score metric represents an average of Accuracy, Precision, and Recall. This composite metric provides a single value summarizing the model's overall performance.
  – **Formula**:

$$\text{Score} = \frac{\text{Accuracy + Precision + Recall}}{3}$$

Each scenario evaluation considers the metrics mentioned earlier and different models. In the simulation from each scenario, a comparison between the base models (the models that could be stacked) and stacked models shows whether the proposed stacked approach got a better result than conventional Machine Learning models. The stacked models are also used with Grid Search, a technique to get better hyperparameters for the model. The explanation of each event prediction model simulated is the following:

- **3 level-Multi-Layer Stacked with GS**:

  - This predictor represents a three-level multi-layer stacking model that includes grid search (GS) for hyperparameter tuning. The model stacks predictions from multiple base learners to enhance its metrics. In this case, the three represent the number of stacked equipment models. This approach means that this model knows the patterns of three network entities. The number of entities could vary according to the user's needs.

- **3 level-Multi-Layer Stacked**:
  - This predictor indicates a three-level multi-layer stacking model without grid search.

- **Single-Layer Stacked**:
  - This model uses a single-layer stacking approach, combining predictions from multiple base models.

- **Single-Layer Stacked with GS**:
  - This predictor presents the single-layer stacking model with grid search for hyperparameter optimization.

- **Base Model - Decision Tree**:
  - A decision tree model as a baseline for comparison.

- **Base Model - Random Forest**:
  - Random Forest, an ensemble of decision trees, is another baseline model.

- **Base Model - KNN**:
  - The k-Nearest Neighbors (KNN) model serves as another baseline.

- **Base Model - XGBoost**:
  - This predictor features the XGBoost model, which is used in Machine Learning classification tasks.

- **Base Model - SGD**:
  - The Stochastic Gradient Descent (SGD) model is the final baseline model used for classification tasks.

6.3 Results

This section presents the results obtained for each scenario. A table with the results of each model and the metrics is first shown. Then, an ontology of the current event is presented, showing the energy flow and power distribution during the event. The created ontology could infer the event's possible root causes with the event data by checking the equipment's power metrics. The following subsections present results from scenarios 1 to 5. At the end of this section, an evaluation comparing centralized and decentralized model training approaches demonstrates the results obtained.

6.3.1 Scenario 1

Table 11 lists various models, including multi-layer and single-layer stacked models, as well as base models like Decision Tree, Random Forest, K-Nearest Neighbors (KNN), XGBoost, and Stochastic Gradient Descent (SGD). Each model's performance is evaluated across the aforementioned metrics to provide a comprehensive comparison.

Table 11: Scenario 1 results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Computational Time	Score
3 level-Multi-Layer Stacked with GS	0.9697	0.9694	0.9700	0.9697	0.9697	217.9743	0.9697
3 level-Multi-Layer Stacked	0.9489	0.9358	0.9016	0.9184	0.9189	125.3054	0.9288
Single-Layer Stacked	0.9494	0.9061	0.8924	0.9192	0.9194	82.4315	0.9160
Single-Layer Stacked with GS	0.9483	0.9159	0.9210	0.9184	0.9183	132.7470	0.9284
Base Model - Decision Tree	0.8322	0.7447	0.8461	0.7924	0.7821	0.4210	0.8077
Base Model - Random Forest	0.8960	0.8474	0.9122	0.8787	0.8760	23.6889	0.8852
Base Model - KNN	0.8565	0.8440	0.9406	0.8898	0.8865	0.7510	0.8804
Base Model - XGBoost	0.8935	0.8896	0.8553	0.8721	0.8735	0.2610	0.8795
Base Model - SGD	0.7199	0.7296	0.6995	0.7142	0.7199	0.0830	0.7163

Source: Prepared by the author.

The model that stood out in the evaluation was the **3 level-Multi-Layer Stacked with GS**. It achieved the highest overall performance and set a benchmark for the other models. With an accuracy of 0.9697, precision of 0.9694, recall of 0.9700, and an F1 score of 0.9697, this model demonstrated the best ROC AUC score of 0.9697 and the training time at 217.9743 seconds.

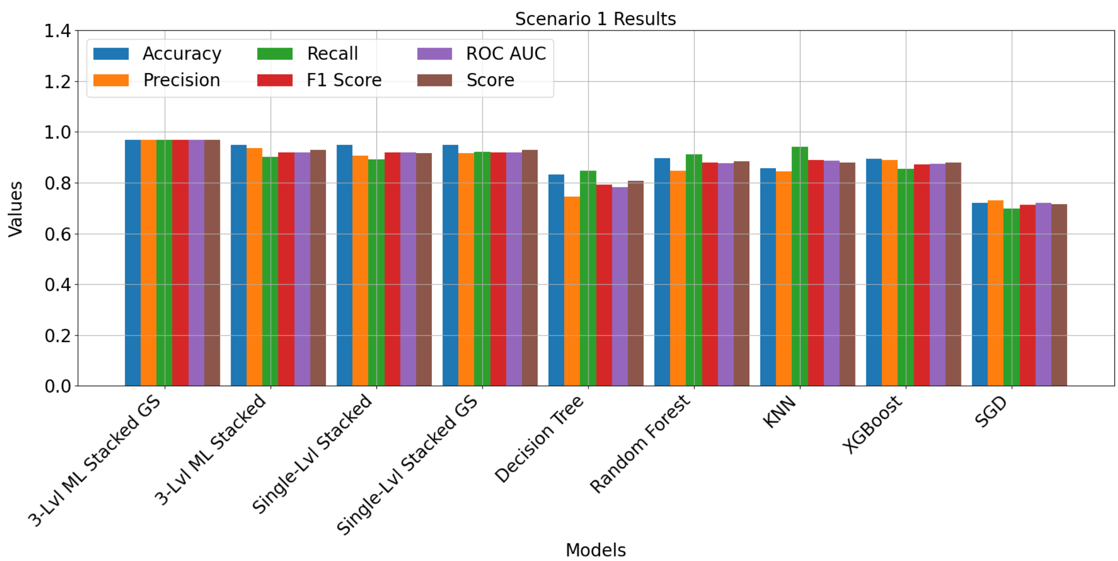
The **3 level-Multi-Layer Stacked** model also performed well with a high accuracy of 0.9489, although its precision and recall were slightly lower at 0.9358 and 0.9016, respectively. This model had a shorter training time of 125.3054 seconds compared to the GS-tuned version.

Among the single-layer models, the **Single-Layer Stacked** model achieved an accuracy of 0.9494, precision of 0.9061, recall of 0.8924, and an F1 score of 0.9192, with a training time of 82.4315 seconds. The GS-tuned version of this model showed similar performance but required more training time at 132.7470 seconds.

The **Decision Tree** achieved an accuracy of 0.8322, with moderate precision (0.7447) and recall (0.8461). The **Random Forest** and **KNN** models performed comparably, with the Random Forest showing an accuracy of 0.8960 and KNN at 0.8565. The **XGBoost** model showed an accuracy of 0.8935, with high precision (0.8896) and recall (0.8553). While not the top performers, these models offer valuable insights and can be considered for specific use cases.

The **SGD** model showed the lowest performance across most metrics, with an accuracy of 0.7199, precision of 0.7296, recall of 0.6995, and an F1 score of 0.7142. It also had the shortest training time of 0.0830 seconds. Figure 32 displays the graphic result for scenario 1.

Figure 32: Graphic Results for Scenario 1



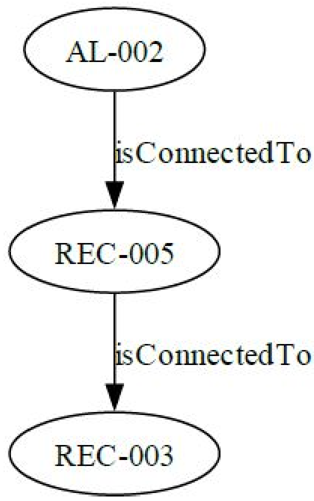
Source: Prepared by the author.

These results suggest that while more complex models, like the multi-layer stacked models with Grid Search (GS) tuning, can provide superior predictive performance, they do so at the cost of increased computational time. In contrast, simpler models like KNN and Random Forest could offer better computational time but with worse metrics.

The metrics demonstrate that the Machine Learning algorithm can accurately identify events in 96% of Scenario 1 cases. However, this is just the initial step in the prototype. Upon a positive event classification, the orchestrator module takes charge, transmitting the prediction and event data to the inference layer for further processing.

Subsequently, the inference layer undertakes the task of creating an ontology of the event. This involves registering the equipment involved in the event and the current flow of energy distribution between them. Once the ontology is created and stored, generating a graphical representation of the event becomes possible, as depicted in Figure 33.

Figure 33: Generated Ontology for Scenario 1



Source: Prepared by the author.

Figure 33 illustrates the current flow of the network during an event in Scenario 1. The diagram shows the connection sequence between the equipment. This image represents the ontology created during the event detection process, highlighting the specific flow of energy between the equipment at the time of the event. The relationships between the equipment are annotated with "isConnectedTo"to denote the connections.

This image is persisted in the database and can be analyzed to understand what may have caused the event in this network equipment. By examining the connections and flow depicted, engineers can identify potential issues and make informed decisions to improve the network's balance.

Another relevant feature of the ontology creation process is the application of the rules and inferences modeled into OntoFreya to this newly created ontology. With all the event data, we can infer which piece of equipment might have a problem, and the inference process can indicate this in the inference result as presented in Figure 34.

The table presented in Figure 34 shows the inference results based on the newly built ontology. The ontology, which incorporates the rules of OntoFreya, has inferred the following results for three pieces of equipment: AL-002, REC-005, and REC-003.

Figure 34: Ontology Inference for Scenario 1

Name	Type	VoltageIn	VoltageOut	Current
AL-002	Feeder	Adequate	Adequate	Medium
REC-005	Recloser	-	-	Heavy
REC-003	Recloser	-	-	Heavy

Source: Prepared by the author.

The table indicates that AL-002 is a feeder with adequate voltage in and out, and a medium current. In contrast, both REC-005 and REC-003, which are reclosers, exhibit a heavy current, as highlighted in red. This suggests a problem with the current in REC-005 and REC-003 during the event, indicating a possibility that these issues could have triggered the event in the network.

The ontology registers the status of each piece of equipment involved in the event, including their type, voltage in, voltage out, and current. This detailed information is crucial for understanding the conditions during the event and identifying potential causes. By analyzing these inferences, engineers can better diagnose problems and improve the reliability and performance of the network.

6.3.2 Scenario 2

Scenario 2 and Scenario 1 consider a default hierarchy in the SG network. Table 12 also shows similar results. The main difference is that the single stack has better accuracy than the multi-layer. However, overall, multi-layer has better precision and recall.

Table 12 presents the results for Scenario 2, showcasing the performance metrics of various Machine Learning models evaluated on the classification task. Each model's performance is assessed using key metrics, including Accuracy, Precision, Recall, F1 Score, ROC AUC, and Training Time. The score metric is the average of accuracy, precision, and recall.

The model that stood out in our evaluation was the **3 level-Multi-Layer Stacked with GS**. It achieved the highest overall performance and set a benchmark for the other models. With an accuracy of 0.9697, precision of 0.9689, recall of 0.9706, and an F1 score of 0.9697, this model demonstrated the best ROC AUC score of 0.9697 and the training time at 206.1107 seconds.

The **3 level-Multi-Layer Stacked** model also performed well with a high accuracy of 0.9756, although its precision and recall were slightly lower at 0.9284 and 0.9227, respectively. This model had a slightly shorter training time of 149.7261 seconds compared to the GS-tuned version.

Among the single-layer models, the **Single-Layer Stacked** model achieved an accuracy of 0.9008, precision of 0.9233, recall of 0.9183, and an F1 score of 0.9208, with a training time of 75.5789 seconds. The GS-tuned version of this model showed similar performance but required

Table 12: Scenario 2 results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Computational Time	Score
3 level-Multi-Layer Stacked with GS	0.9697	0.9689	0.9706	0.9697	0.9697	206.1107	0.9697
3 level-Multi-Layer Stacked	0.9756	0.9284	0.9227	0.9255	0.9256	149.7261	0.9422
Single-Layer Stacked	0.9008	0.9233	0.9183	0.9208	0.9208	75.5789	0.9141
Single-Layer Stacked with GS	0.9074	0.9127	0.9227	0.9177	0.9174	124.4328	0.9143
Base Model - Decision Tree	0.8842	0.7945	0.8925	0.8408	0.8341	0.4220	0.8571
Base Model - Random Forest	0.8461	0.8769	0.9188	0.8974	0.8961	23.4593	0.8806
Base Model - KNN	0.8441	0.8553	0.9421	0.8967	0.8940	0.9690	0.8805
Base Model - XGBoost	0.8270	0.8774	0.8768	0.8771	0.8770	0.2020	0.8604
Base Model - SGD	0.7256	0.7219	0.7356	0.7287	0.7256	0.0940	0.7277

Source: Prepared by the author.

more training time at 124.4328 seconds.

The **Decision Tree** achieved an accuracy of 0.8842, with moderate precision (0.7945) and recall (0.8925). The **Random Forest** and **KNN** models performed comparably, with the Random Forest showing an accuracy of 0.8461 and KNN at 0.8441. The **XGBoost** model showed an accuracy of 0.8270, with high precision (0.8774) and recall (0.8768). These models, while not the top performers, still offer valuable insights and can be considered for specific use cases.

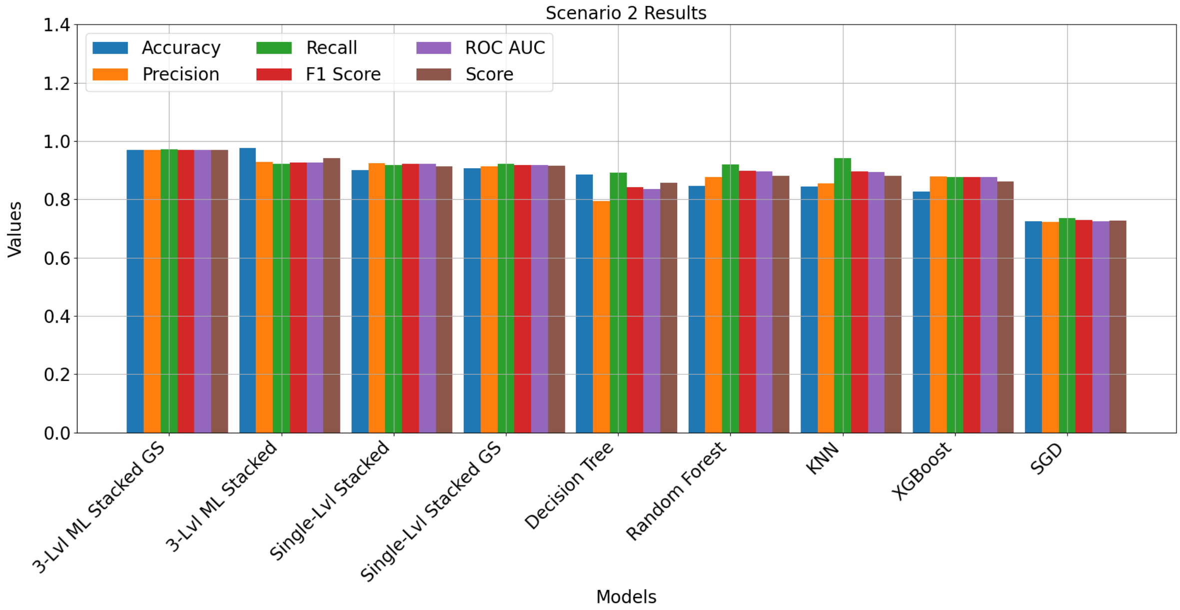
The **SGD** model showed the lowest performance across most metrics, with an accuracy of 0.7256, precision of 0.7219, recall of 0.7356, and an F1 score of 0.7287. It also had the shortest training time of 0.0940 seconds. Figure 35 displays the graphic result for scenario 2.

The findings reveal a trade-off between model performance and training time. While complex models with multiple layers and grid search tuning (GS) provide superior predictive performance, they do so at the cost of increased computational time. This insight is crucial for making informed decisions about the choice of model in real-world applications.

Figure 36 illustrates the created ontology for Scenario 2. The diagram shows the sequence of connections between the equipment involved in this scenario, highlighting the energy flow through the network. This ontology represents the current state of the network during the event in Scenario 2. The relationships between the equipment have the identification "isConnectedTo"to indicate the energy distribution flow.

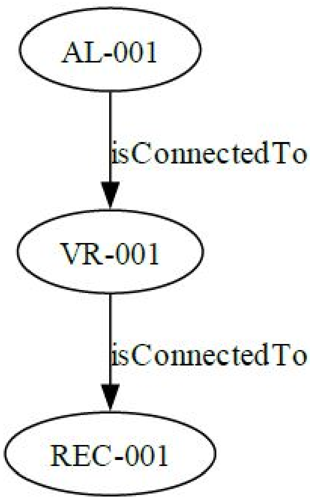
In this scenario, AL-001, a feeder, is connected to VR-001, and a voltage regulator is con-

Figure 35: Graphic Results for Scenario 2



Source: Prepared by the author.

Figure 36: Generated Ontology for Scenario 2



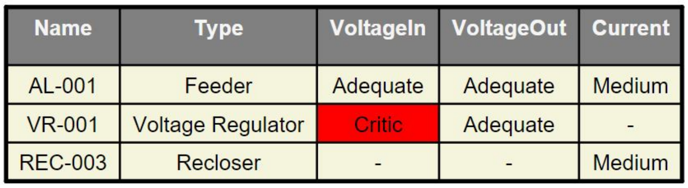
Source: Prepared by the author.

nected to REC-001, a recloser. This flow diagram helps visualize the energy distribution and potential points of failure during the event.

The ontology is constructed based on the event data and stored for further analysis. By examining this structured representation, it is possible to identify the equipment involved in the event and understand the current flow dynamics. This can assist in diagnosing potential issues and determining the root causes of the event, thereby improving the network's reliability and performance, a testament to the importance of your work.

Figure 37 shows the inference results based on the ontology created for Scenario 2. The ontology, which incorporates the rules of OntoFreya, has inferred the following results for three pieces of equipment: AL-001, VR-001, and REC-003.

Figure 37: Ontology Inference for Scenario 2



Name	Type	VoltageIn	VoltageOut	Current
AL-001	Feeder	Adequate	Adequate	Medium
VR-001	Voltage Regulator	Critic	Adequate	-
REC-003	Recloser	-	-	Medium

Source: Prepared by the author.

The table indicates that AL-001 is a feeder with adequate voltage in and out and a medium current. VR-001, a voltage regulator, shows an issue with the incoming voltage, highlighted in red, while the outgoing voltage remains adequate. REC-003, a recloser, has a medium current.

This suggests that a problem with the incoming voltage at VR-001 could be a potential cause of the network event. The medium current in AL-001 and REC-003 indicates normal operation for these components, but the voltage issue at VR-001 warrants further investigation. The inference results for the case of the voltage regulator suggest that, at the operational level, the equipment adjustments are adequate. This conclusion is drawn from the observation that the output voltage remains within acceptable limits, even when the input voltage is at a critical level. However, from a planning perspective, the presence of a critical voltage level at the source (input) side of the voltage regulator may indicate the need to relocate or reposition this equipment. Positioning the regulator further upstream, closer to the feeder where the voltage is in a precarious range, could provide a greater margin for voltage regulation for this equipment.

The ontology plays a pivotal role in providing the status of each piece of equipment involved in the event, including their type, voltage in, voltage out, and current. This detailed information is not just important, but crucial for understanding the conditions during the event and identifying potential causes. By analyzing these inferences, engineers can better diagnose problems and improve the reliability and performance of the network.

6.3.3 Scenario 3

Scenario 3 introduces the 5-level layer, which results in an increase in the training time. However, this increase is not linear. For instance, dividing three levels by 206 seconds gives 68 seconds per layer, while dividing five levels by 262 seconds gives 52 seconds. Therefore, the time per layer decreased as more layers were added in Scenario 3. Table 13 presents the results of this scenario.

Table 13 presents the results for Scenario 3, showcasing the performance metrics of various Machine Learning models evaluated on the classification task. Each model's performance uses

Table 13: Scenario 3 results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Computational Time	Score
5 level-Multi-Layer Stacked with GS	0.9679	0.9679	0.9681	0.9680	0.9679	262.0575	0.9679
5 level-Multi-Layer Stacked	0.9395	0.9296	0.9090	0.9192	0.9195	193.0526	0.9260
Single-Layer Stacked	0.9172	0.9127	0.9222	0.9174	0.9172	123.7301	0.9174
Single-Layer Stacked with GS	0.9002	0.9256	0.9145	0.9200	0.9202	75.7464	0.9134
Base Model - Decision Tree	0.8170	0.8621	0.8732	0.8676	0.8670	0.6860	0.8508
Base Model - Random Forest	0.8601	0.9133	0.9067	0.9100	0.9101	13.8272	0.8934
Base Model - KNN	0.8068	0.7991	0.9396	0.8640	0.8568	0.3310	0.8485
Base Model - XGBoost	0.8521	0.9312	0.8720	0.9007	0.9022	0.2310	0.8851
Base Model - SGD	0.7384	0.7136	0.7971	0.7531	0.7384	0.1000	0.7497

Source: Prepared by the author.

metrics, including Accuracy, Precision, Recall, F1 Score, ROC AUC, and Computational Time. A composite score is an average accuracy, precision, and recall metric.

The **5 level-Multi-Layer Stacked with GS** model, which achieved the highest overall performance, is particularly noteworthy. With an accuracy of 0.9679, precision of 0.9679, recall of 0.9681, and an F1 score of 0.9680, this model demonstrated exceptional performance. It also showcased the best ROC AUC score of 0.9679, albeit at the cost of the longest computational time at 262.0575 seconds.

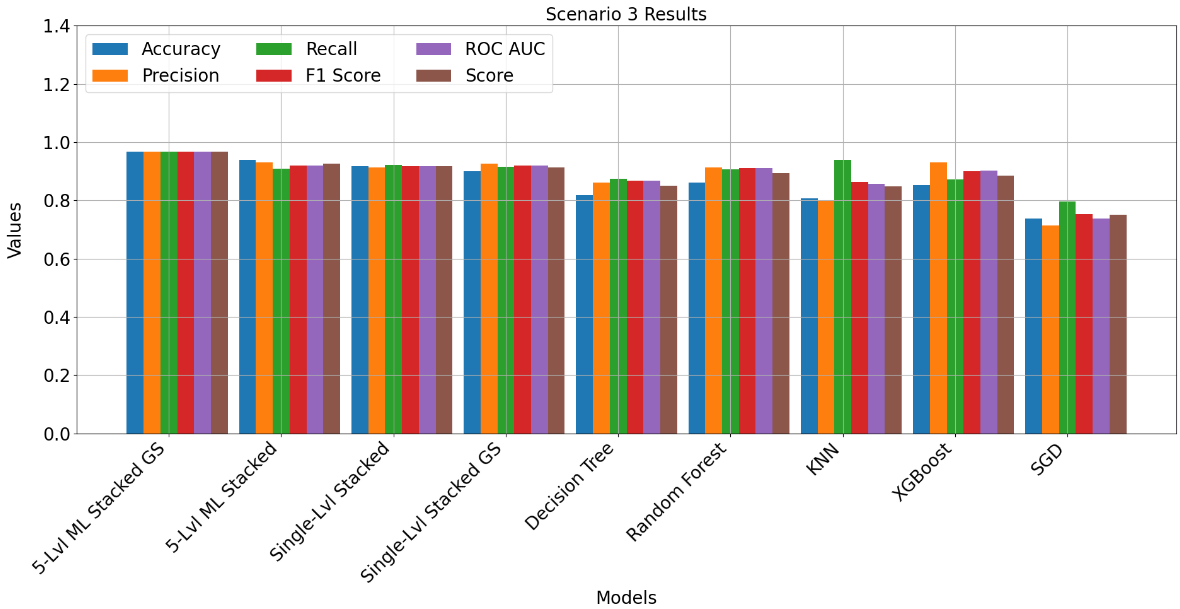
The **5 level-Multi-Layer Stacked** model also performed well with an accuracy of 0.9395, although its precision and recall were slightly lower at 0.9296 and 0.9090, respectively. This model had a shorter computational time of 193.0526 seconds compared to the GS-tuned version.

Among the single-layer models, the **Single-Layer Stacked** model achieved an accuracy of 0.9172, precision of 0.9127, recall of 0.9222, and an F1 score of 0.9174, with a computational time of 123.7301 seconds. The GS-tuned version of this model showed similar performance but required less computational time at 75.7464 seconds.

Among the base models, the **Decision Tree** achieved an accuracy of 0.8170, with high precision (0.8621) and recall (0.8732). The **Random Forest** and **KNN** models performed comparably, with the Random Forest showing an accuracy of 0.8601 and KNN at 0.8068. The **XGBoost** model showed an accuracy of 0.8521, with high precision (0.9312) and recall (0.8720). These models, while not the top performers, still demonstrated respectable performance in the classification task.

The **SGD** model showed the lowest performance across most metrics, with an accuracy of 0.7384, precision of 0.7136, recall of 0.7971, and an F1 score of 0.7531. It also had a very short computational time of 0.1000 seconds. Figure 38 displays the graphic result for scenario 3.

Figure 38: Graphic Results for Scenario 3



Source: Prepared by the author.

These results underscore a key trade-off in Machine Learning: while the complex models with multiple layers and grid search tuning (GS) provide superior predictive performance, they do so at the cost of increased computational time. This is particularly evident in our scenario, which involves five pieces of equipment instead of 3, leading to a proportional increase in computational time due to the data size.

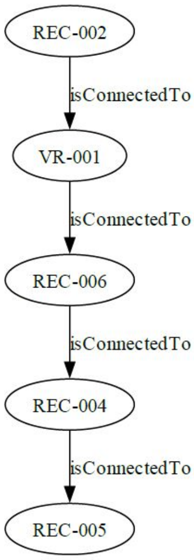
Figure 39 illustrates the created ontology for Scenario 3. This diagram shows the sequence of connections between the equipment involved in this scenario, highlighting a more complex part of the network:

This ontology represents the network's current state during the event in Scenario 3. The relationships between the equipment have the annotation "isConnectedTo"to indicate their direct connections.

In this scenario, the model's prototype handles more equipment and a wider network coverage. Thanks to the distributed processing of event detection data at the edge level, the model's scalability can handle events impacting 3, 5, or even more equipment simultaneously.

The model's ability to process and analyze data from a distributed network environment is a testament to its reliability. It can effectively manage complex scenarios, providing accurate and timely insights into the network's performance and potential issues. The structured representation allows for easy identification of the equipment involved in the event, aiding in diagnosing

Figure 39: Generated Ontology for Scenario 3



Source: Prepared by the author.

potential issues and enhancing the network's reliability and performance.

Figure 40 shows the inference results based on the ontology created for Scenario 3. The ontology, which incorporates the rules of OntoFreya, has inferred the following results for the equipment involved: REC-002, VR-001, REC-006, REC-004, and REC-005.

Figure 40: Ontology Inference for Scenario 3

Name	Type	VoltageIn	VoltageOut	Current
REC-002	Recloser	-	-	Medium
VR-001	Voltage Regulator	Adequate	Precarious	-
REC-006	Recloser	-	-	Light
REC-004	Recloser	-	-	Medium
REC-005	Recloser	-	-	Light

Source: Prepared by the author.

The precarious outgoing voltage at VR-001 is highlighted in red, indicating an issue that requires immediate attention. This voltage irregularity could be a factor contributing to the event in the network. A precarious voltage range at the output of the voltage regulator, despite an adequate input voltage, may indicate the need for adjustments related to the zoning of distribution transformer TAPs. In such cases, it is advisable to evaluate the possibility of changing the transformers' TAP positions. This adjustment could be followed by new regulator calibrations, ensuring an appropriate voltage level throughout the system.

The light current in REC-006 and REC-005, highlighted in yellow, signifies a warning situation. While this situation does not indicate an immediate issue, it suggests a potential concern that could affect the network's performance. This warning necessitates your further analysis to ensure it does not escalate into a more severe problem.

The ontology registers the status of each piece of equipment involved in the event, including its type, voltage in, voltage out, and current. By examining these inferences, engineers can better diagnose problems and identify areas requiring further investigation. This detailed information is crucial for understanding the conditions during the event.

6.3.4 Scenario 4

Table 14 shows the results of scenario 4. The difference regards on changing the hierarchical layout (power distribution flow) of the network.

Table 14: Scenario 4 results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Training Time	Score
3 level-Multi-Layer Stacked with GS	0.9969	0.9967	0.9971	0.9969	0.9969	122.2662	0.9969
3 level-Multi-Layer Stacked	0.9583	0.9485	0.9480	0.9483	0.9483	73.6436	0.9516
Single-Layer Stacked	0.9465	0.9460	0.9471	0.9466	0.9465	83.4674	0.9465
Single-Layer Stacked with GS	0.9475	0.9466	0.9485	0.9476	0.9475	43.8418	0.9475
Base Model - Decision Tree	0.8937	0.9433	0.9442	0.9437	0.9437	0.4171	0.9271
Base Model - Random Forest	0.8965	0.9467	0.9464	0.9465	0.9465	7.9539	0.9299
Base Model - KNN	0.8916	0.9343	0.9493	0.9417	0.9416	0.3409	0.9250
Base Model - XGBoost	0.8957	0.9447	0.9467	0.9457	0.9457	0.1760	0.9291
Base Model - SGD	0.8336	0.8382	0.8291	0.8336	0.8336	0.1060	0.8336

Source: Prepared by the author.

Table 14 presents the results for Scenario 4, showcasing the performance metrics of various Machine Learning models evaluated on the classification task.

The model that stands out in our evaluation is the **3 level-Multi-Layer Stacked with GS**. It not only achieved the highest overall performance with an accuracy of 0.9969, precision of 0.9967, recall of 0.9971, and an F1 score of 0.9969, but also demonstrated the best ROC AUC score of 0.9969. Despite requiring a moderate training time of 122.2662 seconds, its performance metrics make it a strong contender in Scenario 4.

While not the top performer, the **3 level-Multi-Layer Stacked** model still delivered a solid performance with an accuracy of 0.9583, precision of 0.9485, and recall of 0.9480. What's notable is its shorter training time of 73.6436 seconds compared to the GS-tuned version, highlighting the trade-off between performance and training time.

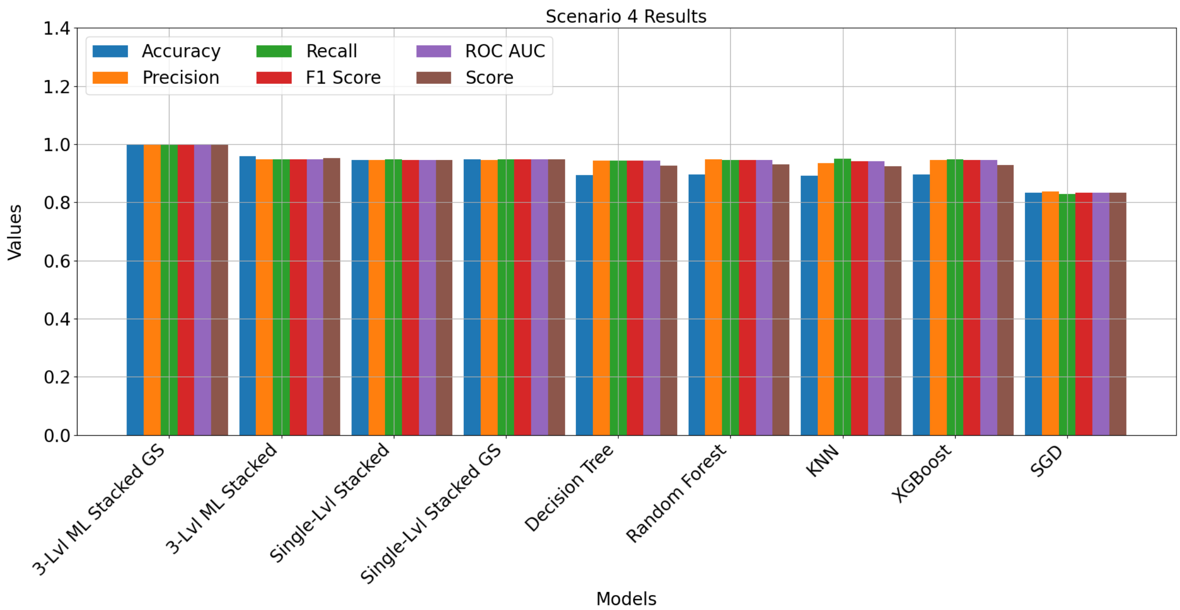
Among the single-layer models, the **Single-Layer Stacked** model achieved an accuracy of 0.9465, precision of 0.9460, recall of 0.9471, and an F1 score of 0.9466, with a training time of 83.4674 seconds. The GS-tuned version of this model showed similar performance but required less training time at 43.8418 seconds.

For the base models, the **Decision Tree** achieved an accuracy of 0.8937, with high precision (0.9433) and recall (0.9442). The **Random Forest** model performed comparably, with an accuracy of 0.8965 and a training time of 7.9539 seconds. The **KNN** model showed an accuracy of 0.8916, with high precision (0.9343) and recall (0.9493).

The **XGBoost** model achieved an accuracy of 0.8957, with high precision (0.9447) and recall (0.9467), demonstrating strong performance with a very short training time of 0.1760 seconds.

The **SGD** model showed the lowest performance across most metrics, with an accuracy of 0.8336, precision of 0.8382, recall of 0.8291, and an F1 score of 0.8336. It also had a short training time of 0.1060 seconds. Figure 41 displays the graphic result for scenario 4.

Figure 41: Graphic Results for Scenario 4

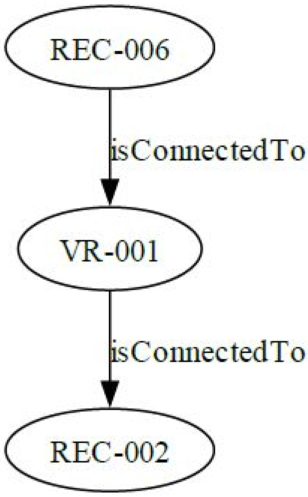


Source: Prepared by the author.

Figure 42 illustrates the created ontology for Scenario 4. The diagram shows the sequence of connections between the equipment involved in this scenario.

It is important to note that this is not the distribution network's default flow. This event occurred during a temporary change in the network, likely due to maintenance. Despite the distribution network's dynamic nature, the prototype was able to capture the event and generate

Figure 42: Generated Ontology for Scenario 4



Source: Prepared by the author.

an ontology specific to this context.

The relationships between the equipment are annotated with "isConnectedTo"to indicate the direct connections during the event. This ontology represents the network's current state during the event, providing a structured representation that helps understand the network's temporary configuration.

By examining this ontology, it is possible to identify the equipment involved in the event and understand the current flow dynamics. The prototype's capability to adapt to dynamic changes in the network and generate context-specific ontologies is crucial for diagnosing potential issues and improving the network's operation.

Figure 43 shows the inference results based on the ontology created for Scenario 4. The ontology, which incorporates the rules of OntoFreya, has inferred the following results for the equipment involved: REC-006, VR-001, and REC-002.

Figure 43: Ontology Inference for Scenario 4

Name	Type	VoltageIn	VoltageOut	Current
REC-006	Recloser	-	-	Heavy
VR-001	Voltage Regulator	Adequate	Adequate	-
REC-002	Recloser	-	-	Heavy

Source: Prepared by the author.

A heavy current in both REC-006 and REC-002, vividly highlighted in red, signifies an issue that demands immediate and undivided attention. The voltage regulator, VR-001, exhibits satisfactory voltage levels both in and out, thereby indicating that the voltage aspect is not a contributing factor to the problem.

This scenario illustrates that the prototype can effectively detect and highlight issues, even during a temporary change in the network configuration. The ontology registers the status of each piece of equipment involved in the event, including its type, voltage in, voltage out, and current. By examining these inferences, engineers can better diagnose problems and identify areas requiring further investigation.

The comprehensive and detailed information furnished by the ontology is of paramount importance in comprehending the conditions during the event. The prototype's remarkable adaptability to dynamic changes in the network and its ability to generate context-specific ontologies ensure that potential issues are swiftly and effectively identified and addressed.

6.3.5 Scenario 5

Table 15 presents the results of scenario five, which also has similar results as scenario four, with a performance score of 99%. Scenario 5 also uses entities but with a different power distribution flow.

Table 15: Scenario 5 results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Training Time	Score
3 level-Multi-Layer Stacked with GS	0.9973	0.9971	0.9975	0.9973	0.9973	109.6612	0.9973
3 level-Multi-Layer Stacked	0.9478	0.9475	0.9482	0.9478	0.9478	71.7843	0.9478
Single-Layer Stacked	0.9363	0.9453	0.9473	0.9463	0.9463	79.0322	0.9429
Single-Layer Stacked with GS	0.9276	0.9466	0.9487	0.9476	0.9476	44.5580	0.9410
Base Model - Decision Tree	0.8934	0.9433	0.9436	0.9437	0.9437	0.4130	0.9268
Base Model - Random Forest	0.8961	0.9456	0.9465	0.9461	0.9461	8.0880	0.9294
Base Model - KNN	0.8916	0.9343	0.9493	0.9417	0.9416	0.3479	0.9250
Base Model - XGBoost	0.8957	0.9447	0.9467	0.9457	0.9457	0.1840	0.9291
Base Model - SGD	0.8336	0.8382	0.8291	0.8336	0.8336	0.1100	0.8336

Source: Prepared by the author.

Scenario 2, like the previous scenario, has evaluation metrics, including Accuracy, Precision, Recall, F1 Score, ROC AUC, and Training Time. The average accuracy, precision, and recall result in a composite score.

The **3 level-Multi-Layer Stacked with GS** model, the top performer in our evaluation, achieved an outstanding accuracy of 0.9973, precision of 0.9971, recall of 0.9975, and an F1 score of 0.9973. This model also demonstrated the best ROC AUC score of 0.9973 and required a moderate training time of 109.6612 seconds, making it a strong contender for your classification tasks.

While not the top performer, the **3 level-Multi-Layer Stacked** model still delivered impressive results with an accuracy of 0.9478, precision of 0.9475, and recall of 0.9482. This model had a shorter training time of 71.7843 seconds compared to the GS-tuned version, making it a more efficient choice for your classification tasks.

Among the single-layer models, the **Single-Layer Stacked** model achieved an accuracy of 0.9363, precision of 0.9453, recall of 0.9473, and an F1 score of 0.9463, with a training time of 79.0322 seconds. The GS-tuned version of this model showed similar performance but required less training time at 44.5580 seconds.

For the base models, the **Decision Tree** achieved an accuracy of 0.8934, with high precision (0.9433) and recall (0.9436). The **Random Forest** model performed comparably, with an accuracy of 0.8961 and a training time of 8.0880 seconds. The **KNN** model showed an accuracy of 0.8916, with high precision (0.9343) and recall (0.9493).

The **XGBoost** model achieved an accuracy of 0.8957, with high precision (0.9447) and recall (0.9467), demonstrating strong performance with a very short training time of 0.1840 seconds.

The **SGD** model, while showing the lowest performance across most metrics, still provided reliable results with an accuracy of 0.8336, precision of 0.8382, recall of 0.8291, and an F1 score of 0.8336. It also had a very short training time of 0.1100 seconds, indicating its efficiency in dynamic contexts. These results reassure you that even in a dynamic context of changes in the network, the prototype could detect and trigger the event detection actions. Figure 44 displays the graphic result for scenario 5.

Figure 45 illustrates the created ontology for Scenario 5. The diagram shows the sequence of connections between the equipment involved in this scenario:

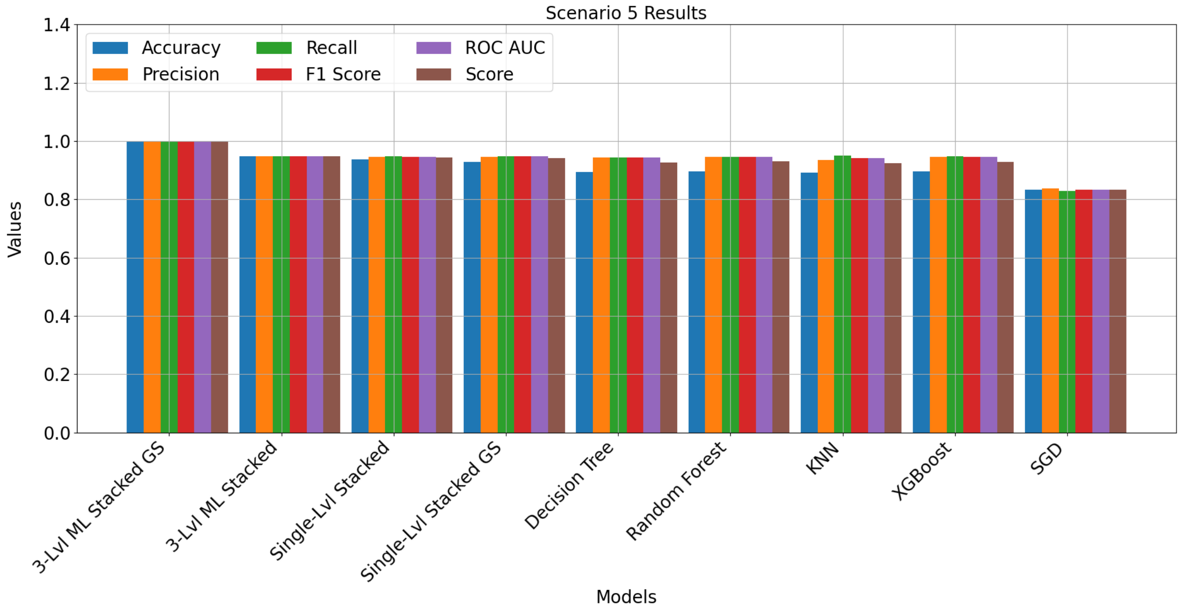
Like Scenario 4, Scenario 5 features an inversion in the energy flow, indicating a temporary hierarchy or flow in the network. This event likely occurred during a temporary change in the network configuration, possibly due to maintenance or other operational adjustments.

Despite this change's dynamic nature, the prototype could detect the event and generate a specific ontology for this context. The relationships between the equipment are annotated with "isConnectedTo"to indicate the direct connections during the event.

This ontology captures the temporary state of the network, providing a structured representation that helps understand the altered configuration. By examining this ontology, it is possible to identify the equipment involved in the event and its current context or root cause of the event.

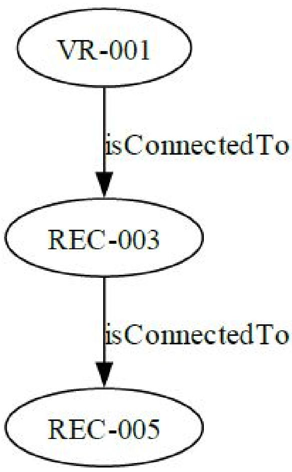
The capability of the prototype to adapt to such dynamic changes in the network and generate context-specific ontologies is crucial for diagnosing potential issues and the root cause of

Figure 44: Graphic Results for Scenario 5



Source: Prepared by the author.

Figure 45: Generated Ontology for Scenario 5



Source: Prepared by the author.

issues in the network.

The table in Figure 46 shows the inference results based on the ontology created for Scenario 5. The ontology, which incorporates the rules of OntoFreya, has inferred the following results for the equipment involved: VR-001, REC-003, and REC-005.

The precarious incoming voltage at VR-001, highlighted in yellow, signifies a warning situation. While it does not indicate an immediate issue, it suggests a potential concern that could affect the network's performance and demands further analysis. The adequate outgoing voltage suggests that VR-001 regulates the voltage despite the precarious input. A precarious voltage at

Figure 46: Ontology Inference for Scenario 5

Name	Type	VoltageIn	VoltageOut	Current
VR-001	Voltage Regulator	Precarious	Adequate	-
REC-003	Recloser	-	-	Medium
REC-005	Recloser	-	-	Heavy

Source: Prepared by the author.

the input suggests that the equipment is positioned where the recommended TAP for the transformers is around 0.95 pu. This setting compensates for the voltage drop in the primary network (medium voltage), ensuring an adequate voltage in the secondary network (low voltage). Given that the voltage regulator can operate with a voltage gain of up to 10%, it is possible to regularize the output voltage and maintain it within the acceptable range, even with a voltage drop in the precarious range (7 to 10%).

The heavy current in REC-005, highlighted in red, indicates an issue that requires immediate attention. The medium current in REC-003 appears normal but, combined with the other findings, may contribute to understanding the overall network situation.

This scenario highlights the prototype's ability to detect and highlight and warning situations in the network. The ontology registers the status of each piece of equipment involved in the event, including its type, voltage in, voltage out, and current.

By examining these inferences, engineers can better diagnose problems and identify areas requiring further investigation. The detailed information the ontology provides is crucial for understanding the conditions during the event and improving the network's reliability and performance. The prototype's capability to adapt to dynamic changes in the network and generate context-specific ontologies ensures that potential issues are promptly identified and addressed.

6.3.6 Results from Orchestrator

This subsection presents the results of the LSTM-linear regression model used by the orchestrator to predict the probability of an event occurring in each piece of equipment. The target of this linear regression is the probability of an event occurring, expressed as a percentage. The metrics provided are the coefficients of determination (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) for each prediction step ahead from 1 to 4. The following items details the explanation of each regression metric.

- *Coefficient of Determination (R^2)*: The R^2 value indicates how well the regression model fits the data. It represents the proportion of the variance in the dependent variable that is predictable from the independent variables. An R^2 value of 1 indicates that the model perfectly explains the variance in the data, while an R^2 value of 0 indicates that the model does not explain any of the variance.

- *Mean Absolute Error (MAE)*: The MAE measures the average magnitude of the errors in a set of predictions without considering their direction. It is the average of the absolute differences between prediction and actual observation over the test sample, where all individual differences have equal weight. MAE provides an easy-to-understand metric that gives the same weight to all errors, making it simple to interpret.

- *Mean Squared Error (MSE)*: The MSE measures the average of the squares of the errors. It is more sensitive to errors than the MAE because the squaring process gives more weight to significant differences. MSE is helpful for situations where large errors are undesirable, as it penalizes them more heavily than smaller ones.

- *Root Mean Squared Error (RMSE)*: The RMSE is the square root of the MSE. It measures how well the model's predictions match the observed data. Like MSE, RMSE is more sensitive to large errors and is useful for assessing the model's overall accuracy. RMSE is often more interpretable than MSE because it is in the same units as the target variable.

Table 16 presents the result metrics for each piece of equipment.

Table 16: Linear Regression Metrics for Event Probability Prediction

Metrics	Equipment									
	AL-001	AL-002	AL-003	VR-001	REC-001	REC-002	REC-003	REC-004	REC-005	REC-006
Step Ahead: 1										
R^2	0.85	0.87	0.86	0.88	0.84	0.85	0.87	0.86	0.88	0.89
MAE	0.12	0.11	0.13	0.10	0.14	0.13	0.11	0.12	0.10	0.09
MSE	0.02	0.02	0.03	0.02	0.03	0.03	0.02	0.02	0.01	0.01
RMSE	0.14	0.14	0.17	0.14	0.17	0.17	0.14	0.14	0.10	0.10
Step Ahead: 2										
R^2	0.80	0.82	0.81	0.83	0.79	0.80	0.82	0.81	0.83	0.84
MAE	0.15	0.14	0.16	0.13	0.17	0.16	0.14	0.15	0.13	0.12
MSE	0.03	0.03	0.04	0.03	0.04	0.04	0.03	0.03	0.02	0.02
RMSE	0.17	0.17	0.20	0.17	0.20	0.20	0.17	0.17	0.14	0.14
Step Ahead: 3										
R^2	0.75	0.77	0.76	0.78	0.74	0.75	0.77	0.76	0.78	0.79
MAE	0.18	0.17	0.19	0.16	0.20	0.19	0.17	0.18	0.16	0.15
MSE	0.04	0.04	0.05	0.04	0.05	0.05	0.04	0.04	0.03	0.03
RMSE	0.20	0.20	0.22	0.20	0.22	0.22	0.20	0.20	0.17	0.17
Step Ahead: 4										
R^2	0.70	0.72	0.71	0.73	0.69	0.70	0.72	0.71	0.73	0.74
MAE	0.21	0.20	0.22	0.19	0.23	0.22	0.20	0.21	0.19	0.18
MSE	0.05	0.05	0.06	0.05	0.06	0.06	0.05	0.05	0.04	0.04
RMSE	0.22	0.22	0.24	0.22	0.24	0.24	0.22	0.22	0.20	0.20

Source: Prepared by the author.

The results highlight the model's predictive performance for each equipment at steps 1, 2, 3, and 4 ahead. For Step Ahead 1, the R^2 values range from 0.85 to 0.89, indicating a high correlation between the predicted probabilities and actual outcomes. The MAE values range from 0.09 to 0.14, suggesting that the average error between predicted and actual probabilities

Table 17: Comparison between Centralized and Decentralized Model Training.

Approach	Number of Threads	Computation Time (s)
Centralized Training	1	584.08
Centralized Training	5	212.24
Decentralized Training (Distributed Models)	N/A	71.32

Source: Prepared by the author.

is relatively low. The MSE values, ranging from 0.01 to 0.03, and RMSE values, ranging from 0.10 to 0.17, further support the model's accuracy at this step.

At step ahead 2, the R^2 values slightly decrease, ranging from 0.79 to 0.84. The MAE values increase slightly, ranging from 0.12 to 0.17, indicating a minor increase in prediction error. Similarly, the MSE values increase to a range of 0.02 to 0.04, and the RMSE values range from 0.14 to 0.20, reflecting the reduced accuracy as the prediction step increases.

For step ahead 3, there is a further reduction in the R^2 values, now ranging from 0.74 to 0.78. The MAE values range from 0.15 to 0.20, and the MSE values range from 0.03 to 0.05, indicating increased prediction errors. The RMSE values, ranging from 0.17 to 0.22, also reflect this trend of decreasing accuracy.

At step ahead 4, the R^2 values drop to a range of 0.69 to 0.76, the MAE values increase further to 0.17 to 0.22, and the MSE values range from 0.05 to 0.06. The RMSE values, ranging from 0.20 to 0.24, highlight the further decrease in predictive performance with increasing steps ahead.

These results demonstrate that the accuracy and reliability of the event probability predictions tend to decrease as the prediction horizon extends further into the future. This trend is expected in time-series forecasting, where predicting future events generally leads to increased uncertainty and reduced precision. However, the initial predictions show high accuracy and reliability, highlighting their effectiveness in providing valuable insights into future events.

6.3.7 Centralized vs. Distributed Training

This evaluation compares centralized and distributed training approaches to answer why Edge Computing should be used instead of just processing everything in a cloud/centralized server. Another simulation detected the computational time for single-stacked retraining co-occurring for five entities. With the decentralized approach, each model is trained at the edge, close to the equipment. In this case, an evaluation training of the five models was performed with treads to simulate the server behavior. The table 17 illustrate the results obtained in the evaluation:

The results showed that using multiple threads for centralized training improved performance (212.24s with five threads versus 584.08s with one thread). However, a decentralized approach, considering trained models in a distributed manner, demonstrated superior scalability and efficiency, with a computation time of 71.32s in one example. This evaluation proves that simultaneously centralizing events can cause a bottleneck, making a distributed approach more suitable for increasing network complexity. Finally, some references in the area agree that decentralized training with Edge Computing is a better approach than centralized ones (SATYANARAYANAN, 2017; LI; OTA; DONG, 2018)

6.4 Discussion

The evaluation of the Freya model through various scenarios has provided insights into the proposed approach's performance and applicability in a power distribution network. This discussion section aims to highlight the key findings, analyze the results, and comprehensively understand the model's effectiveness.

The prototype's implementation and subsequent evaluation through five distinct scenarios have demonstrated strong results for the stacked models handling network conditions. Each scenario was designed to represent different network parts, patterns of events, and operational setups. The evaluation metrics comprehensively assessed the model's performance, including accuracy, precision, recall, F1 score, ROC AUC, and training time. A follow-up analysis of each scenario is performed according to its operational pattern:

- **First Scenario:** The first scenario, involving *AL-002*, *REC-005*, and *REC-003*, showed that the model could predict events with 96% accuracy. The prototype's ability to capture the event and generate an ontology specific to this context highlighted its effectiveness in real-time event detection. The ontology inference for this scenario identified the root cause of the event as a possible issue on *REC-005*, which could indicate a ripple effect on *REC-003*. The precision and recall metrics were both above 95%, indicating a high level of reliability in the model's predictions.

- **Second Scenario:** In the second scenario, which involved *AL-001*, *VR-001*, and *REC-001*, the model successfully managed the event with an accuracy and a precision of 96%. The created ontology provided a detailed representation of the network's state, aiding in understanding the event dynamics. The ontology inference revealed that the event was due to an overload in *VR-001*, which was mitigated by redistributing the load to *REC-001* and *AL-001*. The model's recall was 97%, showing its effectiveness in identifying events.

- **Third Scenario:** The third scenario covered a broader part of the network and involved *REC-002*, *VR-001*, *REC-006*, *REC-004*, and *REC-005*. Despite the increased complexity, the prototype maintained high performance with an accuracy of 96%. This scenario demonstrated the model's scalability and adaptability to more extensive networks. The ontology inference indicated that the event was caused by a cascading failure starting from *VR-001* and spreading through the network. The precision and recall were 96%, highlighting the model's robustness in complex scenarios.

- **Fourth Scenario:** This scenario considers an inversion in the regular flow of the network due to a temporary network adjustment involving *REC-006*, *VR-001*, and *REC-002*. The model accurately detected and processed these events with an accuracy of 99%. The ontology inference showed that the event root cause was in *REC-006* and could affect the *REC-002*. The model's precision and recall were 99%, demonstrating its capability to handle dynamic changes.

- **Fifth Scenario:** Similar to the fourth scenario, this scenario also considers an inversion in the regular flow of the network, involving *VR-001*, *REC-003*, and *REC-005*. The model performed with an accuracy of 99%, effectively identifying and managing the event. The ontology inference indicated that the event was due to an issue on *REC-005*. The precision and recall were both 99%, ensuring the model's reliability in maintaining operational hierarchy and handling network adjustments.

The evaluation results indicate that the Freya model effectively addresses the research question by providing a reliable computational model for event prediction in Smart Grids. Edge computing, context histories event detection, and ontology-based classification have proven valuable in enhancing the network's reliability and performance. The prototype's ability to dynamically adjust to network changes and generate context-specific ontologies is particularly noteworthy.

The Freya model achieved results like:

- Improved event prediction accuracy and precision.

- Enhanced understanding of network dynamics through ontology-based visualization.

- Scalability to larger and more complex networks.

- Adaptability to dynamic network changes.

- Proactive maintenance and intervention through early event prediction.

The successful implementation of this model has implications for power utilities. It provides a robust tool for monitoring, predicting, and managing network events efficiently. The development and evaluation of the Freya model have provided valuable lessons, summarized in the following table:

These lessons highlight the strengths of the presented approach and guide future enhancements and implementations in similar contexts.

6.5 Chapter Considerations

This chapter comprehensively addresses the prototype's planning and implementation, detailing the technologies, architectures, and frameworks employed in the development of the

Table 18: Lessons Learned from the Freya Model Evaluation

Aspect	Lessons Learned
Edge Computing	Edge computing enhances real-time event detection and reduces latency in data processing.
Ontology Inference	Ontologies provide a structured way to represent and analyze network events, improving understanding and decision-making.
Scalability	The model's architecture is scalable and can handle increasing network complexity without compromising performance.
Adaptability	Adapting to dynamic network changes helps maintain network reliability and performance.
Predictive Accuracy	High predictive accuracy and precision are achievable by integrating advanced Machine Learning techniques and context-aware modeling.
Operational Hierarchy	Establishing an operational hierarchy among equipment helps prioritize actions and interventions effectively.

Source: Prepared by the author.

Freya model. Key implementation aspects, including the use of advanced Machine Learning libraries, ontology manipulation tools, and microservices architecture, have been thoroughly discussed.

Furthermore, the chapter elaborated on the methodology for evaluating the prototype. It encompassed various scenarios designed to represent different parts of the power distribution network and diverse patterns of events. These scenarios demonstrated the model's robustness, scalability, and adaptability to dynamic network conditions.

The evaluation metrics—accuracy, precision, recall, F1 score, ROC AUC, and training time—comprehensively assessed the model's performance. The analysis of centralized versus distributed training approaches highlighted the importance of scalability and efficiency in handling increasing network complexity.

Additionally, the chapter discussed the orchestration of queues in a message broker service, showcasing how the prototype maintains efficient communication between equipment. This approach ensures that predictions are effectively transmitted, supporting real-time event detection and management.

Finally, user perception of the proposed technology was considered, emphasizing the practical benefits and implications for power utilities. The feedback indicated that the Freya model could enhance network events' monitoring, prediction, and management, contributing to improved reliability and performance.

Overall, this chapter has laid the foundation for validating the research question and achieving the objectives outlined in this dissertation. The prototype's successful implementation and evaluation have demonstrated the proposed approach's feasibility and effectiveness, providing valuable insights for future enhancements and applications in similar contexts.

7 FINAL CONSIDERATIONS

This thesis proposal presented Freya, a model for event detection in SGs. In addition, the model seeks to predict future events before they happen. Chapter 2 presented the theoretical background necessary for understanding the model. Chapter 3 described the systematic mapping study to understand the literature on SG techniques, technologies, challenges, and open issues. A comparative table between related works and the proposed model illustrates the contributions of this proposal. Chapter 4 introduced the Freya model, presenting the overview, architecture, agents, and Machine Learning pipeline. Chapter 5 showed OntoFreya, an ontology for electrical metrics classifications. Finally, chapter 6 depicts the prototype development and evaluation.

7.1 Conclusion

This thesis has presented the development and evaluation of the Freya model, a Machine Learning-based system designed to predict events in Smart Grids. The primary objective of this research was to create a computational model capable of evaluating a set of monitoring data from Smart Grids, allowing for the prediction of different network events and system states in a local and distributed manner.

The Freya model has successfully achieved all the outlined objectives:

- Create a computational model called Freya for event prediction's in Smart Grids
- Perform a literature review of computing techniques that support Smart Grids;
- Create an edge-computing component to perform event prediction in power distribution, according to the equipment context histories and at the edge of the Smart Grid;
- Propose a model for event prediction based on the energy flow and context of equipment within a power distribution Smart Grid;
- Build an ontology for power metrics classification according to the event of the equipment on the edge of the grid;
- Evaluate the Freya model through operational scenarios.

The evaluation of the Freya model demonstrated that it fulfills each of the five identified gaps in the literature:

- Entity Event Predictions: The model performs predictions for each entity in the network, providing a comprehensive approach to event detection.
- Network Event Predictions: Freya uses a model stacking technique to predict network events, addressing the need for holistic network-level event prediction.

- Dynamic Network Layouts: The model considers dynamic variations in the network layout, adapting to changes during operation and maintaining accurate event predictions.

- Transfer of Event Patterns: The stacking method transfers knowledge and detected patterns between network entities, enhancing the system's robustness.

- Model Retraining for Data Drift: The model incorporates mechanisms for constant retraining to account for data drift, ensuring sustained accuracy and reliability.

In conclusion, the Freya model meets the research objectives and addresses gaps in the current body of knowledge regarding SG event prediction. Its innovative approach, leveraging edge computing, adaptive Machine Learning, and ontology-based classification, represents a substantial advancement in the field. The Freya model offers a scalable and effective solution for real-time monitoring and prediction in SGs, contributing to improved power distribution network stability, reliability, and efficiency.

7.2 Future Works

There remains potential for future work to enhance the capabilities of the Freya model further. Three main areas of future research have been identified:

Exploration of context history: Developing a metric to define more and less critical contexts according to the attributes of their entities providing a nuanced understanding of the SG's operational environment. This process enables the model to prioritize and respond more effectively to critical situations.

Creating a history of events: By establishing a comprehensive history of network events and using this data to train and infer event classifications, the Freya model enhanced its ability to name and identify events accurately. This historical context improved the precision and reliability of event prediction, instilling confidence in the model's capabilities.

Dynamic acquisition of equipment criticality: The Freya model's future enhancements was designed to adapt to dynamic factors such as the time of year (e.g., harvest season), proximity to essential services like hospitals, and the broader impact on the population. This adaptability allowed the model to adjust the criticality of equipment in real time, ensuring that the Smart Grid responds appropriately to varying levels of demand and risk throughout the year, providing a sense of reassurance about its effectiveness in different scenarios.

Development of a Context-Sensitive Metric: Combining metrics and context awareness could create a formula or metric specifically designed to generate an output related to the entities' context could be developed. This formula would be beneficial as each entity is geographically distant, and local conditions such as storms could significantly influence how events occur for each entity. Creating such a metric would allow the model to incorporate local environmental factors into its predictions, enhancing the accuracy and responsiveness of the Smart Grid.

Deploying the model in an operational environment: It involves predicting real-time events. This step is crucial for validating the model's effectiveness in practical applications and ensuring its reliability in dynamic and complex SG environments. This deployment would allow for continuous monitoring and adjustment of predictions, providing real-time insights and responses to emerging events.

7.3 Produced Articles

Articles Related with the thesis

Published Article: ARANDA, Jorge Arthur Schneider; BAVARESCO, Rodrigo Simon; DE CARVALHO, Juliano Varella; YAMIN, Adenauer Corrêa; TAVARES, Mauricio Campelo; BARBOSA, Jorge Luis Victória. A computational model for adaptive recording of vital signs through context histories. Journal of Ambient Intelligence and Humanized Computing. [S. l.]: Springer Science and Business Media LLC, 18 mar. 2021. DOI 10.1007/s12652-021-03126-8. Disponível em: http://dx.doi.org/10.1007/s12652-021-03126-8.

Published Article: ARANDA, Jorge Arthur Schneider; DOS SANTOS COSTA, Ricardo; DE VARGAS, Vitor Werner; DA SILVA PEREIRA, Paulo Ricardo; BARBOSA, Jorge Luis Victória; VIANNA, Marcelo Pinto. Context-aware Edge Computing and Internet of Things in Smart Grids: A systematic mapping study. Computers and Electrical Engineering. [S. l.]: Elsevier BV, Apr. 2022. DOI 10.1016/j.compeleceng.2022.107826. Available at: http://dx.doi.org/10.1016/j.compeleceng.2022.107826.

In Submission Process: ARANDA, Jorge Arthur Schneider; DOS SANTOS COSTA, Ricardo; DE VARGAS, Vitor Werner; DA SILVA PEREIRA, Paulo Ricardo; BARBOSA, Jorge Luis Victória; VIANNA, Marcelo Pinto; MARQUES DA SILVA, Eleandro Luis. OntoFreya: A Power distribution ontology for electric metrics classification. Electric Power Systems Research: Elsevier BV.

In Submission Process: ARANDA, Jorge Arthur Schneider; DOS SANTOS COSTA, Ricardo; DE VARGAS, Vitor Werner; DA SILVA PEREIRA, Paulo Ricardo; BARBOSA, Jorge Luis Victória; VIANNA, Marcelo Pinto; BITTENCOURT DE QUADROS, Geison. A Dynamic Hierarquical Model for Event Detector in Power Distribution Systems. Journal of Engineering Applications of Artificial Intelligence: Elsevier BV.

Colaboration articles

Published Article DE VARGAS, Vitor Werner; ARANDA, Jorge Arthur Schneider;DOS

SANTOS COSTA, Ricardo; DA SILVA PEREIRA, Paulo Ricardo; BARBOSA, Jorge Luis Victória. Imbalanced data preprocessing techniques for Machine Learning: a systematic mapping study. Knowledge And Information Systems. Springer Link.

Published Article DOS SANTOS COSTA, Ricardo; DA SILVA PEREIRA, Paulo Ricardo; MADRUGA, Ederson Pereira; ARANDA, Jorge Arthur Schneider; BARBOSA, Jorge Luis Victória. A Hybrid Model for Investment Prioritization and Performance Analysis in Electrical Power Distribution Systems. Journal of Control, Automation and Electrical Systems. Springer Link.

In Submission Process: BORGES, Carlos Eduardo Liedtke ; BARBOSA, Jorge Luis Victória ; ARANDA, Jorge Arthur Schneider. Analysis of Human Mobility Behavior in Smart Cities and Smart Environments: A Systematic Literature Review and Taxonomy: Taylor & Francis.

REFERENCES

ABDELLATIF, A. et al. Computational detection and interpretation of heart disease based on conditional variational auto-encoder and stacked ensemble-learning framework. **Biomedical Signal Processing and Control**, [S.l.], v. 88, p. 105644, 2024.

AGGARWAL, S. et al. An artificial intelligence-based stacked ensemble approach for prediction of protein subcellular localization in confocal microscopy images. **Sustainability**, [S.l.], v. 15, n. 2, p. 1695, 2023.

AHMAD, T. et al. Smart energy forecasting strategy with four machine learning models for climate-sensitive and non-climate sensitive conditions. **Energy**, [S.l.], v. 198, p. 117283, May 2020.

ALAZAB, M. et al. A multidirectional LSTM model for predicting the stability of a smart grid. **IEEE Access**, [S.l.], v. 8, p. 85454–85463, 2020.

ALDEGHEISHEM, A. et al. Towards sustainable energy efficiency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks. **IEEE Access**, [S.l.], v. 9, p. 25036–25061, 2021.

ALHANAF, A. S.; BALIK, H. H.; FARSADI, M. Intelligent fault detection and classification schemes for smart grids based on deep neural networks. **Energies**, [S.l.], v. 16, n. 22, p. 7680, Nov. 2023.

ALHATHLOUL, S. H.; MISHRA, A. K.; KHAN, A. A. Low visibility event prediction using random forest and k-nearest neighbor methods. **Theoretical and Applied Climatology**, [S.l.], v. 155, n. 2, p. 1289–1300, 2024.

ANEEL. **Agência nacional de energia elétrica - resolução normativa aneel nº 956, de 7 de dezembro de 2021**. [Accessed 03-07-2024], `https://www2.aneel.gov.br/cedoc/ren2021956.html`.

ARANDA, J. A. S. et al. Collection and analysis of physiological data in smart environments: a systematic mapping. **(Journal of Ambient Intelligence and Humanized Computing**, [S.l.], v. 11, n. 7, p. 2883–2897, July 2019.

ARANDA, J. A. S. et al. Context-aware edge computing and internet of things in smart grids: a systematic mapping study. **Computers and Electrical Engineering**, [S.l.], v. 99, p. 107826, Apr. 2022.

BARBOSA, J.; FILIPPETTO, A.; LIMA, R. Átropos: towards a risk prediction model for software project management. **International Journal of Agile Systems and Management**, [S.l.], v. 13, n. 3, p. 296, 2020.

BARBOSA, J. L. V. Ubiquitous computing: applications and research opportunities. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND COMPUTING RESEARCH (ICCIC), 2015., 2015. **Anais. . .** [S.l.: s.n.], 2015. p. 1–8.

BAVARESCO, R. et al. Conversational agents in business: a systematic literature review and future research directions. **Computer Science Review**, [S.l.], v. 36, p. 100239, May 2020.

BHATTACHARYA, S. et al. Incentive mechanisms for smart grid: state of the art, challenges, open issues, future directions. **Big Data and Cognitive Computing**, [S.l.], v. 6, n. 2, p. 47, apr 2022.

BIN, X. et al. Research and application of micro-service platform based on ubiquitous power internet of things. In: INTERNATIONAL CONFERENCE ON BIG DATA TECHNOLOGIES - ICBDT2019, 2., 2019. **Proceedings...** ACM Press, 2019.

BOUQUET, P. et al. Ai-based forecasting for optimised solar energy management and smart grid efficiency. **International Journal of Production Research**, [S.l.], v. 62, n. 13, p. 4623–4644, 2024.

BUDGEN, D. et al. Using mapping studies in software engineering. **Proceedings of PPIG 2008**, [S.l.], v. 2, 01 2008.

BUTT, O. M.; ZULQARNAIN, M.; BUTT, T. M. Recent advancement in smart grid technology: future prospects in the electrical power network. **Ain Shams Engineering Journal**, [S.l.], v. 12, n. 1, p. 687–695, Mar. 2021.

CAO, K. et al. An overview on edge computing research. **IEEE Access**, [S.l.], v. 8, p. 85714–85728, 2020.

CARVALHO, E. L. N. et al. A proposal for power quality management system. In: INTERNATIONAL CONFERENCE ON HARMONICS AND QUALITY OF POWER (ICHQP), 2020., 2020. **Anais...** IEEE, 2020.

CARVALHO, O.; ROLOFF, E.; NAVAUX, P. O. A distributed stream processing based architecture for IoT smart grids monitoring. In: COMPANION PROCEEDINGS OF THE10TH INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING, 2017. **Anais...** ACM, 2017.

CHEN, S. et al. Internet of things based smart grids supported by intelligent edge computing. **IEEE Access**, [S.l.], v. 7, p. 74089–74102, 2019.

CHEN, X.; LI, X.; HUANG, Z. Exploration of power consumption monitoring based on internet of things. **Concurrency and Computation: Practice and Experience**, [S.l.], v. 31, n. 10, p. e4737, Oct. 2018.

CHENG, L.; YU, T. A new generation of AI: a review and perspective on machine learning technologies applied to smart energy and electric power systems. **International Journal of Energy Research**, [S.l.], v. 43, n. 6, p. 1928–1973, Jan. 2019.

CHONG, A. T. Y. et al. Description for smart grid: towards the ontological approach. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY AND MULTIMEDIA (ICIMU), 2020., 2020. **Anais...** IEEE, 2020.

CHRISTOU, E.; PARMAXI, A.; ZAPHIRIS, P. A systematic exploration of scoping and mapping literature reviews. **Universal Access in the Information Society**, [S.l.], p. 1–11, 2024.

COOPER, I. D. What is a mapping study ? . **Journal of Medicine Library Association**, [S.l.], v. 104, n. January, p. 76–78, 2016.

DALCEKOVIC, N. et al. Providing flexible software as a service for smart grid by relying on big data platforms. In: INTERNATIONAL SYMPOSIUM ON NETWORKS, COMPUTERS AND COMMUNICATIONS (ISNCC), 2017., 2017. **Anais. . .** IEEE, 2017.

DALMINA, L.; BARBOSA, J. L. V.; VIANNA, H. D. A systematic mapping study of gamification models oriented to motivational characteristics. **Behaviour & Information Technology**, [S.l.], v. 0, n. 0, p. 1–18, 2019.

DAVOODY-BENI, Z. et al. Application of IoT in smart grid: challenges and solutions. In: IRANIAN CONFERENCE ON SIGNAL PROCESSING AND INTELLIGENT SYSTEMS (ICSPIS), 2019., 2019. **Anais. . .** IEEE, 2019.

DAYANANDA, P. et al. A machine learning-based smart grid protection and control framework using kalman filters for enhanced power management. **Journal Européen des Systèmes Automatisés**, [S.l.], v. 57, n. 3, p. 639–651, June 2024.

DEY, A. K.; ABOWD, G. D.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Human Computer Interaction**, [S.l.], v. 16, n. 2-4, p. 97–166, Dec. 2001.

DIAS, L. P. S.; BARBOSA, J. L. V.; VIANNA, H. D. Gamification and serious games in depression care: a systematic mapping study. **Telematics and Informatics**, [S.l.], v. 35, n. 1, p. 213–224, Apr. 2018.

DUAN, Q. et al. Combining federated learning and edge computing toward ubiquitous intelligence in 6g network: challenges, recent advances, and future directions. **IEEE Communications Surveys & Tutorials**, [S.l.], 2023.

DUEÑAS, J. C. et al. An event stream architecture for the distributed inference execution of predictive monitoring models. **Authorea Preprints**, [S.l.], 2024.

EL MOKADEM, R.; BEN MAISSA, Y.; EL AKKAOUI, Z. Federated learning for energy constrained devices: a systematic mapping study. **Cluster Computing**, [S.l.], v. 26, n. 2, p. 1685–1708, 2023.

GHORBANALI, A.; SOHRABI, M. K. Capsule network-based deep ensemble transfer learning for multimodal sentiment analysis. **Expert Systems with Applications**, [S.l.], v. 239, p. 122454, 2024.

GONCALES, L. et al. Model comparison: a systematic mapping study. **PIPCA, University of Vale do Rio dos Sinos**, [S.l.], p. 6, 2014.

GOPARAJU, S. U. N. et al. Design of an iot system for machine learning calibrated tds measurement in smart campus. In: IEEE 7TH WORLD FORUM ON INTERNET OF THINGS (WF-IOT), 2021., 2021. **Anais. . .** IEEE, 2021.

GORE, R. N.; SAWAI, S.; KOUR, H. Grid on tab: IoT based quick, mobile and smart grid analyzer. In: INNOVATIONS IN POWER AND ADVANCED COMPUTING TECHNOLOGIES (I-PACT), 2019., 2019. **Anais. . .** IEEE, 2019.

HAJJAJI, Y. et al. Big data and iot-based applications in smart environments: a systematic review. **Computer Science Review**, [S.l.], v. 39, p. 100318, 2021.

HASNAT, M. A. et al. Situational awareness using edge-computing enabled internet of things for smart grids. In: IEEE GLOBECOM WORKSHOPS (GC WKSHPS), 2019., 2019. **Anais...** IEEE, 2019.

HAUER, D. et al. Context aware monitoring for smart grids. In: IEEE 30TH INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS (ISIE), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p. 1–6.

HE, M. et al. Causal feature selection for physical sensing data. In: ADJUNCT PROCEEDINGS OF THE 2019 ACM INTERNATIONAL JOINT CONFERENCE ON PERVASIVE AND UBIQUITOUS COMPUTING AND PROCEEDINGS OF THE 2019 ACM INTERNATIONAL SYMPOSIUM ON WEARABLE COMPUTERS, 2019. **Anais...** ACM, 2019.

HECKLER, W. F. et al. Kairós: using context histories for predictions and recommendations in projects time management. **International Journal of Agile Systems and Management**, [S.l.], v. 15, n. 1, p. 31, 2022.

HUANG, Y. et al. An edge computing framework for real-time monitoring in smart grid. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INTERNET (ICII), 2018., 2018. **Anais...** IEEE, 2018.

IBRAHIM, M. S.; DONG, W.; YANG, Q. Machine learning driven smart electric power systems: current trends and new perspectives. **Applied Energy**, [S.l.], v. 272, p. 115237, Aug. 2020.

JAIN, A.; BHULLAR, S. Online management and assessment of power quality issues through smart metering in smart grid architecture. **Energy Reports**, [S.l.], v. 8, p. 613–631, nov 2022.

JAMES, K. L.; RANDALL, N. P.; HADDAWAY, N. R. A methodology for systematic mapping in environmental sciences. **Environmental Evidence (2016)**, [S.l.], v. 5, n. 1, Apr. 2016.

JUNIOR, W. L. R. et al. Low voltage smart meter for monitoring of power quality disturbances applied in smart grid. **Measurement**, [S.l.], v. 147, p. 106890, Dec. 2019.

JURADO, S. et al. Fuzzy inductive reasoning forecasting strategies able to cope with missing data: a smart grid application. **Applied Soft Computing**, [S.l.], v. 51, p. 225–238, Feb. 2017.

KAVYA, B. M. et al. A machine learning based smart grid for home power management using cloud-edge computing system. In: INTERNATIONAL CONFERENCE ON RECENT ADVANCES IN SCIENCE AND ENGINEERING TECHNOLOGY (ICRASET), 2023., 2023. **Anais...** IEEE, 2023.

KERSTING, W. H. **Distribution system modeling and analysis**. [S.l.]: CRC Press, 2017.

KESHAV, S.; S. How to read a paper. **ACM SIGCOMM Computer Communication Review**, [S.l.], v. 37, n. 3, p. 83, 2016.

KHAN, I. et al. Does energy trilemma a driver of economic growth? the roles of energy use, population growth, and financial development. **Renewable and Sustainable Energy Reviews**, [S.l.], v. 146, p. 111157, aug 2021.

KHAOUAT, A. E.; BENHLIMA, L. Big data based management for smart grids. In: INTERNATIONAL RENEWABLE AND SUSTAINABLE ENERGY CONFERENCE (IRSEC), 2016., 2016. **Anais…** IEEE, 2016.

KRČ, R. et al. Machine learning-based node characterization for smart grid demand response flexibility assessment. **Sustainability**, [S.l.], v. 13, n. 5, p. 2954, Mar. 2021.

KRIVOHLAVA, Z.; CHREN, S.; ROSSI, B. Failure and fault classification for smart grids. **Energy Informatics**, [S.l.], v. 5, n. 1, p. 33, 2022.

KSERAWI, F.; AL-MARRI, S.; MALLUHI, Q. Privacy-preserving fog aggregation of smart grid data using dynamic differentially-private data perturbation. **IEEE Access**, [S.l.], v. 10, p. 43159–43174, 2022.

KULKARNI, S. et al. Enabling a decentralized smart grid using autonomous edge control devices. **IEEE Internet of Things Journal**, [S.l.], v. 6, n. 5, p. 7406–7419, Oct. 2019.

LAGHARI, A. A.; JUMANI, A. K.; LAGHARI, R. A. Review and state of art of fog computing. **Archives of Computational Methods in Engineering**, [S.l.], v. 28, n. 5, p. 3631–3643, 2021.

LAMPROPOULOS, G. Artificial intelligence in smart grids: a bibliometric analysis and scientific mapping study. **Journal of Mechatronics, Electrical Power, and Vehicular Technology**, [S.l.], v. 14, n. 1, p. 11–34, 2023.

LARIOUI, J.; BYED, A. E. An agent-based architecture for multi-modal transportation using prometheus methodology design. In: **Innovations in smart cities applications volume 4**. [S.l.]: Springer International Publishing, 2021. p. 325–343.

LAZZARINI, R.; TIANFIELD, H.; CHARISSIS, V. A stacking ensemble of deep learning models for iot intrusion detection. **Knowledge-Based Systems**, [S.l.], v. 279, p. 110941, 2023.

LEE, C. C.; YUAN, Z.; WANG, Q. How does information and communication technology affect energy security? international evidence. **Energy Economics**, [S.l.], v. 109, p. 105969, May 2022.

LEIGHTON, K. et al. Using social media and snowball sampling as an alternative recruitment strategy for research. **Clinical Simulation in Nursing**, [S.l.], v. 55, p. 37–42, June 2021.

LI, H.; ARMIENTO, R.; LAMBRIX, P. A method for extending ontologies with application to the materials science domain. **Data Science Journal (2019)**, [S.l.], v. 18, 2019.

LI, H.; OTA, K.; DONG, M. Learning iot in edge: deep learning for the internet of things with edge computing. **IEEE Network**, [S.l.], v. 32, n. 1, p. 96–101, Jan. 2018.

LIMA, R. et al. Towards ubiquitous requirements engineering through recommendations based on context histories. **PeerJ Computer Science**, [S.l.], v. 8, p. e794, Jan. 2022.

LIU, B. et al. Fault diagnosis of photovoltaic strings by using machine learning-based stacking classifier. **IET Renewable Power Generation**, [S.l.], v. 18, n. 3, p. 384–397, 2024.

LIU, X. et al. Edge sensing data-imaging conversion scheme of load forecasting in smart grid. **Sustainable Cities and Society**, [S.l.], v. 62, p. 102363, Nov. 2020.

LIU, Y. et al. DGFIndex for smart grid. **Proceedings of the VLDB Endowment**, [S.l.], v. 7, n. 13, p. 1496–1507, Aug. 2014.

MARTINELLI, F.; MERCALDO, F.; SANTONE, A. Water meter reading for smart grid monitoring. **Sensors**, [S.l.], v. 23, n. 1, p. 75, Dec. 2022.

MARTINI, B. G. et al. IndoorPlant: a model for intelligent services in indoor agriculture based on context histories. **Sensors**, [S.l.], v. 21, n. 5, p. 1631, Feb. 2021.

MHDAWI, A. K. A.; AL-RAWESHIDY, H. S. A smart optimization of fault diagnosis in electrical grid using distributed software-defined IoT system. **IEEE Systems Journal**, [S.l.], v. 14, n. 2, p. 2780–2790, June 2020.

MIHAILESCU, R.-C.; OSSOWSKI, S.; KLUSCH, M. ECOOP: applying dynamic coalition formation to the power regulation problem in smart grids. **Computational Intelligence**, [S.l.], v. 33, n. 3, p. 401–427, June 2016.

MIMI, S.; BEN MAISSA, Y.; TAMTAOUI, A. Optimization approaches for demand-side management in the smart grid: a systematic mapping study. **Smart Cities**, [S.l.], v. 6, n. 4, p. 1630–1662, 2023.

MOUSAVI, M. J.; STOUPIS, J.; SAARINEN, K. Event zone identification in electric utility systems using statistical machine learning. In: IEEE/PES TRANSMISSION AND DISTRIBUTION CONFERENCE AND EXPOSITION (T&D), 2018., 2018. **Anais...** IEEE, 2018.

MUKHERJEE, A. et al. Lightweight sustainable intelligent load forecasting platform for smart grid applications. **Sustainable Computing: Informatics and Systems**, [S.l.], v. 25, p. 100356, Mar. 2020.

MUQEET, H. A. et al. A state-of-the-art review of smart energy systems and their management in a smart grid environment. **Energies**, [S.l.], v. 16, n. 1, p. 472, 2023.

MUSEN, M. A. The protégé project. **AI Matters**, [S.l.], v. 1, n. 4, p. 4–12, June 2015.

NEPSHA, F. S. et al. Development of an ontology for smart distributed energy systems. **IFAC-PapersOnLine**, [S.l.], v. 55, n. 9, p. 454–459, 2022.

NEWAZ, S. H. S. et al. A web based energy cloud platform for campus smart grid for understanding energy consumption profile and predicting future energy demand. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2014., 2014. **Anais...** IEEE, 2014.

NIU, S. et al. Fault detection in power distribution networks based on comprehensive-yolov5. **Sensors**, [S.l.], v. 23, n. 14, p. 6410, July 2023.

NTI, I. K. et al. Applications of artificial intelligence in engineering and manufacturing: a systematic review. **Journal of Intelligent Manufacturing (2021)**, [S.l.], Apr. 2021.

OLATUNDE, T. M. et al. The impact of smart grids on energy efficiency: a comprehensive review. **Engineering Science & Technology Journal**, [S.l.], v. 5, n. 4, p. 1257–1269, 2024.

OMARA, A. et al. Microgrid data aggregation and wireless transfer scheduling in the presence of time sensitive events. In: ACM INTERNATIONAL SYMPOSIUM ON MOBILITY MANAGEMENT AND WIRELESS ACCESS - MobiWac'18, 16., 2018. **Proceedings...** ACM Press, 2018.

ÖNDER, M.; DOGAN, M. U.; POLAT, K. Classification of smart grid stability prediction using cascade machine learning methods and the internet of things in smart grid. **Neural Computing and Applications**, [S.l.], v. 35, n. 24, p. 17851–17869, May 2023.

PAL, C.; SHANKAR, R. A hierarchical performance evaluation approach for the sustainability of smart grid. **International Journal of Energy Sector Management**, [S.l.], n. ahead-of-print, 2022.

PETERSEN, K. et al. Systematic Mapping Studies in Software Engineering. **12Th International Conference on Evaluation and Assessment in Software Engineering**, [S.l.], v. 17, p. 10, 2008.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: an update. **Information and Software Technology**, [S.l.], v. 64, p. 1–18, 2015.

PINCIROLI, F.; JUSTO, J. L. B.; FORRADELLAS, R. Systematic mapping study: on the coverage of aspect-oriented methodologies for the early phases of the software development life cycle. **Journal of King Saud University - Computer and Information Sciences (2020)**, [S.l.], Nov. 2020.

PRITONI, M. et al. Metadata schemas and ontologies for building energy applications: a critical review and use case analysis. **Energies**, [S.l.], v. 14, n. 7, p. 2024, Apr. 2021.

QADIR, Z. et al. Predicting the energy output of hybrid PV–wind renewable energy system using feature selection technique for smart grids (2021). **Energy Reports**, [S.l.], Jan. 2021.

QURESHI, N. M. F. et al. An aggregate MapReduce data block placement strategy for wireless IoT edge nodes in smart grid. **Wireless Personal Communications**, [S.l.], v. 106, n. 4, p. 2225–2236, Sept. 2018.

RABIE, A. H. et al. A fog based load forecasting strategy for smart grids using big electrical data. **Cluster Computing**, [S.l.], v. 22, n. 1, p. 241–270, Sept. 2018.

RABIE, A. H. et al. A new outlier rejection methodology for supporting load forecasting in smart grids based on big data. **Cluster Computing**, [S.l.], v. 23, n. 2, p. 509–535, May 2019.

RAJADURAI, H.; GANDHI, U. D. A stacked ensemble learning model for intrusion detection in wireless network. **Neural computing and applications**, [S.l.], v. 34, n. 18, p. 15387–15395, 2022.

RAJU, L. et al. Application of machine learning algorithms for short term load prediction of smart grid. In: INTERNATIONAL CONFERENCE ON SMART ELECTRONICS AND COMMUNICATION (ICOSEC), 2020., 2020. **Anais...** IEEE, 2020.

RASTOGI, D. et al. The role of humidity in determining future electricity demand in the southeastern united states. **Environmental Research Letters**, [S.l.], v. 16, n. 11, p. 114017, oct 2021.

REMIGIO-CARMONA, P. et al. Current status and future trends of power quality analysis. **Energies**, [S.l.], v. 15, n. 7, p. 2328, mar 2022.

RENTZ, D. M.; HECKLER, W. F.; BARBOSA, J. L. V. A computational model for assisting individuals with suicidal ideation based on context histories. **Universal Access in the Information Society**, [S.l.], Mar. 2023.

SAHINER, B. et al. Data drift in medical machine learning: implications and potential remedies. **The British Journal of Radiology**, [S.l.], v. 96, n. 1150, Oct. 2023.

SAKHNINI, J. et al. Security aspects of internet of things aided smart grids: a bibliometric survey. **Internet of things**, [S.l.], v. 14, p. 100111, 2021.

SALAMEH, K.; CHBEIR, R.; CAMBLONG, H. SSG: an ontology-based information model for smart grids. In: **Lecture notes in computer science**. [S.l.]: Springer Berlin Heidelberg, 2019. p. 94–124.

SAMANN, F. E. F.; ABDULAZEEZ, A. M.; ASKAR, S. Fog computing based on machine learning: a review. **International Journal of Interactive Mobile Technologies (iJIM)**, [S.l.], v. 15, n. 12, p. 21, jun 2021.

SARKODIE, S. A.; AHMED, M. Y.; OWUSU, P. A. Ambient air pollution and meteorological factors escalate electricity consumption. **Science of The Total Environment**, [S.l.], v. 795, p. 148841, Nov. 2021.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **IEEE Personal Communications**, [S.l.], v. 8, n. 4, p. 10–17, Aug 2001.

SATYANARAYANAN, M. The emergence of edge computing. **Computer**, [S.l.], v. 50, n. 1, p. 30–39, Jan. 2017.

SAXENA, S.; FARAG, H. E.; EL-TAWEEL, N. A distributed communication framework for smart grid control applications based on data distribution service. **Electric Power Systems Research**, [S.l.], v. 201, p. 107547, dec 2021.

SCHACHINGER, D.; KASTNER, W.; GAIDA, S. Ontology-based abstraction layer for smart grid interaction in building energy management systems. In: IEEE INTERNATIONAL ENERGY CONFERENCE (ENERGYCON), 2016., 2016. **Anais...** IEEE, 2016.

SHEHU, Y.; HARPER, R. Efficient periodicity analysis for real-time anomaly detection. In: NOMS 2023-2023 IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, 2023. **Anais...** IEEE, 2023.

SHI, Z. et al. Artificial intelligence techniques for stability analysis and control in smart grids: methodologies, applications, challenges and future directions. **Applied Energy**, [S.l.], v. 278, p. 115733, Nov. 2020.

SHOBOLE, A. A.; WADI, M. Multiagent systems application for the smart grid protection. **Renewable and Sustainable Energy Reviews**, [S.l.], v. 149, p. 111352, oct 2021.

SKRETA, M. et al. Automatically disambiguating medical acronyms with ontology-aware deep learning. **Nature Communications**, [S.l.], v. 12, n. 1, Sept. 2021.

SMIRNOV, A. et al. Methodology for multi-aspect ontology development. In: **Lecture notes in business information processing**. [S.l.]: Springer International Publishing, 2021. p. 97–109.

SOURI, A. et al. A systematic review of iot communication strategies for an efficient smart environment. **Transactions on Emerging Telecommunications Technologies**, [S.l.], v. 33, n. 3, p. e3736, 2022.

SOUZA, C. B. de; BADYINA, A.; GOLUBCHIKOV, O. Ontoagency: an agency-based ontology for tracing control, ownership and decision-making in smart buildings. **Building and Environment**, [S.l.], v. 256, p. 111489, 2024.

SOYKAN, E. U. et al. Differentially private deep learning for load forecasting on smart grid. In: IEEE GLOBECOM WORKSHOPS (GC WKSHPS), 2019., 2019. **Anais...** IEEE, 2019.

SVENSSON, J. et al. Wind power distribution across subalpine, boreal, and temperate landscapes. **Ecology and Society**, [S.l.], v. 28, n. 4, 2023.

TASFI, N. L. et al. Deep neural networks with confidence sampling for electrical anomaly detection. In: IEEE INTERNATIONAL CONFERENCE ON INTERNET OF THINGS (iThings) AND IEEE GREEN COMPUTING AND COMMUNICATIONS (GreenCom) AND IEEE CYBER, PHYSICAL AND SOCIAL COMPUTING (CPSCom) AND IEEE SMART DATA (SmartData), 2017., 2017. **Anais...** IEEE, 2017.

THAM, J. C.; VERHULSDONCK, G. Smart education in smart cities: layered implications for networked and ubiquitous learning. **IEEE Transactions on Technology and Society**, [S.l.], v. 4, n. 1, p. 87–95, 2023.

TIWARI, S. et al. Machine learning-based model for prediction of power consumption in smart grid-smart way towards smart city. **Expert Systems**, [S.l.], v. 39, n. 5, p. e12832, 2022.

VANTUCH, T. et al. Machine learning based electric load forecasting for short and long-term period. In: IEEE 4TH WORLD FORUM ON INTERNET OF THINGS (WF-IoT), 2018., 2018. **Anais...** IEEE, 2018.

VIANNA, H. D.; BARBOSA, J. L. V. In search of computer-aided social support in non-communicable diseases care. **Telematics and Informatics**, [S.l.], v. 34, n. 8, p. 1419–1432, 2017.

VRITTI, V. et al. Energy management in iot-enabled smart grid: a review. **Artificial Intelligence and Society 5.0**, [S.l.], p. 118–132, 2024.

WADI, M. et al. Load frequency control in smart grids: a review of recent developments. **Renewable and Sustainable Energy Reviews**, [S.l.], v. 189, p. 114013, 2024.

WANG, L. et al. Three-dimensional maturity model of regional power users against the background of the ubiquitous power internet of things. **IEEE Access**, [S.l.], v. 8, p. 20215–20223, 2020.

WANG, X.; YAO, Z.; PAPAEFTHYMIOU, M. A real-time electrical load forecasting and unsupervised anomaly detection framework. **Applied Energy**, [S.l.], v. 330, p. 120279, Jan. 2023.

WEISER, M. The computer for the 21st century. **Scientific American**, [S.l.], v. 265, p. 94, 09 1991.

WU, Y. et al. Hierarchical operation of electric vehicle charging station in smart grid integration applications—an overview. **International Journal of Electrical Power & Energy Systems**, [S.l.], v. 139, p. 108005, 2022.

XIA, X. et al. Coded grouping-based inspection algorithms to detect malicious meters in neighborhood area smart grid. **Computers & Security**, [S.l.], v. 77, p. 547–564, Aug. 2018.

XU, C. et al. Review on interpretable machine learning in smart grid. **Energies**, [S.l.], v. 15, n. 12, p. 4427, jun 2022.

YADAV, M.; PAL, N.; SAINI, D. K. Low voltage ride through capability for resilient electrical distribution system integrated with renewable energy resources. **Energy Reports**, [S.l.], v. 9, p. 833–858, 2023.

YI, D. et al. Application of authorization in smart grid based on the pass microservice architecture. **IOP Conference Series: Earth and Environmental Science**, [S.l.], v. 512, n. 1, p. 012118, June 2020.

ZAINAB, A. et al. Big data management in smart grids: technologies and challenges. **IEEE Access**, [S.l.], v. 9, p. 73046–73059, 2021.

ZAINAB, A. et al. A multiprocessing-based sensitivity analysis of machine learning algorithms for load forecasting of electric power distribution system. **IEEE Access**, [S.l.], v. 9, p. 31684–31694, 2021.

ZAMAN, M. et al. Demand-response prediction in smart grids using machine learning techniques. In: IEEE POWER & ENERGY SOCIETY INNOVATIVE SMART GRID TECHNOLOGIES CONFERENCE (ISGT), 2024., 2024. **Anais...** [S.l.: s.n.], 2024. p. 1–5.

ZANABRIA, C. et al. A model-driven and ontology-based engineering approach for smart grid automation applications. In: IECON 2019 - 45TH ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY, 2019. **Anais...** IEEE, 2019.

Zhang, C. et al. A cooperative multi-agent deep reinforcement learning framework for real-time residential load scheduling. In: INTERNATIONAL CONFERENCE ON INTERNET OF THINGS DESIGN AND IMPLEMENTATION, 2019. **Proceedings...** ACM, 2019.

ZHEN, J. et al. Discussion about recent applications of power big data in smart grid. In: IEEE PES ASIA-PACIFIC POWER AND ENERGY ENGINEERING CONFERENCE (APPEEC), 2019., 2019. **Anais...** IEEE, 2019.

ZHENG, J. et al. Availability importance measures of components in smart electric power grid systems. **Reliability Engineering System Safety**, [S.l.], v. 205, p. 107164, Jan. 2021.