Universidade de Aveiro
2024

Miguel Rosas Marques

Deteção de zonas de calor e previsão de consumo elétrico

Electricity consumption hotspot detection and Prediction

Universidade de Aveiro
2024

Miguel Rosas Marques

Deteção de zonas de calor e previsão de consumo elétrico

Electricity consumption hotspot detection and Prediction

"*Do not go where the path may lead, go instead where there is no path and leave a trail*"

— Ralph Waldo Emerson

Universidade de Aveiro
2024

Miguel Rosas Marques

Deteção de zonas de calor e previsão de consumo elétrico

Electricity consumption hotspot detection and Prediction

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Computacional , realizada sob a orientação científica do Doutor José Maria Fernandes, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

This dissertation is dedicated to my parents, for their moral and emotional support and in the case of my parents also the financial support. I also want to dedicate to Professor José Maria Fernandes and António Santos for their support, as well the rest of the team I became a part of. Lastly I want to dedicate to all the people who I met and interacted, specially the friends I made, which led me taking the path I took, as this journey did not start on my master neither my bachelors, but much prior to that.

o júri / the jury

presidente / president	Prof. Doutor João Antunes da Silva professor associado da Faculdade de Engenharia da Universidade do Porto
vogais / examiners committee	Prof. Doutor João Antunes da Silva professor associado da Faculdade de Engenharia da Universidade do Porto
	Prof. Doutor João Antunes da Silva professor associado da Faculdade de Engenharia da Universidade do Porto

agradecimentos / acknowledgements

I would like to express my deepest appreciation to Professor José Maria Fernandes, Bosch, António Santos and Felipe Silva for the opportunity to work on this project. I am also grateful to the Professor José Maria Fernandes for taking the role of being my advisor, António Santos and Felipe Silva for welcoming me and making me part of the team, as well in the case of António being my company advisor. I would like to extend my sincere thanks to the team I became a part of, which although not directly involved in the master thesis, taught me about the professional world. I'm also grateful for friends, brother and parents, António Santos and the team for their support and moments.

Palavras Chave eletricidade, energia, previsão, zonas de calor.

Resumo

Redes elétricas são estruturas essenciais que desempenham um papel crucial nos dias de hoje, providenciando eletricidade para consumidores. Acesso a eletricidade melhora a qualidade de vida de indivíduos. A eletricidade ultrapassa indivíduos, uma vez que existem indústrias e negócios que dependem desta, fazendo com que seja importante manter as redes elétricas estáveis e consistentes. Com o aumento de consumo de energia, que é intensificado pela eletrificação de dispositivos, infraestruturas e sistemas, redes elétricas enfrentam um desafio para manter a estabilidade. A Bosch precisa de garantir a estabilidade de redes elétricas, baseando-se na monitorização dos equipamentos a múltiplos níveis geográficos, visto que as redes elétricas estão distribuídas espacialmente , código-postal, área, cidade, país, uma vez que é importante verificar se o consumo elétrico poderá comprometer a eletricidade de uma rua ou de uma zona maior. O objetivo desta dissertação é melhorar a estabilidade de redes elétricas, detectando distúrbios na variação de consumo em tempo real. Tendo em conta este objetivo, áreas com alta variação de consumo, picos são detectados, o que pode melhorar a distribuição de eletricidade e prevenir sobrecargas na rede. A previsão de consumo elétrico complementa o método anterior, uma vez que pode-se detetar distúrbios que ainda não acontecerem, e prevenir que esses distúrbios afetem a rede elétrica, melhorando então a estabilidade da mesma. A solução desenvolvida é capaz de detectar áreas a necessitar de eletricidade, que pode ser usado para organizar a eletricidade numa rede elétrica, redirecionando eletricidade para as áreas que a necessitam.

Keywords electricity, energy, hotspots, prediction.

Abstract

Power grids are essential infrastructures as they play a crucial role in modern society, providing electricity to consumers. Access to electricity enhances living standards, improving quality of life for individuals, enabling access to modern conveniences. Electricity goes beyond individuals as there are industries, businesses and commerce who depend on electricity, so it is critical to keep power grids stable and consistent. With the ever growing electrical consumption, further aggravated by the electrification of devices, infrastructures and systems, power grids face a challenge in order to keep the stability. Bosch needs to ensure the stability of dependent power grids, supported on the monitoring from the level of their equipment to network and geographical level, comprehending from postcode, area, city to country, as power grids are distributed spatially and it is important to not detect at very specific levels but also at more general areas, which can help detect if the consumption will compromise electricity in a street or a broader area. This dissertation aims to improve the stability of power grids, by detecting disturbances on power consumption in real time. To this end, areas with high disturbances, hotspots are determined, which can help optimize load sharing and improve the power grid stability and prevention of overloads on these networks. Furthermore the prediction of power consumption complements the hotspot detection algorithm, enabling a better response by applying the method on future values, and therefore detecting disturbances further in time, resulting on a better power grid stability, as there is more time to react to disturbances detected. The solution presented provides relevant information of areas requiring electricity, this knowledge can be utilized to manage the power grid electricity, redirecting to areas in need.

Contents

List of Figures

List of Tables

Glossary

IoT	Internet of Things
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
MAD	Median Absolite Deviation
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
GD	Gradient Descent
KNN	K-Nearest Neighbor
LOF	Local Outlier Factor
SVM	Support Vector Machine
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ANN	Artificial Neural Networks
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
FFNN	Feedforward Neural Networks
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Network
AE	AutoEnconder
ReLU	Rectified Linear Unit
KNN	K-Nearest Neighbors
ELU	Exponential Linear Unit

CHAPTER 1

Introduction

With the production and deployment of thousands of Bosch heat pumps [1], each one equipped with multiple sensors, generating large quantities of data. This data in their raw format might not have much value, but it can be processed to extract valuable information and patterns. With this in mind, Bosch wants to monitor their appliances consumption in order to ensure better power grids management. The monitorization will focus on hotspot detection geographically, discerning areas of high and low variation on power consumption. The solution developed should receive the data from the appliances through the gateways that they have equipped. The data received should be stored, processed and visualized on a dashboard for a better understanding of the data.

The ever increasing concern with sustainable energy is evident across diverse industries and sectors, extending beyond individual endeavors to encompass corporations and governments.[2], [3]

Furthermore the electrification of industries, heating and transports led to an increase in energy consumption. Governments planned to institute limitations on the electrical grids in order to curtail energy consumption. This regulatory approach is designed to mitigate overall energy consumption. Additionally, the integration of Smart Grid technologies plays a pivotal role in this context, enabling enhanced management of electricity distribution and contributing to the overarching objective of optimizing energy efficiency within the contemporary energy landscape. Smart Grids allow electricity providers to monitor their grids, understanding the load of each individual network upon their grid. In order to provide electricity to every household, energy companies need to manage their grid, messaging users who are overloading the framework, regarding their electricity usage and warning them to reduce their consumption. If people don't comply with the warning, they can get fined. The European Union has been proactive on energy consumption, leading to the implementation of policies.[4], [5].

Smart Meter systems [6], [7] are able to measure electricity provided into the grid and used from the grid. This is beneficial for both the consumers and providers of energy[8]. These systems provide more rigorous measurements of energy consumption, resulting in an accurate electricity bill. Furthermore, this technology provides consumers with the option to adapt their energy usage, by consuming on periods of low electricity price. Additionally, information collected by Smart Meter can provide insights for citizens about the energy their household provides to the grid.

1.1 Smart Meter from an Internet of Things perspective

The implementation of Smart Meter requires data protection and security. In contemporary discourse, the Internet of Things (IoT) stands as a recurrent and prominent subject within various industries and sectors, with varying degrees of emphasis [4]–[8]. This paradigm facilitates the interconnection of sensors embedded in devices via the Internet and specific protocols, enabling seamless communication with other interconnected devices. The sensors capture relevant data, which is subsequently transmitted to other devices for processing, capable of generating Big Data. The IoT serves to remotely govern devices and gain comprehensive insights not only into the products and machines but also the end-users. These analytical outcomes hold substantial utility across diverse fields, contributing to enhancements in product quality, demand prediction, automation processes, and other strategic applications. The IoT enhances Smart Grids technologies. This interconnected system allows for monitoring, data collection, and analysis of grid performance metrics. This grants companies the ability to remotely monitor and control grid components, optimize energy distribution, and respond promptly to events in the system. To this end, in order to optimize electrical grids performance, it is important to identify spatially zones where there is a higher or lower consumption of electricity.

The implementation of Smart Meter requires data protection and security. In contemporary discourse, the IoT stands as a recurrent and prominent subject within various industries and sectors, with varying degrees of emphasis [9]–[13],. This paradigm facilitates the interconnection of sensors embedded in devices via the Internet and specific protocols, enabling seamless communication with other interconnected devices. The sensors capture relevant data, which is subsequently transmitted to other devices for processing, capable of generating Big Data. The Internet of Things serves to remotely govern devices and gain comprehensive insights not only into the products and machines but also the end-users. These analytical outcomes hold substantial utility across diverse fields, contributing to enhancements in product quality, demand prediction, automation processes, and other strategic applications. The Internet of Things enhances Smart Grids technologies. This interconnected system allows for monitoring, data collection, and analysis of grid performance metrics. This grants companies the ability to remotely monitor and control grid components, optimize energy distribution, and respond promptly to events in the system. To this end, in order to optimize electrical grids performance, it is important to identify spatially zones where there is a higher or lower consumption of electricity.

1.2 Scope

Germany has already implemented regulations regarding Smart Meters, citizens with a heat pump, more than 6000 kWh yearly consumption or plant operators with an installed capacity of more than 7kW need to be equipped with Smart Meter Gateways. [14]

Within business priorities of Bosch one is to provide a solution to detect and manage the unwanted power consumption patterns in time and on a multiple scale. The failure to ensure such patterns may be translated into fees or real and impactful commercial restrictions on the energy area.

1.3 Objectives

With the electrification of more and more products, Sustainable energy became an important topic in the present day. With the Sustainability in mind, Bosch developed heat pumps. These devices can be used to heat water and the house and are much more efficient when compared to other traditional devices. Each heat pump is connected through a gateway, transmitting metrics in real time, producing large quantities of data.

With this in mind, Bosch vision is to improve their product and also take advantage of the colossal data generated by heat pumps for a more sustainable world by monitoring their appliances energy consumption not necessarily at an individual level but as a collective. The solution developed should be able to identify hotspots and visualization of the information on a dashboard.

Departing from the Bosch perspective, the main objectives are:

- Integrate basic detection and prediction solutions for unwanted pattern detection in both time and space, optimizing power grids management, ideally preventive.
  - The detection method will focus on the identification of areas with high variation of power consumption as well areas of high consumption.
  - The prediction of appliances energy consumption.
- Provide a dashboard to visualize multiple metrics (power consumption, variation of consumption) spatially at multiple scales.

CHAPTER 2

State of Art

Today there are a variety of models for both hotspot detection and forecast. Furthermore there also a large variety of databases, each with their pros and cons. Performing experimental evaluation would not be feasible as they are too time consuming.

This chapter will review foundational studies and recent advances in the field, identifying prevailing theories, key methodologies, and gaps in the literature. By synthesizing these insights, this section aims to provide a comprehensive understanding of the current state of research on hotspot detection and forecast, reviewing similar cases to the one at hand.

Research on Artificial Intelligence which comprises Machine Learning and Deep Learning has rapidly expanded, fueled by the Artificial Intelligence (AI) boom. Scholars have approached the topic on different scenarios, nevertheless the results obtained by the researchers can be transversal to the problem of hotspots and energy forecast.

The literature review will also not be limited to the ML and DL, but will also focus on more traditional methods, namely statistics and graphs. Which will also give a comparison between the two different cases.

Furthermore there is a need to store time series data to enable historical data consultation, additionally it is also important to manage real-time data as there is a consistent flux of events. Therefore, multiple databases and data streaming technologies are going to presented in the literature review.

2.1 BOSCH and energy consumption Management

Dichotomization of BOSCH main objectives:

- Be able to handle different levels of abstract at geographical level
- Identify disturbances and hotspots on power grids in real time.
- Forecast of power consumption in real time.
- Identify abrupt changes in consumption at different levels
- Visualization of information pertinent to energy consumption on a dashboard.

With this in mind, the objectives is not only to detect individual changes of variation, but also areas, thereby detecting hotspots, at different spatial levels. These are related, as an area can be defined as a set of individual appliances.

2.2 The Main Concepts

Some basic concepts must be address prior to any solution proposal

- Power grid and its spatial nature and issues
- Disturbance on a power grid, variation of consumption and hotspots
- Prediction
- Real time detection

Power grid conveys electricity from power plants to homes and businesses. It interconnects the generation, transmission and distribution units. The power station that is in charge of generating energy is located far away from populated areas, near the fuel source. The electrical grid can be classified into two types:

- Regional grids interconnect different transmission systems of a partial area through a transmission line.
- National Grid, formed by interconnecting different regional grids.

The interconnection of the grid makes the system economical and reliable, reducing the reserve generation capacity in each area. As in the case of a sudden increase in load or loss of generation in a zone, it borrows from the adjacent interconnected area. It also reduces the reserved generation capacity, known as spinning reserve, which consists of a generator running and ready to supply power instantaneously.

Another important component of Power Grids are Grid Operators, also known as system operators, responsible for maintaining the electrical grid secure and reliable. They have a major role in balancing and ancillary service markets, to guarantee that the demand matches the supply and therefore ensure the system security. For this reason they have a need to constantly monitor the grid's performance.

Due to the interconnection and hierarchy of the Power grid components, the spatial component plays an important role in the grid operator's task. It is important to detect disturbances on the power grids at different scales spatially, ideally to match the Power Grids layout for more accurate results. As these disturbances are detected and the faster they are, the grid operators can respond quickly to the sudden changes on the electrical grids.

Power grids can have issues or failures, namely:

- Blackout is a loss of electric power in an area.
- Brownout is a drop in voltage in an electrical power power system. These can be intentional, for load reduction in an emergency or to prevent a total grid power outage (Blackout) due to high demand.
- Black start is the process utilized to recover from a shutdown, restoring an electric power station or part of the electric grid to operation, without relying on the external power system transmission network.

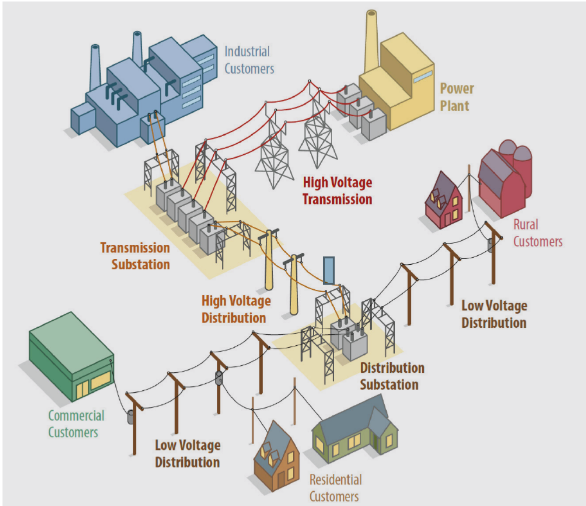


Figure 2.1: Power Grid Abstract Representation, from [15]

- Load shedding occurs when electrical generation and transmission systems may not meet peak demand requirements. In this case, an intentional Brownout might occur to prevent disruptions such as blackouts or equipment damage.

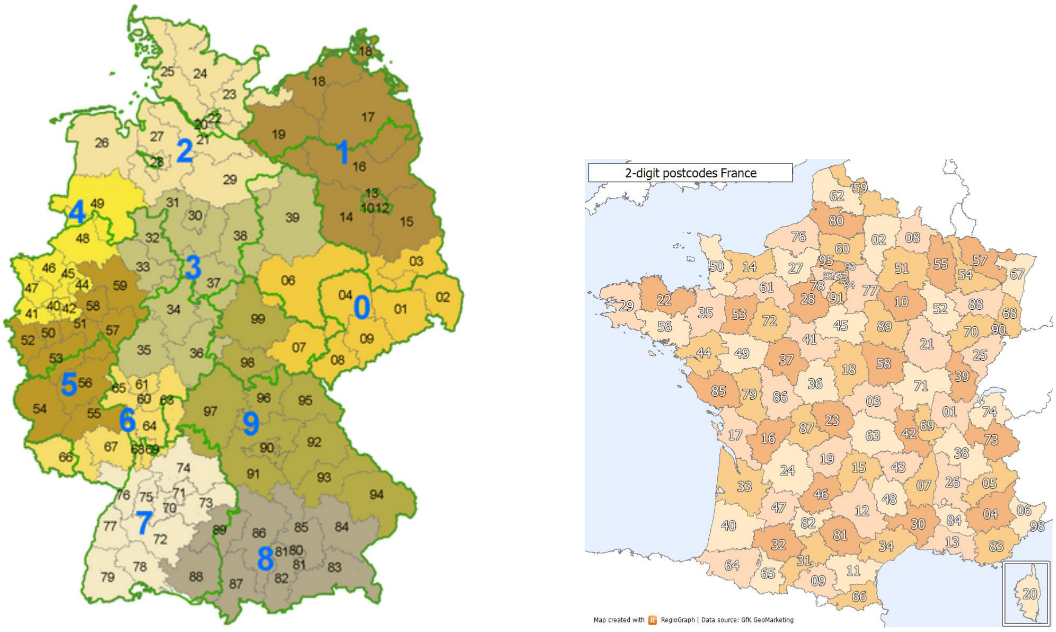
The detection of zones of high variation of energy consumption can help mitigate the last type of failure, which also implies reducing the frequency of blackouts and brownouts.

Different areas are defined, as there are different levels of hierarchy that need to be analyzed, these levels are defined using the postcode, and country. For each country, a fine grained mesh is made, grouping appliances by whole postcode, a higher level is accomplished by considering all digits except the last three, as postcodes format vary from country to country, getting a more generalized view of the network topology. At the top of the hierarchy is the country level for a more general view of the variation of consumption.

The figure 2.2 shows that Germany Layout is better defined spatially, as zones with the same first digit are next to each other, but in France that is not the case. So Germany could have another level of hierarchy, when compared to France. As the zip code format and layout can differ from country to country, it is important to define a hierarchy that works for any country. Using the postal code there is one option to define general areas that work for any given country which is to consider the first two digits of the zipcode, smaller areas can also be defined using the whole postcode. It is also possible to define areas using cities.

The following figure is an example of the hierarchy defined. At the top of the hierarchy are the countries, for each country there are two major levels, the cities level and the 2 digit postcode areas (general areas). Each of these areas is then composed by a list of postal codes, which represents the lowest level, as appliances spatial data is obtainable by their respective postcode, which can be cross referenced to obtain latitude and longitude coordinates.

The main idea of defining the hierarchy is to easily trace the root of Burden Hotspots on a more general view, by iteratively selecting a more fine grained grid. For example, at the country level, will be analyzed in order to detect areas of interest, representing Burden Hotspots. Then for a given area detected as hotspot, it is possible to visualize the major



(a) Germany, from [16]

(b) France, from [17]

Figure 2.2: Germany and France Postcode Layout for the first 2 digits

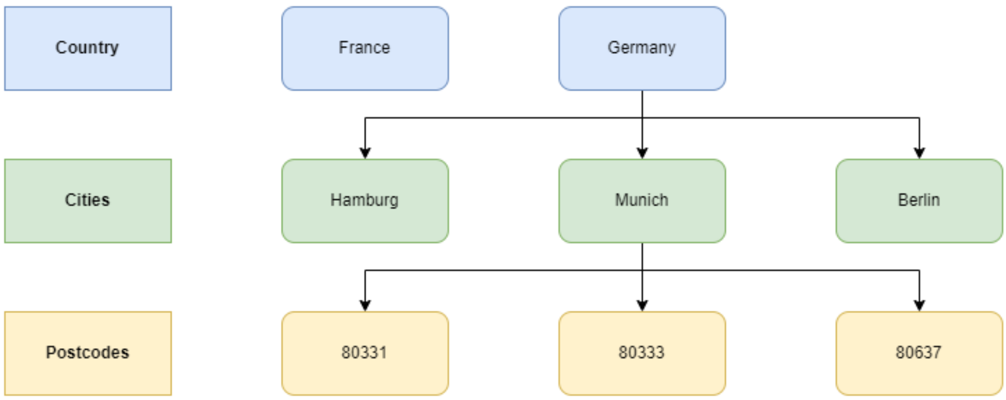


Figure 2.3: Example of the Defined Hierarchy, postcode 80331 is located in the City of Munich, that is located in Germany

contributors (postcodes) as they will also be detected as hotspots.

Variation of power consumption of an appliance can be calculated as the difference between the last two events of power consumption. A disturbance on a power grid can be defined as a high variation of power consumption of an appliance, sudden changes in the consumption are important, as they require more or less energy from a power grid, and help understand the flow of power needed on the network. It is also important to detect variations spatially, in order to locate areas who have a higher need of power, and optimize the power network.

A hotspot is an area or a point with high variation of power consumption, the temporal scale is the last value of variation of power consumption, as the focus is to analyze in real time. This translates to the sum of all the variation of power consumption of appliances according to their hierarchy group.

Horby et al. [18] states that the term hotspot is broadly used in research papers, leading to ambiguous definitions, in order to have a clear understanding, they advise the usage of a more precise term.

Furthermore it might be important to take into account the spatial dimension, to enable the detection of local hotspots. Horby et al.[18] suggests using a term according to the definition of hotspot, in this case, Burden Hotspots is adopted, as an area with high variation of consumption power represents a hotspot and also a burden by requiring more power from the grid.

Burden Hotspots could be easily visualized through a Kernel Density Estimation 2.4, but the goal of Burden Hotspots is not only visualization, but also the automatic identification. The latter can be used as an input for power grids management algorithms, in order to redirect or optimize the power grid.

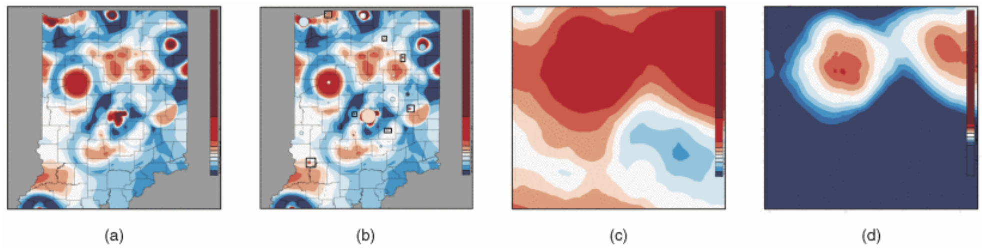


Figure 2.4: Kernel density estimate (KDE) heatmaps visualized as a percentage of syndromic population over the total population seen. (a) KDE heatmap. (b) Contextualizing the KDE heatmap by overlaying patient data aggregated through nearest neighbor groupings. (c) A zoomed in view of a local hotspot. (d) Contextualizing a hotspot through interactive coloring. Image obtained from [19].

Real time detection is a term used for software systems that are required to operate within strict time constraints. These real-time programs must ensure a guaranteed response within the specified time limits.

2.3 Normality Tests

There are multiple methods, Prabhaker Mishra et al. [20], presents methods to test the normality of the data. The authors categorize the tests to graphical and numerical (including statistical tests). As the test needs to be automated the numerical tests are the priority although they could be less sensitive.

Prabhaker Mishra et al. [20] present the most widely used methods to test the normality of the data, namely Kolmogorov-Smirnov test and Shapiro-Wilk test. The Shapiro-wilk is more capable for sample sizes ($n < 50$ observations) whereas Kolmogorov-Smirnov (KS test) more appropriate for larger sample sizes ($n > 50$ observations), but one of its limitations is the location, scale and shape that need to be known and cannot be estimated.

2.4 Prediction

The prediction of power consumption also goes in hand with the identification of disturbances on power grids, and this can be extended from real time assessment of power grids to one instant ahead [21], [22]. As the variance of power consumption can be calculated as the difference between the forecasted value and the current value of power consumption, therefore enabling a faster response from the power suppliers and a better management of their network.

2.5 Hotspot Detection Methods

Another important aspect of the pipeline is the detection of Burden hotspots. Sensors provide a collection of data, corresponding to different variables, resulting in complex patterns. Furthermore it's also important to deal with missing data, adding another layer of complexity to the model. Additionally in the real world, time series are typically non-stationary. Time series have temporal semantics, and due to the multi-dimensionality of the data, but it can also have external and spatial semantics.

For this problem, there are only temporal and external components. Some characteristics of temporal semantics are: trend, periodicity and seasonality. As for external factors are variables that influence the time series, but are not part of it. Another aspect to take into account is the complexity of the model, real-time analysis and monitoring of appliances for hotspot detection requires efficient algorithms, in order to achieve low latency.

One approach to detect hotspots is to detect anomalies in the grid [23]. The goal of anomaly detection is to detect rare instances (unusual or suspicious) that deviate from the norm. This approach regularly is an unsupervised task, where the model learns the normal behavior of the data, this is used to detect observations that deviate from the majority of the data. In this case anomalies represent households with either a higher or lower variation of consumption, when compared to other users on the same grid.

Another approach is to classify the variation of consumption on multiple degrees, according to the value of consumption variation, where each degree has a label. This method objective is to learn the label dependency on features to predict the class of unseen instances.

2.6 Traditional Approaches in Anomaly Detection: Statistical and Rule-Based Methods

There are Statistical methods for anomaly detection, these methods are very simple and may have restrictions on the data distribution. Some of these methods require Normal Distributions, however non-Gaussian distributions can be transformed into Normal Distributions through mathematical transformations. Another option to deal with the data distributions is to do a normality test and choose the adequate statistical method. Some of the statistical methods are: Median Absolite Deviation, Modified Z-Score, Standard Deviation Method, Tukey's Method, and others [24]. The Statistical methods mentioned are generally used to detect observations that are global or local extremes with respect to a threshold. Rule-Based algorithms rely on established rules or thresholds to identify deviations from expected behavior. In this category, the methods are relatively simple compared to the ones presented later. Their low complexity makes them suitable for real-time analysis due to the very low cost computationally, on the other hand, they can not capture complex patterns. This group of the algorithms is more adequate for univariate data, but it can also be implemented on multivariate data[25] at the cost of increased complexity on defining the rules or thresholds.

2.7 Exploring networks and anomaly detection

Given the topology of the power grids it is important to map the appliances spatially and create networks [26]. One method to define a network is graphs, in order to create the graph the location of appliances is utilized. Given the appliance's postcode it is possible to obtain the latitude and longitude. Using these coordinates, it is possible to pass them from spherical coordinates to cartesian coordinates. Graphs are composed of Vertices (Nodes) which can represent entities or objects, and Edges (Links), which represent connections or relationships between pairs of vertices. In this case the vertices represent areas and its properties and the link between the different vertices represent their connection, as neighbors.

Anomaly detection, also referred as outlier detection is the identification of events or observations which deviate significantly from the majority of the data [27]. Initially anomalies were investigated, in order to clean the data, however in many applications the anomalies themselves are of significant interest and represent the most desired observations on the data. For the problem at hand, anomalies are the focus, as they represent the zones of high variation of consumption, which should deviate from the rest of zones if the difference is significant. Anomaly has been used in a range of different fields, one of which is Ecosystem Disturbance Detection [28].

2.8 Graph

Graphs can be utilized to analyze networks, this type of approach is very useful on complex networks.

Graphs are composed of nodes and edges, edges connect different nodes and can represent different attributes. Graphs can be classified into several types, based on its characteristics and structures. Graph-based approaches are suitable to handle vast and intricate networks, making them suitable for analyzing systems with an extensive amount of interconnected components.

On the other hand, analyzing large and dynamic graphs in real time, presents challenges for real-time processing, as it can be computationally intensive, introducing latency on response times. Additionally, in the cases of large-scale networks, it may require substantial computational resources.

To overcome the drawbacks of this method, elaborated algorithms to optimize the processing time and reduce computationally resources.

There are multiple variants to detect hotspots on graphs, each of these consider different characteristics.

- Density of Nodes
  - Considers the number of connections or edges in a specific region of the graph in relation to the total number of possible connections[8], [29].
- Node or Edge Values
  - Nodes with high centrality measures (e.g. degree centrality, betweenness centrality) may indicate focal points within the network and represent potential hotspots. Edges with high weights or values may signify strong connections or interactions between nodes, suggesting areas of interest or activity[30], [31].
- Spatial Analysis
  - Spatial analysis techniques can be employed to detect spatially concentrated areas of activity or influence. Spatial clustering algorithms and spatial correlation can be used to identify spatial hotspots based on the proximity and similarity of nodes[32], [33].

Liu et al.[32] explored how Deep Learning methods for spatial geodemographic classification, where the authors utilized Graph Neural Networks. In their research they studied different ways of constructing spatial graphs, and their impact on classification results. The methods analyzed for spatial graphs construction were the following:

- K-Nearest Neighbors (KNN), involves connecting each point to its k closest point.
- Contiguity, defines neighbors based on spatial adjacency, one case of this is Queen's Contiguity [34].
- Distance threshold, creates edges between points that are within a specified distance of each other.
- Creates a mesh of triangles connecting points without any points inside the circumcircle of any triangle.

Pourbahrami et al.[35] and Bloemheuvel et al.[36] presented different algorithms for spatial graph construction, where the authors presented multiple algorithms and their complexity which is fundamental for the real time component of the Burden Hotspots Detection. Low complexity ones are going to be the focus, in order to reduce latency on results calculations.

2.9 Machine Learning and Deep Learning

Machine Learning and Deep Learning are interconnected as Deep Learning (DL) is a subset of Machine Learning (ML), and the last one is part of the Artificial Intelligence universe [37]. Machine Learning and Deep Learning have garnered significant attention as a focal point of research, due to its ability on pattern and relationships recognition. This feature provides ML and Deep Learning with the capacity to solve problems by learning the problems specific data intricate relationships and patterns. Artificial Intelligence has been on a rise, with an increasing development and studies[20] to push its boundaries further.

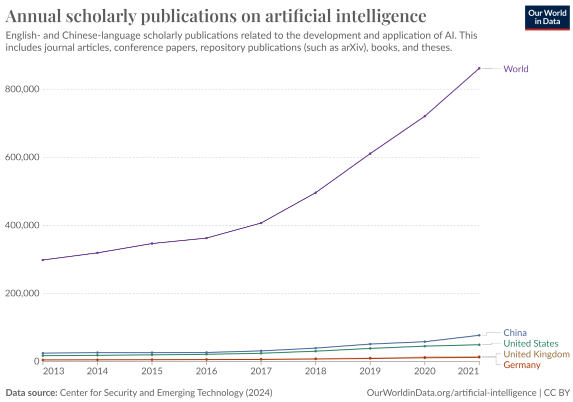


Figure 2.5: Annual scholarly publications on Artificial Intelligence. Image obtained from [38].

Machine Learning and Deep Learning have four different categories namely supervised, semi-supervised, unsupervised and reinforcement learning, each category defines how the model is trained. Each one of them have different requirements from data:

Supervised Learning needs labeled data, in order to learn a mapping pattern from the inputs to outputs. This type of data is very rare in real world datasets

Unsupervised learning does not require labels, the models who are trained with this method, explore the inherent structure of the data to discover hidden patterns and relationships without explicit guidance.

Semi-Supervised Learning, is a hybrid approach from the previous two methods. It is trained on labeled and unlabeled data. This approach affords the advantage of examining patterns and relationships in correlation with the designated label with the addition of exploring hidden patterns.

Reinforcement Learning is based on reward or penalty, where an agent learning makes decisions by interacting with the environment, with the goal of maximizing the cumulative reward. Machine Learning and Deep Learning models can be defined by their objective, generally besides the categories mentioned earlier, the models can also be divided into Classification and Regression. The objective of Classification is to assign input data points into classes, whereas Regression is to predict continuous numerical values.

In the case of DL models, each model architecture has a set of customizable intrinsic parameters which are specific for certain models, with the addition of the neuron weights that can be optimized during the training of the model. There are also hyperparameters, external to the model which depend on the chosen methods for Loss Function, Optimization and Regularization, Batch size, Learning rate and Epochs, these intertwined to optimize the model during the training.

Loss Functions quantifies the difference between the model predictions and the real values, measuring the model performance. Classification commonly uses cross-entropy loss, measuring the difference between the predicted probability and actual class, Regression usually uses Mean Squared Error (MSE) or Mean Absolute Error (MAE) [39], MSE is sensitive to outliers, as they are going to be penalized more since the error is squared, resulting on a higher penalty on outliers. Root Mean Squared Error (RMSE) is also sensitive to outliers, but does not penalize the error as much as MSE. MAE is less sensitive to outliers, as all errors are treated equally.

Optimization refers to the process of adjusting the model neurons weights and biases, to minimize the error of the model, which is calculated by Loss Functions. Some of the optimization algorithms are Gradient Descent (GD), AdaGrad, Adam and others[40]. Adam strong point is its suitability for large datasets with high number of features on non-convex problems. Which is probably the worst scenario possible as it has local minimums to where it could converge.

Regularization is used to prevent overfitting, this occurs when the model learns the training data too well, and fails to generalize to new and unseen data, resulting in a worse performance on testing the model.

Batch size is the number of samples processed before the model weights are updated. This parameter defines how frequently the model updates its weights, it also affects the stability and speed of the training process.

Dropout randomly sets a fraction of the input units to zero during the model training, this encourages the neural network to learn more robust features. Dropout helps reducing overffiting, ensuring the the network does not rely too heavily on specific neurons, resulting on a better generalization.

The learning rate serves as a crucial hyperparameter in optimization algorithms, dictating the magnitude of each step taken during iterations towards minimizing a loss function. This parameter substantially affects the degree to which recently acquired information overrides prior knowledge, thereby symbolizing the speed at which a machine learning model learns. This parameter controls how the model is adapted to the problem. The smaller the learning rate, the smaller the changes on weights after each update and the longer it takes to train.

Epochs is the number of times the model iterates over the entire dataset during the training process. Each epoch consists of one full pass forward and backward through the training dataset, where the model updates its parameters in response to the data to minimize the chosen loss function. Multiple epochs are typically required to sufficiently train a model until it converges to a state where further training does not significantly improve performance.

2.9.1 Machine Learning

Machine Learning methods can also be categorized according to their principle into Clustering, Prediction and Classification methods. One key note is that these types of models can also be located in another category of Machine Learning models depending on its implementation. Furthermore some of the following methods can be used for anomaly detection according to its implementation.

Clustering methods are used to group related data points into a category, based on its features and characteristics without prior knowledge and labels. These methodologies are frequently employed in the realm of data analysis to discern noteworthy patterns or trends within datasets. They can be further categorized into distance-based and density-based methods. Examples of techniques aligned with this principle include:

- Distance Based
  - KNN [41]
  - K-Means [42]
- Density Based
  - Local Outlier Factor (LOF) [43], [44]
  - Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [45]

Prediction methods involve modeling the relationship and patterns between an observation and its historical data, to make informed forecasts or estimates about future outcomes. These types of ML fall under the category of unsupervised learning, where the algorithm learns from labeled examples to predict the target variable of new unseen data.

One of the most simple ones is Autoregressive Moving Average (ARMA), but there are other variants which further develop this model. ARMA is composed of two parts, AutoRegressive models the relationship between current observation and its past values, whereas Moving Average represents the relationship between the current observation and residual errors and from a moving average model applied to its lagged values. These models are adequate for continuous variables. [46], [47]

Classification in ML, is a type of supervised learning, trained to categorize input data into predefined classes or labels. Its main objective is to map the input data into discrete output categories, enabling the algorithm to make predictions on new unseen data. There are two types of classification, with respect to the different number of classes:

- Binary Classification, only has two classes, typically represented as (true, false) and (yes, no), this type of classification is often used on ML anomaly detection.
- Multiclass Classification, refers to tasks with more than two labels.

Some algorithms of this type are:

- Support Vector Machine (SVM) [48]
- Isolation Forest [49], [50]
- Random Forest [51]

2.9.2 Deep Learning

As it was mentioned prior, DL is a subset of ML, focusing on Artificial Neural Networks (ANN), more specifically Deep Neural Networks. This branch extends ML further, making it useful for real-world applications. Deep Learning major advantage lies in its ability to learn intricate patterns and internal representations within data. Furthermore, the gap between these two types of approaches tends to widen as the amount of data increases, with Deep Learning having a better performance the larger the quantity of data.

Due to its higher complexity, DL often exhibits superior performance compared to traditional machine learning models. Their complexity comes at the cost of efficiency, presenting challenges on both the training and inference phases, requiring more computational resources

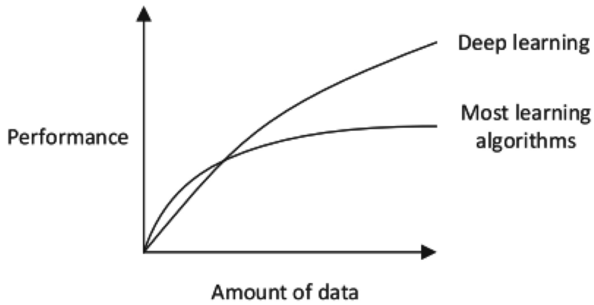


Figure 2.6: Deep Larning and Machine Learning performance by quantity of data [37].

and training time. To overcome the training time challenge, parallelization can be implemented, in order to decrease the training time. The longer inference times, that comes with DL, are critical to scenarios where low latency is important such as real-time applications.

Training a Deep Neural Network presents challenges on multiple aspects, as it was mentioned prior, it takes a significant amount of time to train this type of Neural Network. Furthermore, it is correlated with the number of neurons and layers. Different layers, different properties and collections of parameters to choose from. All of this creates an infinite amount of combinations when building a model, and choosing the most adequate set of parameters has been proven to be difficult, usually made by trial and error. This is, building multiple models with different parameters and evaluating its performance, by selecting the one with the higher accuracy or minimum error. Depth is often related to the number of layers in a neural network, whereas width is with the number of neurons on layers. [52] concluded that widening a Neural Network of different depths improved performance, but there is a bottleneck, once the number of parameters becomes too high, in order to, prevent overfitting, regularization is needed, to count the model learning the training data too well and failing in generalizing, which will result on poor performance on unseen observations.

The number of layers affects generalization, the higher the depth, the better the model can learn the features, at multiple levels of abstraction, detecting intermediate features between the raw data and high level classification. On the other hand, shallow networks are good at memorizing, showing a good performance on cases it was trained with, but the polar opposite on unseen data. [53], [54] Shuffling the data before training improves the machine learning algorithms performance. [55] There are other methods that can improve the model convergence of the training and performance, namely normalization, composed by different methods, [56]–[61], and dropout to prevent overfitting [62]–[66].

Another core aspect of DL models is their interpretability, where it is often referred as Black-Box, this name derives from the difficulty in explaining how a result was obtained. Knowing the data and how the model is trained is therefore very important to get an idea of what the model was trained to inference.

Deep Learning methods can also be categorized based on their principles into Discriminative (Supervised), Generative (Unsupervised) and Hybrid Learning models. Discriminative deep networks can be used for Classification and Regression, its focus lies on modeling in multiple dimensions, and composing extensive knowledge over time. Some examples of Discriminative architectures are:

- Recurrent Neural Networks (RNN) [67]–[69]
- Convolutional Neural Networks (CNN) [70], [71]
- Feedforward Neural Networks (FFNN) [72]
- LSTM [73], [74]

Generative deep learning architectures study the underlying probability distribution of the data, in order to model it, generating new and realistic data samples. Some architectures of this deep learning type are:

- Generative Adversarial Network (GAN)
- Gated Recurrent Unit RNN [75], [76]
- AutoEnconder (AE) [77]–[80]

Hybrid Deep Learning models are the integration of multiple models, Discriminative, Generative and non-deep learning models. The models presented are only a small fraction of all the available methods, these also represent the base methods as these can be further developed into different variants. There have been numerous researches on hotspot detection in a multitude of different fields, namely seismic activities [81], [82], epidemiology [83], pollution [84], renewable energy [85] and crime [86].

Recurrent Neural Networks are commonly used for sequence prediction, including time series forecasting, due to its design, RNN are designed to capture sequential dependencies. The most basic architecture has a hard time learning long-term dependencies in sequential data, there are 2 cases, Vanishing Gradient and Exploding Gradient.

Vanishing Gradient occurs when gradients during backpropragation become extremely small as they are propagated backwards through the layers of a neural network. When the gradients become very small, the network is unable to effectively update its parameters to learn long-range dependencies or capture important patterns in the data. This problem is more notable on networks with activation functions whose gradients are between 0 and 1, like tanh and sigmoid activation functions.

Exploding Gradient is the opposite of the Vanishing Gradient, instead of becoming very small, it becomes extremely large, overflowing, resulting in NaN values. The problems faced by the basic RNN lead to the development of new variants, namely Long Short-Term Memory and Gated Recurrent Unit. LSTM mitigates the vanishing gradient by introducing a gating mechanism. This mechanism is composed by three gates, the input gate, forget gate and output gate, which control the flow of information. Gated Recurrent Unit (GRU) also incorporates gating units to alleviate the vanishing gradient. There are 2 units, that manage the information, the reset gate and update gate.

Feedforward Neural Network

Feedforward neural networks are a type of neural network described by the direction of the flow of information, where connections between the nodes or neurons do not form a cycle. Resulting in a unidirectional flow, from the input layer, through any hidden layers and finally through the output layer[87].

There are different types of layers for the hidden layers, but dense layers are chosen. As it was mentioned previously, dense layers have activation functions, and it is important to define different activation functions for the different layers in order to capture more patterns on the data.

- Input Layer is made of a collection of artificial input neurons. They are responsible for transferring the information from the initial neurons layers to the system for processing, initiating the workflow.
- Hidden Layer i is responsible for transforming inputs into something that they output layer can use. For this they apply weights to the inputs before passing through an Activation Function. This is the process that allows the network to learn non-linear relationships between the input and output data.
- Output Layer is the final layer and responsible for the output, and similar to Hidden Layers also has a Activation Function.

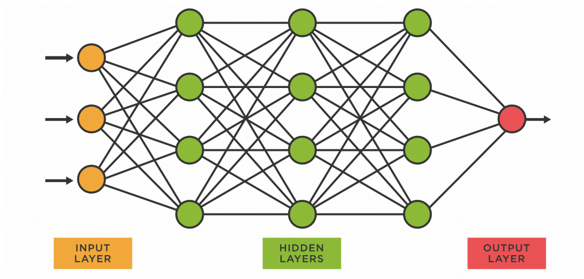


Figure 2.7: Visual Representation of a Feedforward Neural Network

Long Short-Term Memory

Long Short-Term Memory was introduced by Hochreiter and Schmidhuber [88]. It became a popular choice for different sequential tasks, including natural language processing, speech recognition, time series forecasting, etc. A LSTM is composed by memory blocks, named cells, each cell has the following structure 2.8:

Each cell has three main components:

- Cell State, this component is responsible for the long-term information.
- Hidden State, controls the flow of information on short-term.
- Gates, are responsible for multiple aspects of the cell, they utilize two different functions, the sigmoid function is used to amplify/diminish the information fed to it (this function is what prevents the vanishing and exploding gradient), whereas tanh is used to transform the data into a normalized encoding of the data.
  - Forget Gate, determines information to be discarded or remembered. It utilizes the current input and hidden state as parameters, recurring to the sigmoid function to calculate a value between 0 and 1, for each element, elements with a value close to 0 are discarded whereas the ones close to one are remembered.
  - Input gate regulates information to the cell state, both the input and hidden state are encoded by a tanh function, and using a process identical to the input gate,

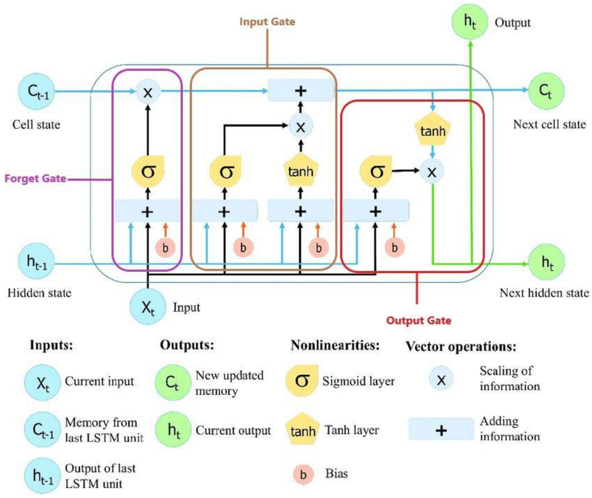


Figure 2.8: Long Short Term Memory Structure. Modified from [89].

the sigmoid function selects the information that is relevant, before adding it to the cell state.

– Output Gate objective is to calculate the next hidden state, the next hidden state which can be used to feed the next LSTM cell or calculate the output recurring to activation functions like softmax or Rectified Linear Unit (ReLU). The hidden state is updated using the long memory, short memory and the input of the cell. Similar to the other gates, the hidden state and input pass through a sigmoid function, but in this case the cell state is also used, after being normalized and encoded by the tanh function, the results of these two functions are then used to calculate the next hidden state.

Although some of the more complex methods are promising, they could not be implemented in the context of Hotspot detection. Supervised Machine Learning and Deep Learning cannot be trained, as labels are not available. Unsupervised Machine and Deep Learning could be implemented, but it would be difficult to measure their performance, furthermore as Machine Learning and Deep Learning are considered a black-box when it comes to understanding their operations on the data and the result obtained.

Statistical methods are a good approach, as samples can be defined by the regions established, when compared to the graphs approach the main difference is the absence of neighbors relationship that can be used for comparison between different appliances. This approach would also require a method to guarantee that the sample size is big enough for meaningful results. The last aspect of anomaly detection using statistical methods is that as Figure 3.1 shows, most of the heat pumps are not used most of the time, this is equivalent to no variation on power consumption which could lead to any variation of power consumption being detected as an anomaly.

As for the rule based methods, thresholds need to be defined, the optimal thresholds can be difficult to calculate. Furthermore, in the cases where Kernel Density Estimation is utilized, irregularly shaped data and discrete data can lead to inaccurate density estimates.

Graph methods are a good approach, as they can encapsulate spatial dependencies through nodes connections, neighbors. Making it easier to define the topology of the grid. The only disadvantage is the latency, as it can be computationally expensive. Graphs approach can also help to individually and spatially differentiate hotspots, as opposed to Rule Based and Statistical Methods which can only label the data as hotspots.

Taking all the methods presented, the most adequate choice for Burden hotspot identification are graphs. On this type of method, several variants were presented, the one that fits the problem at hand, is the one that considers Node or Edge Values. As by the hotspot definition, a hotspot is not an area with high connectivity, or an area with strong spatial correlation, it is an area or a point with high variation of power consumption. Spatial dimension is achieved through a comparison between a node and its neighbors.

On the Graph Node or Edge Values two research papers were presented, Chen et al. [30] presented a method which used DBSCAN to cluster densely distributed data and remove sparsely distributed data which is considered as noise. In the context of the burden hotspots identification problem, sparse points cannot be regarded as noise, as every point is important to the power grid as long it has a variation of power consumption, in order to control the flow of power through the grid. The researchers also used K-means, which is good to separate high density areas into smaller clusters, reduce the graphs complexity, the latency and computational resources utilized. Tabarej et al. [31] algorithm is more aligned with the definition of Burden Hotspot.

2.10 Databases

There have been multiple studies on evaluating databases performance, highlighting different databases, data, metrics and operations performed by Database (DB).

- InfluxDB [90]–[95]
- TimescaleDB [90]–[92]
- ClickHouse [90], [91], [93]
- MonetDB [93]
- extremeDB [93]
- MongoDB [94]

The studies concluded some of the DBMS specializations, which were the following:

- InfluxDB reflects an highly scalable database, with his constant ingestion rate as the database increases, constant query latencies as the insertion rates increase and a solid performance on queries.
- TimescaleDB strong points are data insertion and aggregation and scalability, being a solid DB on all aspects but not excelling in any.
- ClickHouse has short query latencies and great scalability, designed for Large Input, Output and Loading data, optimized to work with datasets.
- MonetDB specializes in insertion.
- extremeDB queries are optimized to Window Operations, Aggregation queries and filtering.

2.11 Streaming

Data Streaming provides low latency, with latency ranging from milliseconds to seconds, which is crucial to analyze or process data in real time, this enables a faster response time to changes. There are different frameworks dedicated to data streaming Puentes et al. [96] did a research on frameworks for data streaming, for two scenarios, data ingestion and data processing, in this research, the focus lies on data ingestion. For data ingestion three frameworks were considered, Kafka, Flume and Nifi, of these three Kafka was considered best, due to some of its features, namely topics, data replication and dynamic structure.

CHAPTER 3

Data Analysis

3.1 Dataset Overview

Data from 17 heat pumps were collected during a period of time through sensors, each heat pump contains a multivariate time series. For each heat pump, the data was recorded at small intervals, tracking multiple variables, composed by multiple time series, one for each of the different variables measured in time. The interval of each heat pump data is different for all of them, having a different starting and ending point in time. The number of heatpumps are only a small sample of appliances, nevertheless the results obtained can be extended later for a bigger universe of appliances.

3.2 Dataset Description

The nature of the data can be described as Irregularly Multivariate Time Series. Each of the appliances individual Multivariate Time Series, are in a csv file. In addition to the multivariate time series of appliances there is also a file which contains static data about appliances and can be cross referenced through the unique id of the appliance, sysid, this file was utilized to extract the maximum power of the appliances.

The data provided has the following measurements that are relative to heat pumps:

- ts: Time Stamp
- ActPow: Current Power Level of the heat generation (%), with values the range [0,100]
- CUHP_HMI_IDUtype: Indoor Product Type
- CUHP_HMI_ODUtype: Outdoor Product Type
- ChActive: Flag reflecting if the heat generator is active to provide space heating (string)
- DHW_E21_T3_START_TEMP: Threshold, domestic hot water start temperature (numeric)
- DHW_E21_T7_STOP_TEMP: Threshold, domestic hot water stop temperature (numeric)
- HwActive: Flag, reflecting if the heat generator is active to provide space heating (string)

- HwTAct: Domestic hot water temperature (ºC)
- HwTSet: Temperature defined to heat storage tank water (ºC)
- HwTStor: Storage tank water temperature (ºC)
- OutTemp: Outdoor air temperature (ºC)
- PrimTSet: Current flow temperature setpoint of heat generator (ºC)
- maximum_power: Maximum power for the appliance (kW)

These metrics are useful to understand electrical consumption by heat pumps, in different scenarios, namely Space Heating, Water Heating, Cooling and when they are not being used. Each of these measurements help differentiate the different scenarios, and for each of them understand how the values are interconnected to the power consumption.

	ts	ActPow	CUHP_HMI_IDUtype	CUHP_HMI_ODUtype	ChActive	DHW_E21_T3_START_TEMP	DHW_E21_T7_STOP_TEMP	HP_EnergyOutCH	HP_EnergyOutDHW	HP_EnergyOutTotal	HwActive	HwTAct	HwTSet	HwTStor	OutTemp	PrimTSet	SysPrimT
0	2022-10-10 12:42:17.142000+00:00	48.0	NaN	NaN	No	NaN	NaN	NaN	NaN	NaN	Yes	49.4	60.0	45.0	NaN	22.0	23.7
1	2022-10-10 12:42:29.164000+00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	49.4	60.0	45.0	16.7	NaN	NaN
2	2022-10-10 12:42:41.132000+00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	49.4	60.0	45.1	NaN	NaN	NaN
3	2022-10-10 12:42:53.133000+00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	49.4	60.0	45.1	NaN	NaN	NaN
4	2022-10-10 12:43:05.132000+00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	49.4	60.0	45.2	NaN	NaN	NaN

Figure 3.1: Sample of raw data of an heat pump

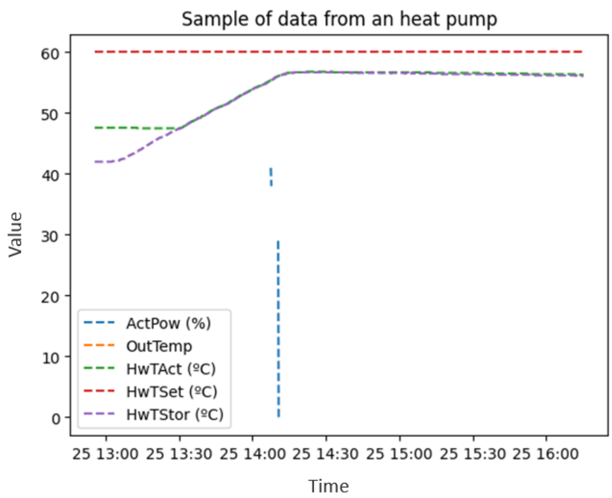


Figure 3.2: Sample of features of raw multivariate time series

In figure 3.2 it is possible to see that the features time series are not continuous, due to a lot of missing values on the multivariate time series.

It is expected for the space heating consumption to have seasonality, due to seasons, in Winter is more cold when compared to the other seasons, and should be where most of the space heating power consumption is located. Hernández et al. conducted a research [97] on the relationship between Weather Variables and Electric Power Demand, the authors concluded that Temperature, Global Solar radiation are negatively correlated with the electric power consumption, on the other hand humidity is positively correlated. Furthermore the authors also analyzed the seasonal correlation of the variables and verified that temperature has higher correlation in Autumn and Spring, while in Summer and Winter it has low correlation.

It is also expected for the appliance cooling consumption to have the opposite behavior, this is, to be used during summer, and also contain a seasonality.

3.3 Data Preprocessing

The raw data needed to be cleaned and prepared for analysis, first the string values of ChActive and HwActive were remapped to binary, 1 representing that is active and 0 the opposite. One characteristic of the data is that only values that changed are registered, this resulted in a lot of missing values. To fill these values, a forward fill was utilized, replacing missing values with the last known value that occurred before it in the sequence. With forward fill, when you encounter a missing value in a dataset, you replace it with the last known value that occurred before it in the sequence. In other words, you propagate the last known value forward until you encounter a non-missing value. This approach was taken, as events are sent when a value changes, this results in a lot of missing values, but the value is not missing, it is the last value registered.

The value of maximum_power for some of the appliances was also missing, the value is extracted from the feature CUHP_HMI_ODUtype. The data had some outliers in ActPow, with values superior to 100%, where they were corrected to the maximum capacity 100%. The time stamp was also converted to the local time. The data of all heat pumps was filtered, removing any data past 27/12/2022, since only a few of the heat pumps contained data prior to this day and could affect the Data Analysis. With the additional removal of heat pumps whose consumption was always zero. Additionally there was the removal of features, namely, ts, CUHP_HMI_IDUtype and CUHP_HMI_ODUtype. AFinally another feature was created which identifies the heat pump by an ID and all the data was merged into one dataset, containing a total of 6 139 013 observations.

	ActPow	ChActive	DHW_E21_T3_START_TEMP	DHW_E21_T7_STOP_TEMP	HwActive	HwTAct	HwTSet	HwTStor	OutTemp	PrimTSet	max_pow	sysid	local_time	time_diff
1094476	0.0	0.0	42.0	60.0	0.0	47.1	60.0	43.4	18.6	0.0	9.0	16NSJNnjLK4MndjZYaKYGKEV	2023-07-24 02:08:33.822	11.996
1094477	0.0	0.0	42.0	60.0	0.0	47.1	60.0	43.4	18.6	0.0	9.0	16NSJNnjLK4MndjZYaKYGKEV	2023-07-24 02:08:45.818	11.991
1094478	0.0	0.0	42.0	60.0	0.0	47.1	60.0	43.4	18.6	0.0	9.0	16NSJNnjLK4MndjZYaKYGKEV	2023-07-24 02:08:57.809	12.034
1094479	0.0	0.0	42.0	60.0	0.0	47.1	60.0	43.4	18.6	0.0	9.0	16NSJNnjLK4MndjZYaKYGKEV	2023-07-24 02:09:09.843	11.978
1094480	0.0	0.0	42.0	60.0	0.0	47.1	60.0	43.5	18.6	0.0	9.0	16NSJNnjLK4MndjZYaKYGKEV	2023-07-24 02:09:21.821	11.995

Figure 3.3: Sample of raw data of an heat pump

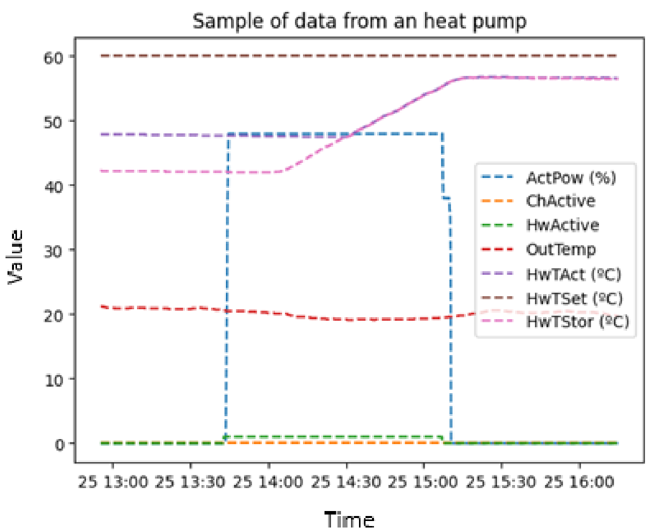


Figure 3.4: Sample of features of post processed multivariate time series

As we can see in the figure 3.4, the multiple time series are continuous in time, as the missing values were dealt with 3.2.

3.4 Data Exploration

In this section, the data will be explored in order to draw conclusions from the dataset provided and formulate hypotheses.

With this in mind, in this section, the following statements will be addressed.

- Create new features, through the already existing data
- Identifying Patterns
- Understand the relationship of different variables

Since the data is not evenly spaced through time, a feature is calculated, time_diff, which contains the difference in time of the previous timestamp to the current one. Additionally the day and month were extracted from the timestamps, since it could contain relevant information for the appliance usage. The data from the different appliances has been merged into a single dataset. Data Analysis, has different purposes: Through Data Analysis, it is possible to get a better understanding of the data presented and draw conclusions about the different relationships presented on all features of the dataset, before building a model. It also plays a crucial role in the identification of the most important features to make predictions. Initially the data will be analyzed in a more general perspective, before taking a more detailed approach. It was noticed that there was energy consumption, when it was neither heating the house or the water, although there is not a flag for it, the appliance can be used for cooling. It is easy to identify most of the cases where the appliance is used for cooling, since if it's using power, but it's not heating anything, its cooling the house. But there are cases where it could be heating water and cooling the house, in this case it is impossible to discern if it's doing both operations or only heating water, given the information provided. To understand the appliance usage type, the percentage of observations for each different type was calculated and are present in the figure 3.5.

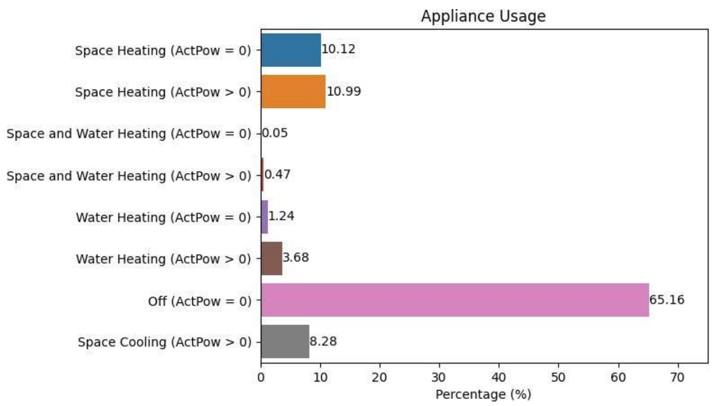
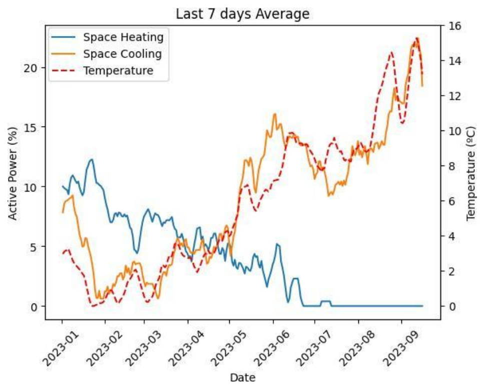


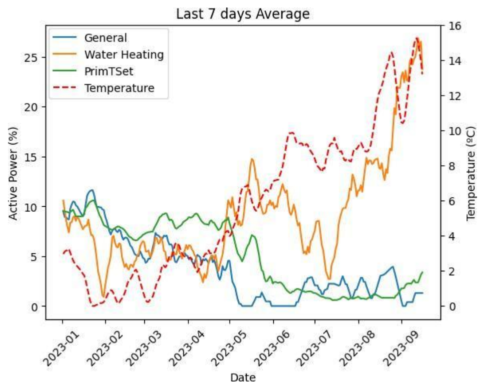
Figure 3.5: Observations count by their type of usage

Examining the data presented on the figure 3.5, it is evident that a predominant duration is characterized by the inactivity of appliances. Furthermore only 23.42% of the time, it is consuming power. This observation suggests a potential enhancement in the scalability of implemented models, allowing them to prioritize more critical consumption scenarios. By focusing these more complex models on instances of elevated consumption. The feature ActPow presents a bias, as 65.16% of observations where ActPow has the value 0, eclipses the number of observations that have a value in the range]0,100], which only amount to 34.84% even though the value interval is larger. If the data fed to train models is not carefully balanced, it would lead to the model accurately predicting when the consumption is 0, but failing to predict values superior to 0 which are the most important cases for the problem at hand.

By filtering out the different appliances ActPow time series observations by their usage, the average was calculated though a 7 day rolling window to verify how if there is seasonality according to their usage. The results are presented in the figure 3.6



(a) ActPow Time series for Space Heating and Space Cooling Usage



(b) ActPow Time series for Water Heating Usage

Figure 3.6: Aggregated ActPow Time Series corresponding to their usage

As it was expected, the figure 3.6, demonstrates that there is seasonality in this dataset on the cases of Space Heating, Cooling and Temperature and they share a relationship. As outside temperature increases, the energy consumption for space heating decreases, and the energy consumption for space cooling increases. As for water heating it was expected that seasons would not have a big impact on the power consumption as it is shown on the figure 3.6b, as the consumption for water heating should have remained stable, as you need hot water trough the whole year. On contrary to what was presumed, the consumption for water heating increases as temperature increase but the temperature could be the not the cause of it, but the small number of appliances present in the dataset.

Pearson Correlation coefficient [98], measures the strength and direction of a linear relationship between two continuous variables. Using Person Correlation Coefficient, the correlation between the different features are presented in the figure 3.7.

Correlation can be used to remove the dimensionality of the data, by removing redundant features, accelerating the training process and simplifying the model developed. On the other hand it is important to keep features that are correlated with what the model objective is (forecasting, classification).

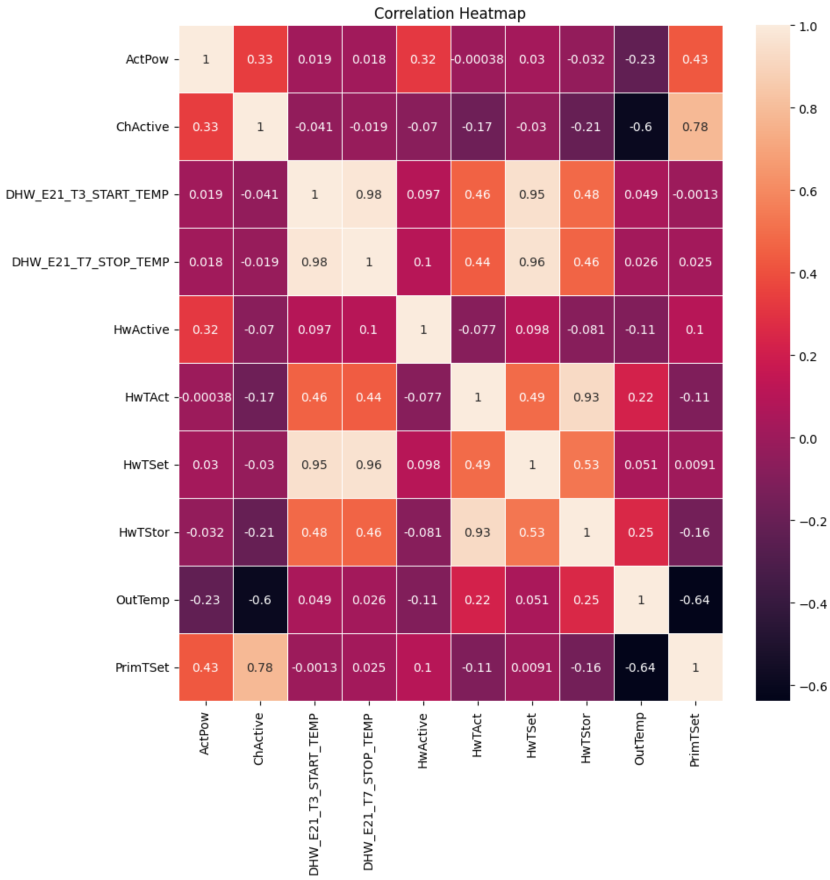


Figure 3.7: Features correlations

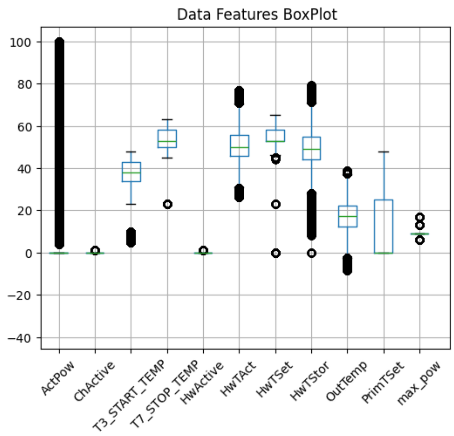
The condition $|\rho_{(feature1,feature2)}| > 0.9$, defines a pair of features with a strong correlation, where $\rho_{f(eature1,feature2)}$, is the Pearson Correlation Coefficient between feature 1 and feature 2. The value of 0.9 was chosen in order to restrict the dimensionality reduction and not remove a lot of features based on correlation, has the dataset is unbalanced.

Considering this threshold, by analyzing the figure 3.7, there are 2 groups who share a strong correlation, (DHW_E21_T3_START_TEMP, DHW_E21_T7_STOP_TEMP, HwTSet) and (HwTAct ,HwTStor). For this reason, of each group presented, only one will be used to train the model, in order to simplify it, resulting in a faster training. Although ActPow does not have a strong correlation with any features, there are some who stand out, namely ChActive, HwActive, OutTemp and PrimTSet.

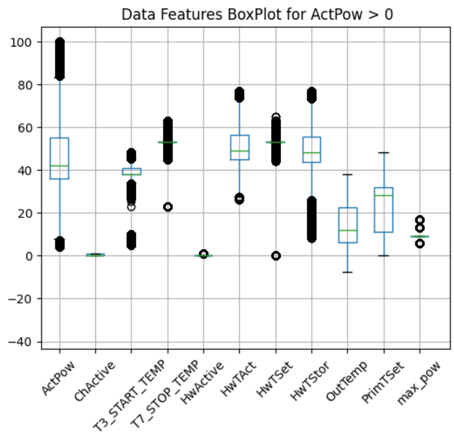
Analyzing the boxplots 3.9 for both cases, where all the data is considered and when appliance is being used two of the features who have the biggest difference are ActPow, PrimTSet and OutTemp. The other features mean and percentile also change although not as notable as the features mentioned prior.

The difference between ranges of features in the boxplots 3.9 presented, shows that before training a model the data needs to be filtered otherwise the model would have a bias towards 0.

One feature was created, time_diff, representing the time interval between events, this feature was calculated using the following formula.



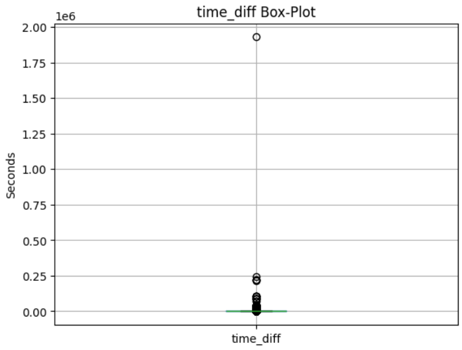
(a) Box plot of features



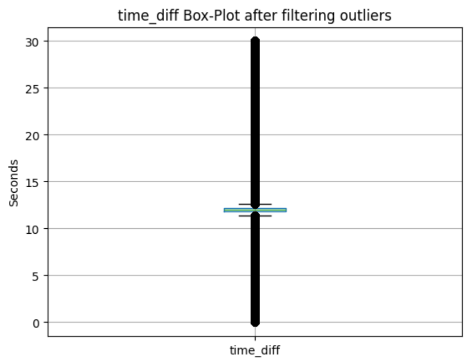
(b) Box plot of features when appliance has consumption

Figure 3.8: Aggregated ActPow Time Series corresponding to their usage

$time\_diff_t = ts_t - ts_{t-1}$, where ts are the timestamps.
Furthermore, the boxplot of the feature time_diff was calculated:



(a) Box plot of feature time_diff



(b) Box plot of feature time_diff after filtering

Figure 3.9: Feature time_diff boxplot

In the figure 3.9a, the outliers do not allow a good visualization of the data due to the scale, in order to extract more information, the value 30 is defined as a threshold to filter the data since as appliance gateways are expected to send events every 12 seconds, and select all values inferior to this threshold, the highest values were removed. After filtering there were a total of 5492453, 10% of observations were removed, the boxplot 3.9b represents the data after the threshold. In this box plot we can see most of the events have a time difference of around 12 seconds which is expected.

The histogram of the feature time_diff was also calculated, represented in the figure 3.10 The histogram also confirm that most events take around 12 second between each other.

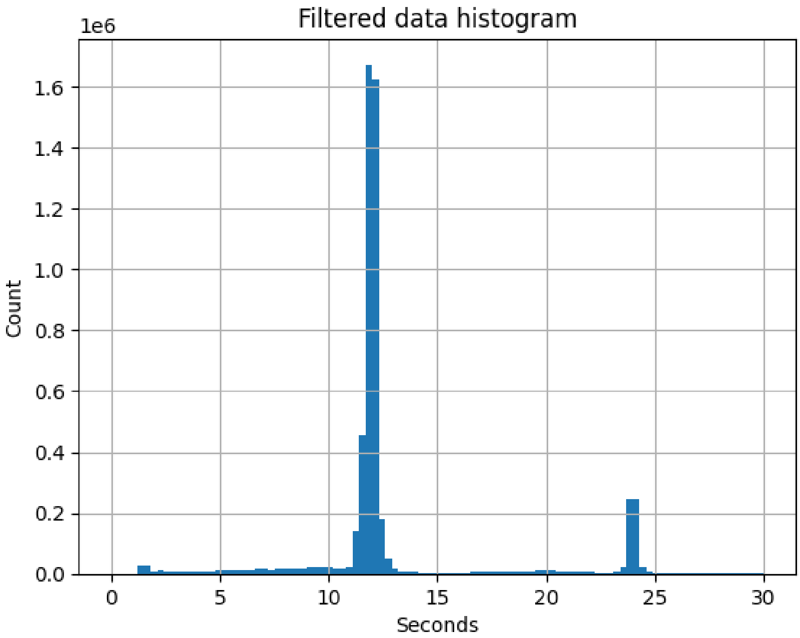


Figure 3.10: Histogram of feature time_diff

3.5 Feature Importance

Feature importance, as the name implicitly says, classifies how a feature impacts the model objective, in this case it is the prediction of ActPow. Although there are several techniques for feature importance [99], according to the different types of data, the authors reccomned different methods.

For the data utilized on this project, as it was previously mentioned on Data Analysis, some of the variables are correlated. Given that the correlated variables were already identified, and it is important to analyze the variables with their correlations, furthermore, there are 13 variables which might not be high dimensional input variables it is a considerable amount, lastly as the data has different types of data, both numerical and categorical, random forests is the method that is more adequate.

A decision tree itself [100]–[102], can be seen as a foundational component of a random forest. A decision tree can be viewed as a basic building block within the more complex structure of a random forest, where each tree contributes to the overall prediction through its unique splits and decisions. Decision trees were utilized to filter relevant parameters for the ActPow. A model was developed and trained with all the features, with the exception of respective ActPow which was the feature the model was going to predict. In other words, the model using all the features with the exception of ActPow, predicted the respective ActPow.

The model obtained a MAE of 0.004 and a MSE of 0.072, which are good results, as ActPow range is [0,100], the error of the model was very low, in comparison to the range of the model.

Analyzing the figure 3.11, ChActive is not a relevant feature in relation to ActPow, this could be due that the OutTemp and PrimTSet are enough to determine when the appliance

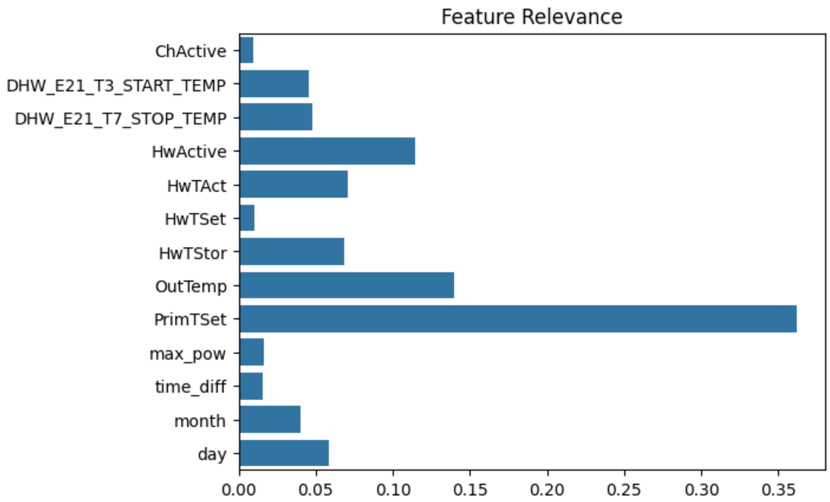


Figure 3.11: Decision Tree Feature Importance, considering all the features

is heating. Furthermore max_pow does not have a big impact on the ActPow, time_diff also does not have a big impact, but this can be explained, as the model is predicting current value, if it was forecasting the next value instead of the current one, the results may differ.

Next DHW_E21_T3_START_TEMP, DHW_E21_T7_STOP_TEMP have similar feature relevance, whereas HwTSet has no relevance at all, since the previous two features already capture the relevant information, as all three have a strong correlation. Similar to the previous case, HwTAct and HwTStor share a strong linear relationship, and both have a good feature importance.

Month and day seem to have an impact but not that great compared to the importance some other features have gotten in the figure 3.11, are they are the 6 and 8 feature with highest relevance in the model prediction, in a total of 14. Furthermore month and day might introduce a bias or error, since it's not only the time of the year that matters but the location of the heat pump. Since at the same time of the year, two different appliances can be located in zones with different seasons or different climates. Although this could be circumvented the latitude and altitude as parameters for the model, this would also increase the complexity of the model.

In order to evaluate on a more detailed manner the features importance on the model accuracy, the following approach was taken: First ChActive and max_pow were removed since they were at the bottom of the importance, 14 and 10 respectively of 14 features, since they had almost no relevance and were not correlated with other features. With this model, the results obtained for MAE and MSE were 0.004 and 0.073 respectively. As it was expected, the accuracy is almost the same when compared to the Decision Tree Model that was trained as with all features. So ChActive and max_pow will not be considered for further training of models.

After this, more features were removed for each feature of the set (DHW_E21_T3_START_TEMP, DHW_E21_T7_STOP_TEMP, HwTSet), different models were trained were only one feature of the set was kept since on the correlation analysis it was concluded that they were highly correlated. The results are presented on the table 3.1

Table 3.1: Decision Tree Error Error according to different features removal

Features Removed	(DHW_E21_T3_START_TEMP, DHW_E21_T3_STOP_TEMP)	(DHW_E21_T7_STOP_TEMP, HwTSet)	(DHW_E21_T3_START_TEMP, HwTSet)
MAE	0.005	0.004	0.005
MSE	0.086	0.074	0.084

So for reference the prior model obtained a MAE and MSE of 0.004 and 0.073 respectively. Analyzing the error the Decision Tree Model obtained from the removal of different features presented in the table 3.1, the pair of features that was removed and obtained better performance was HwTSet and DHW_E21_T7_STOP_TEMP. It's possible to see that removing HwTSet and DHW_E21_T7_STOP_TEMP has almost no change on the error as MAE remained the same and MSE increased by 0.001, which represents an percentual increase of 1.37%. So for this reason both HwTSet and DHW_E21_T7_STOP_TEMP are not going to be considered to train further models.

Table 3.2: Decision Tree Error according to feature removal of HwTStor and HwTAct

Feature Removed	HwTStor	HwTAct
MAE	0.012	0.006
MSE	0.202	0.112

We have another set of features that are correlated, HwTStor and HwTAct, in order to see which one of the two is removed, we trained two models where in one HwTStor considered in the data utilized to train the model and the second was the feature HwTAct was not considered. The results are presented in table 3.2. Analyzing the results presented in the table 3.2, it's possible to see that removing HwTAct had lowest impact in the error when compared to HwTStor has the MAE and MSE obtained for HwTAct is inferior to the ones obtained HwTStor. Removing HwTAct from the features utilized to train the Decision Tree Model resulted in a increase of MAE of 50% whereas MSE saw an increase of 51%. So it seems that removing either one of the features HwTStor and HwTAct results in a good loss in performance, for this reason both will be kept, even though they are correlated

Finally both month and day were removed, which yielded the following MAE of 0.005 and MSE of 0.084, which is an increase of 25% and 13% compared to the model . As the loss in model accuracy is significant, these 2 features are going to be considered when training a model.

So after this analysis the features that will be considered for training are: DHW_E21_T3_START_TEMP, HwActive, HwTAct, HwTStor, OutTemp, PrimTSet, time_diff, month and day.

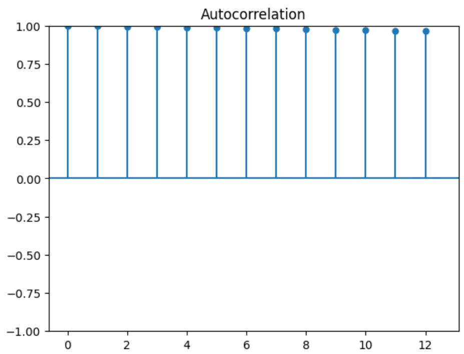
And the features that were removed were: ChActive, max_pow, HwTSet and DHW_E21_T7_STOP_TEMP.

3.6 Lagged Values

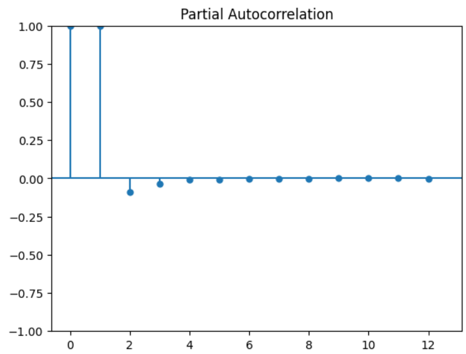
Lagged values [103] in a time series, refer to past observations of the series. Analyzing lagged values helps identifying patterns, as well the persistence over time. These can be measured using autocorrelation and partial autocorrelation.

Autocorrelation [104] analyzes the degree of similarity between a given time series and a lagged version of itself, determining if and how the values in a time series are correlated with the previous values (lagged values). Partial autocorrelation [104] also measured the correlation between a time series, with the key difference of removing the intervening lags when compared to autocorrelation, isolating the direct relationship between the timer series at time t and its values at time $t - k$.

With this in mind, both autocorrelation and partial autocorrelation will be used to detect patterns on variable ActPow, and try to detect the number of observations relevant to forecast the ActPow.



(a) Autocorrelation of ActPow time series



(b) Partial Autocorrelation of ActPow time series

Figure 3.12: ActPow time series correlations

Analyzing figures 3.12 there seems to be an almost perfect autocorrelation with the past observations, partial autocorrelation tells a different story, having a very high correlation to the past event and almost no correlation on [2,3] events prior. This could be due to the fact that in the majority of the events the consumption is 0, resulting in a pattern where the value basically continues the same. Partial correlation suggests knowing the last event is enough, as with this information the last events don't add much more. Nevertheless it might be interesting to test different lagged values and compare the results.

CHAPTER 4

Methodology

4.1 Burden Hotspot Detection

Taking into account the SoA conclusions on methodologies for Burden Hotspot Detection, the method chosen is the one developed by [31]. The following image describes the flow chart 4.1 of the algorithm proposed by the authors.

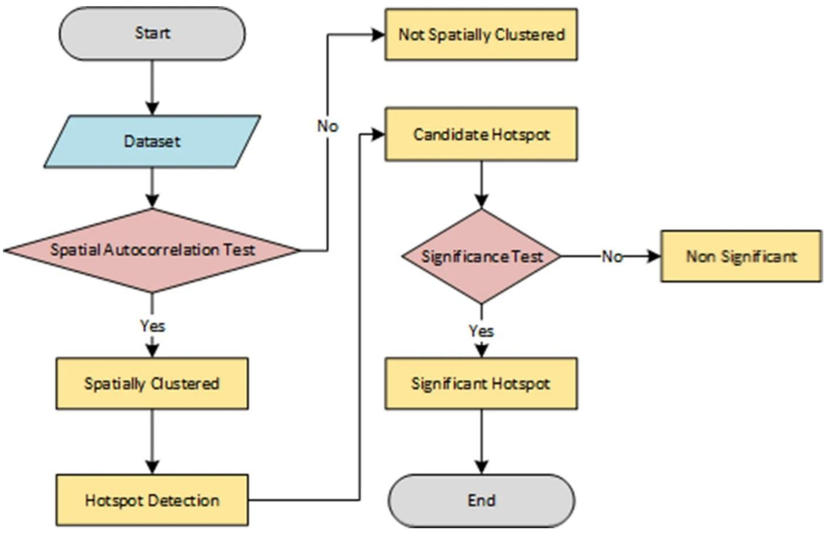


Figure 4.1: Flowchart of Algorithm, from [31]

As it was mentioned prior, spatial dependency is not going to be considered but spatial coordinates will be used to determine neighbors, with this in mind, only the Hotspot Detection part and forward will be considered. The next figure 4.2 shows in more detail the Hotspot detection algorithm.

Looking at the detailed flowchart of the algorithm 4.2, there are some processes of the algorithm that needed to be defined, namely the Determination of Spatial Neighbor, Threshold Selection, Boundary Value Analysis and Significance Test.

The way authors implemented the algorithm, they essentially subsample the Dataset into three samples, Lower Approximation, Boundary Region and Upper Approximation, and

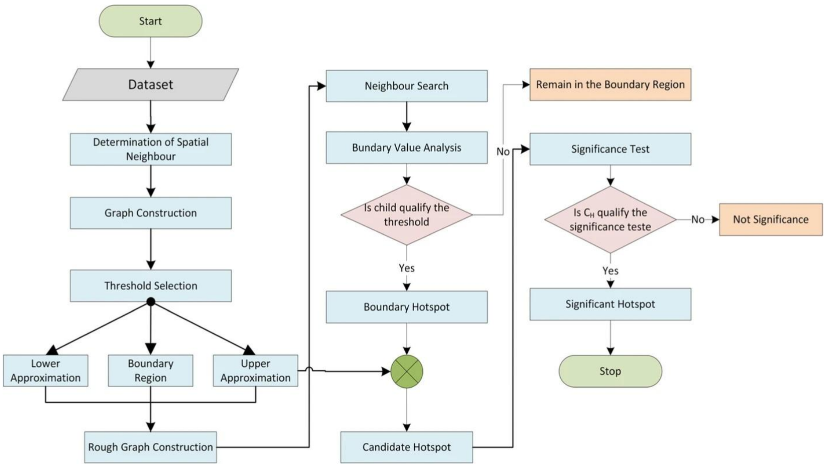


Figure 4.2: Detailed Flowchart of Algorithm, from [31]

evaluates the Lower Approximation Sample (which contains the highest values) in order to detect the hotspots. But the same method could be applied to the Upper Approximation (which contains the lower values) to detect coldspots which is important for the variation of energy consumption as the values can be either negative or positive.

One important aspect is the determination of Spatial Neighbors, [31] in order to build a graph and define the neighbors utilized the Queen's Contiguity [34]. But as it was described on State of Art there are different alternatives as it was presented on SoA. Furthermore there are different methods to calculate the distance where some are approximations, which have lower computational complexity and lower latency.

The two methods presented next, calculate the distance between the nodes.

The manhattan distance, where the distance is computed by the following formula:

$manhattan_d = |x_2 - x_1| + |y_2 - y_1|$, where x_1 and y_1 are spatial coordinates of one of the nods, and x_2, y_2 are the spatial coordinates of the other node.

There is also the Euclidean distance which is more accurate than manhattan distance but it is also slightly more expensive computationally as it needs to compute the squared values and also perform a root computation. Euclidean distance can be computed using the following expression:

$euclidean_d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, where x_1 and y_1 are spatial coordinates of one of the nods, and x_2, y_2 are the spatial coordinates of the other node.

In order to choose the method most adequate for the detection of Spatial Neighbors, appliance data was generated, for the purpose of benchmarking the Spatial Neighbors detection methods, only spatial coordinates are relevant. So for a given appliance, latitude and longitude were generated though a normal distribution.

In the figure 4.3, the red line presents a threshold that can be used as a benchmark, as there are other operations besides the graph construction, namely as the query latencies and the operations applied on the graph for hotspot detection and the hotspot detection itself. From the methods presented, Queen's Contiguity surpasses the threshold presented, whereas KKNN with its two variants does not, at least until 100 000 appliances. There is only a slight

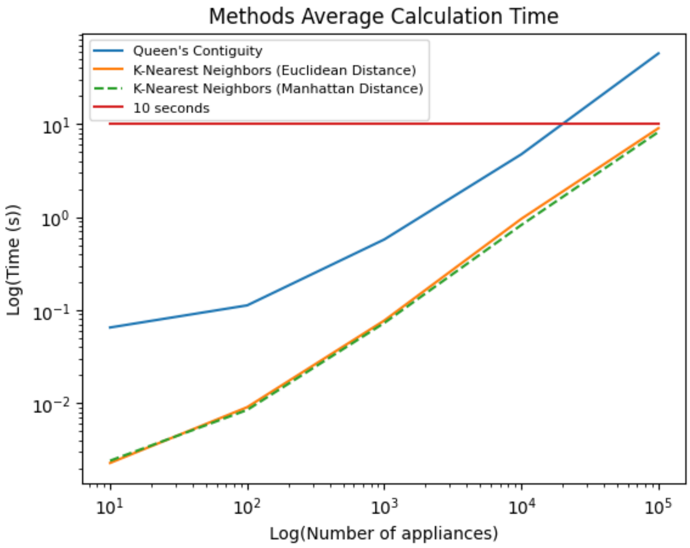


Figure 4.3: Spatial Neighbors Detection Algorithm execution time

difference between the two variants, and the KNN with Euclidean distance will be adopted as it depicts more accurate distances, which result in more accurate definition of neighbors, and the latency is slightly higher.

With the method utilized defined for Determination of Spatial Neighbors, next in the Burden Hotspot Detection Algorithm 4.2 is the threshold selection for the rough graph based construction, the authors Tabarej et al. [31] use the third and second quantile.

Another aspect that comes with this change is threshold selection, on the rough graph construction, if the same thresholds are utilized which is the quantile(0.75) for hotspots, which for coldspots translates to quantile(0.25), implies that 50% of the data are lower or upper regions on the graph construction which is a big percentage of the data when considering that each of the neighbors that are labeled as boundary might change it status to one of the previous regions on the boundary analysis. So the quantiles should be more restricted than the ones of the paper. In order to have the same percentage of upper regions in the graph construction 25%, the quantiles used are 0.125 and 0.875 for coldspots and hotspots respectively. For hotspots detection applying the quantiles on the lower and upper samples, instead of the original sample has the benefits. As the topology influences the sample, if the hotspots are dispersed, implies a higher number of neighbors than densely hotspots. On a sample with pontual hotspots, they are dispersed which leads to a higher number of neighbors being considered for the lower sample, when compared to a sample with dense hotspots, where they are neighbors of each other, as some of the neighbors are already in the lower region of the map. With a larger lower region sample, the more hotspots are detected which is useful when they are dispersed, whereas with a smaller lower region sample where the center of hotspot is detected. In scenarios where there are pontual and dense hotspots are present in the graph, it is more complicated to analyze as the value each node has dictates which are hotspots.

Besides the values defined for the threshold selection on the algorithm, there is also the sample considered for said thresholds.

There are multiple ways to calculate the thresholds considering samples at different levels of geography.

- One option is to calculate the thresholds on a subsample of the data, instead of using the whole available data, it used only the ones that share the same area, example postcodes that share the same city.
- Another option is to also only consider a subsample, but a broader one, by including the appliances of neighbors cities for example. This method also requires calculating the neighbors at the top of spatial hierarchy before, going into a more fine grained view.
- Other option is to consider all appliances from two hierarchies up, so in order to calculate the threshold for for appliances in Berlin for example, all the appliances in the Germany Country would be considered.
- Last option is to consider all the data to determine the threshold.

The first three methods present some problems relative to the sample size, as they are subsamples, the sample size is lower. Furthermore, the sample needs to be at least 30 values [105] as lower approximations to guarantee a sample big enough to compute the Z-Scores (Z_i). $Z_i = \frac{\eta(G(status_i==lower))-mean(G(status==lower))}{std(G(status==lower}$ This formula reflects that the Z Score is only calculated considering a sample of nodes that were labeled as part of lower region.

The last method presented where the all data is considered to define the threshold values is the more adequate as the sample size bigger it is less likely to have scenarios where the sample is small and therefore it is not possible to use the Z Score. Taking this into account the latter method will be employed.

Another important aspect is the Boundary Value Analysis, in this step observations of neighbors of Lower Approximation nodes are added to the Lower Approximation Sample if the sum of these two nodes are superior to a threshold. As it was mentioned previously the algorithm can be adapted to also detect coldspots which is also of interest. So on boundary analysis if a node is considered a neighbor of both a Lower Approximation node and Upper Approximation node, it stays labeled as Boundary region.

For the significance test the authors use the Z-Score, but this method should only be applied when samples follow Gaussian Distributions. Due to the nature of the data, it is not possible to guarantee the normal distributions for all samples.

The normality of the data needs to be tested, as it was described on State of Art there are multiple methods for this purpose, but Shapiro-Wilk [106] is more adequate has Kolmogorov-Smirnov test tests whether a sample came from a given reference probability distribution, which in the problem at hand, the probability distribution is not known. Shapiro-Wilk tests hypotheses, if the test is significant, it's not possible to reject the hypotheses of the data following a normal distribution, if it is not significant, the hypotheses of the data following a normal distribution is rejected, and to deal with this scenario, percentiles are used.

With all this defined, the figure 4.4, shows the algorithm proposed, and implemented. The flowchart depicts the algorithm flow adapted for hotspot and coldspot detection.

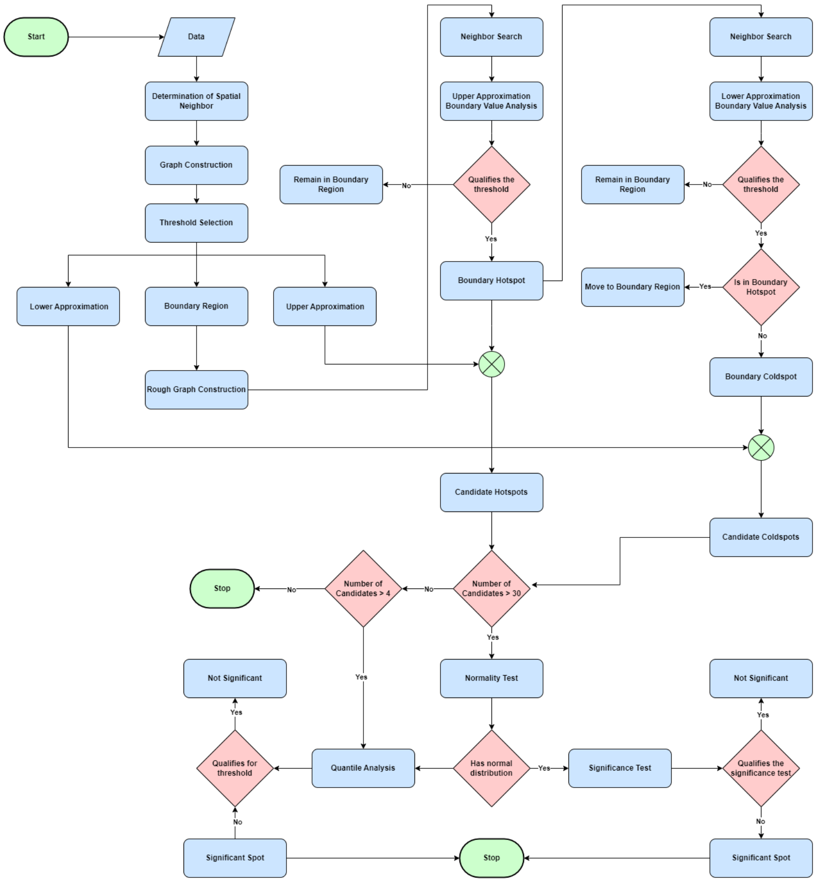


Figure 4.4: Flowchart of proposed algorithm

First the neighbors are identified using the K-Nearest Neighbors, after that graph is constructed.

The threshold selection is made utilizing the quantiles to classify nodes with lower values as lower approximations, and nodes with higher values as upper approximations. The quantiles used were 0.125 and 0.875 respectively.

After this for each neighbor of nodes labeled as upper approximation the boundary analysis is made. The threshold utilized for this case is the maximum value of all nodes. If the sum of the upper approximation node value plus its neighbor is superior to the threshold (maximum value), its status is changed to Boundary Hotspot. The same is done for nodes labeled as lower approximation, in this case the threshold is the minimum and they are changed to Boundary Coldspot.

Since we are adding the node value to its neighbor and for coldspots the value needs to be inferior, if the sample is positive most of the times the condition will not be met, so in order to negate this, the median is subtracted from the sample placing half of the values on the negative side, and the other half on the positive side. With this, lowest values are negative and the sum of two negatives will be lower than the quantile established.

If a neighboring node of a lower approximation node respects the condition of being inferior to minimum, and the same node is a Boundary hotspot, it is removed from the boundary hotspot and labeled as Boundary region. Otherwise it is labeled as boundary coldspot.

Next, the boundary hotspots and upper approximation are merged into a sample, candidate hotspots. Boundary coldspots and lower approximation are also merged into a sample, candidate coldspots. Following this, if candidate hotspot sample has more than 30 values, the normality test is calculated, if the p-value of the test is significant (significance level of 95%). The right-tailed Z-Score test is utilized to detect significant hotspots at a significance level of 95%.

If it is not significant or the sample has fewer observations than 30 and more than 4, a quantile of 0.95 of the candidate hotspot sample is used to detect hotspots.

After this the candidate coldspots samples analyzed similarly to the candidate coldspots. If the sample is significant for the normality test, a left-tailed Z-Score test is utilized. If the sample is not significant for the normality test a quantile of 0.05 is used.

The burden hotspots algorithm can be applied to the lowest levels of the spatial hierarchy, such as postcodes, and subsequently aggregated to higher levels. This process can be repeated, applying the algorithm progressively at each higher level of spatial aggregation.

4.2 Prediction of Active Power

The dataset utilized for the Prediction of Active power is the one mentioned on Data Analysis, more specifically the one after pre processement was done 3.3, which data from 17 heatpumps, 6 139 013 observations. This dataset contains information about appliances parameters, the ones presented in the data analysis, one of which is the ActPow which are the values to be predicted. The data is already cleaned, as missing values were dealt with.

4.2.1 Preprocessing the data

During the Data Analysis, it was concluded that the dataset was unbalanced 3.8a. For this reason is it important to balance the values of the data that is going to be used to train the models.

4.2.2 Unbalanced Dataset

Following the preprocessing of the data on the Data Analysis, there are still some additional steps to be taken before training a model. The dataset has uneven distributions for the ActPow which is the variable the model wants to predict. By separating the dataset into different intervals and counting the quantity of observations on each interval the following values were obtained.

Taking into account the values for each interval, 76.45% of the observations have no consumption. In order to achieve better generalizations, it is important to even the intervals, this can be achieved through two methods, undersampling and oversampling. Prediction algorithms trained on imbalanced data often exhibit poor predictive performance. These

Table 4.1: AcTPow Observations Count by Interval

Range	Observations Count
0	4 662 807
]0,10]	1 927
]10,20]	10 276
]20,30]	270 799
]30,40]	387 647
]40,50]	327 403
]50,60]	198 681
]60,70]	89 216
]70,80]	54 763
[80,90]	34 763
]90,100]	63 048

models tend to have a bias towards the majority class, thereby neglecting minority examples that are crucial for many applications.

These two methods are the opposite of each other. Oversampling offers a technique to rebalance classes prior to model training. By duplicating data points from the minority class, oversampling equalizes the distribution, ensuring that algorithms do not overlook important but infrequent classes. This is adequate when the data is considered small. On the other hand, undersampling techniques involve removing instances from the majority class in the training dataset to achieve a more balanced class distribution. The latter is suitable for a dataset with large amounts of observations.

The method used to even the data was undersampling, 35 000 observations were chosen of each interval. The only classes that do not fulfill the previous quantity of observations are the intervals]0, 10] and [10, 20], which have lower instances, for these two cases all observations were considered. Although they have lower observations it is reasonable as they are the interval where the consumption is at the lowest, and the focus of this study is the hotspots, it is much more important to have high accuracy on the upper values of consumption. This resulted in a dataset with a total of 351 690 observations. This means that the training dataset will have 281 351 observations, and the validation and testing dataset will have 35 169 observations each. The observations training, validation and testing datasets were chosen at random.

4.2.3 Data Split

In machine learning and deep learning, not all data is utilized to train a model. It is also important to measure the model performance. For this reason the data is splitted into Training Dataset, Validation Dataset and Test Dataset.

The training dataset is the dataset used to train the model, it is also the dataset that takes the majority of the data, since the larger the quantity of data there is available to train the model, the better the model learns.

Validation Dataset is a sample of data utilized to evaluate the model performance while tuning its hyperparameters.

The test dataset is a sample of data utilized to provide an unbiased evaluation of a final model fit on the training dataset.

In this dataset the model evaluates unseen data which results in a more accurate performance. Due to the large number of observations the data was split into training, validation and test, with the percentages 80%, 10%, 10% respectively. Heat pumps can be used for space and water heating, but space heating is mostly used in winter, for this reason the time series needed to be splitted at random, otherwise the training data could have all the winter season or none of it, resulting in inaccurate results in either training or validation.

4.3 Models

As it was described in State of Art, several Deep Learning Architectures were presented, of those two stand out for the forecast of ActPow, which are LSTM and RNN as one of the benefits of using them is processing long sequences, which in this case is the multivariate time series.

LSTM offers a better control of information due to its gates, even if the sequences are short, handling the information flow better compared to RNN [107]. On the other hand, LSTM has a higher computational cost due to its higher number of parameters compared to RNN, and one of the components Long Term Memory has no benefit. Both models have their pros and cons but LSTM will be used as it handles information better. For a model with lower computational cost, FFNN will be used which are also described in the State of Art. These two architectures will provide a choice of sacrificing models error performance for scalability or the other way around.

4.3.1 Long Short-Term Memory

Long Short-Term Memory comes a set of procedures and configurations that need to be established. The LSTM model architecture needs to be defined, the architecture affects the model performance and it is important to evaluate different architectures as they can obtain different results. Hyperparameters are essential to prevent some scenarios, namely overfitting and underfitting. In order to find the best parameters the models need to be trained and the results need to be evaluated, though the results do not reflect the final model, as the models will not converge completely. The final model will be chosen after choosing the best hyperparameters.

Overffiting occurs when a model excels on the training data, but poorly on the testing data.

Underfitting occurs when the model has poor performance on both training and testing data.

Model Architecture and Design

The list of hyperparameters defined are:

- Activation Layers
- Recurrent Activation Layers

- Dense Activation Layer
- Number of neurons
- Number of layers
- Number of steps

Activation layers and Recurrent Activation layers represent the functions Tanh and Sigmoid on 2.8 respectively. The activation layer defined was Tanh whereas the recurrent activation layer was Hard Sigmoid.

Dense activation layer, also known as a fully connected layer, is a layer in which each neuron is connected to every neuron in the previous layer. This is the last layer of the model, and is connected to the last layer of the LSTM model, representing the output layer. The activation function chosen for the layer is ReLU, due to its properties of producing an output superior or equal to 0, which goes in line with the objective of predicting the Active Power of a heat pump which is between 0 and 100. Additionally Exponential Linear Unit (ELU) activation function will also be considered, it is similar to ReLU, but it can produce negative values.

The number of neurons refers to the number of hidden units within each layer. Neurons are responsible for learning features from the input sequence. The more neurons, the more complex patterns the model can learn. The number of neurons was [32, 64].

Similar to the number of neurons, the more layers a model has, the more complex patterns the model is able to learn. Each layer captures different levels of abstraction, starting with basic features on the first layers and complex features on higher layers. Initially the number of layers defined was [2,3], but after the analysis of the scalability of solution 4.6, it was reconsidered and changed to [1,2], where the models with two LSTM layers, have the second layer with half of the neurons in order to reduce the complexity of the model and improve performance.

Number of steps represents the sequence length of the input data, it determines how many time points the model looks at when making a prediction. Since we want to predict in real time, it is important to not have a large number of time steps, as it increases the computational time for training and prediction. Defining a baseline of two minutes for the interval of information the model will use, with the interval between events being expected as 12 seconds and it's also the interval most events 3.10, it is equivalent to 10 time steps. This amount of time steps will ensure that the majority of the time, two minutes will be taken into account. A model with 3 time steps, is also going to be trained, based on the results of Partial Autocorrelation on 3.6, where the previous event has a very strong correlation, and the events 2 and 3 a very low correlation.

In the image presented 4.5, the blue lines represent the regular dropout whereas the red arrow represents the recurrent dropout.

In this figure 4.6, it is possible to see that on the input layer, passes an array with dimensions [3,10], representing [Number of past observations, Number of features], after that the values are normalized and passed to the LSTM layer. In the case of the 1 LSTM layer, the output is then passed to the final layer where it outputs 1 value and it is also where the

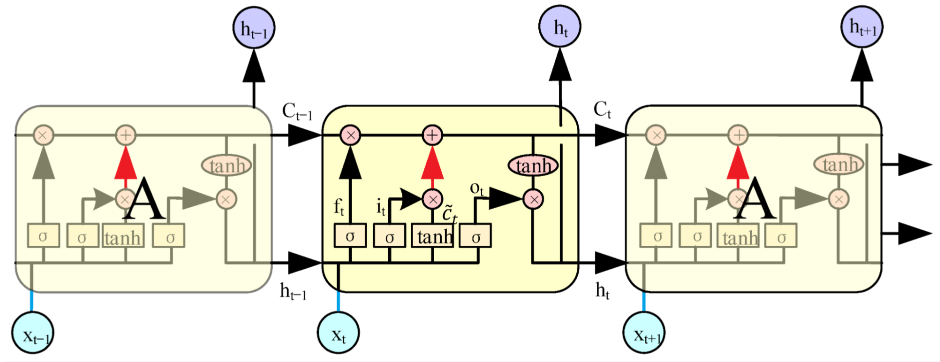
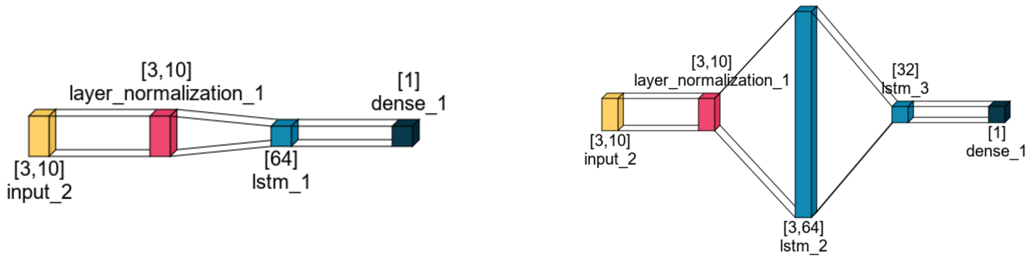


Figure 4.5: Diagram of the LSTM network, modified from [108]



(a) Model with 1 LSTM layer **(b)** Model with 2 LSTM layers

Figure 4.6: Visual Representation of the LSTM model layers

activation function ELU or ReLU is. On the case with the second layers the second layer receives the output array the first LSTM layer which has a higher dimension that the initial output, it then reduces the amount of information into 32 values which are passed to the final layer.

4.3.2 Feedforward Neural Network

There are different types of layers for the hidden layers, but dense layers are chosen. As it was mentioned previously, dense layers have activation functions, and it is important to define different activation functions for the different layers in order to capture more patterns on the data. In order to save time training this model, day and month were removed

- For the hidden layers, dense layers will be utilized and the following activation functions will be used, Hard Sigmoid and Tanh.
- For the output layer, the activation functions defined were the same as the ones used on LSTM models, ReLU and ELU.
- The number of neurons defined are also equal to the LSTM model with [32, 64]
- One layer will be used.
- The number of steps utilized is only 1 as this model cannot capture temporal dependencies

4.4 Training Hyperparameters

As described in State of Art, in Deep Learning models there are a set of parameters to define for training a model, namely, Batch Size, Dropout, Optimizer, Loss Functions, Learning

Rate, Epochs. The following values were used for training:

- Batch Size: 64
- Dropout: 20%, in the case of LSTM there is also recurrent dropout which is also defined as 20%.
- Optimizer: Adam
- Loss Function: MSE
- Learning Rate: 0.01
- Epochs: 500

4.5 Implementation

4.5.1 Model Training

The training of the model was done with python, and the package TensorFlow[109], tensorflow allows to set callbacks on model training.

Callbacks perform actions at various stages of training (e.g. at the start or end of an epoch, before or after a single batch, etc). Several callbacks were defined in order to monitor the models performance and adjust some of the parameters based on it.

The first one is ReduceLROnPlateau, callback adjusts the learning rate. Reducing the learning rate after a certain number of epochs has passed and the loss function of training did not decrease. It multiplies the learning rate by a factor (factor is inferior to 1) up to a limit that can be defined, if the limite is reached traning stops, helping the model converge.

The factor was defined as 0.5, the number of epochs without a decrease in the loss function was defined as 10 and lastly the limit established for the learning rate was 0.0005, in order to not extend the training time for minor improvements.

The second callback is EarlyStopping, this callback monitors the validation loss, stopping training after a certain number of epochs have been completed without any improvement on models performance on the validation dataset. This callback prevents the model from overfitting, and when activated restores the weights models to the previous model with the best performance on the validation loss. The number of epochs needed for this callback to be called is 30.

4.5.2 Pipeline

A pipeline is the partition of complex problems into smaller ones, making it easier to manage, monitor and scale.This results in a sequence of processing elements (processes, threads, coroutines, functions, etc) that execute tasks or instructions, organized so that the output of each element is the input of the next.

With the purpose of real time detection, the pipeline needs to have a few core components to deal with the large quantities of data. One of those aspects is the database, this should be efficient, achieving high performance on data insertion and low query latency, furthermore it should be scalable. In the case of this dissertation, the focus likes on a database capable of dealing with time series in real time and historical.

Taking into account the studies described in State of Art, InfluxDB proved its strong points and was consistent, representing a reliable database for online workloads and ingestion.

The pipeline should also be able to manage data in real time, for this reason different frameworks at on the table, as it was described in State of Art, Kafka was considered the best for data ingestion.

The data sent by the gateways is raw, its values are in hexadecimal format, and need to be translated to a more legible data format. Once the data is received in raw format, it is translated and sent to a Kafka topic, via a producer. Telegraf establishes the connection between Kafka and the database, by consuming events from it and storing it on InfluxDB. The data is then queried from the database and processed, in order to detect hotspots and predict power consumption. The results are then stored on InfluxDB, and subsequently queried by grafana to display the information to the user.

The version of InfluxDB utilized is the OSS version, more specifically the InfluxDB v2.7.10.

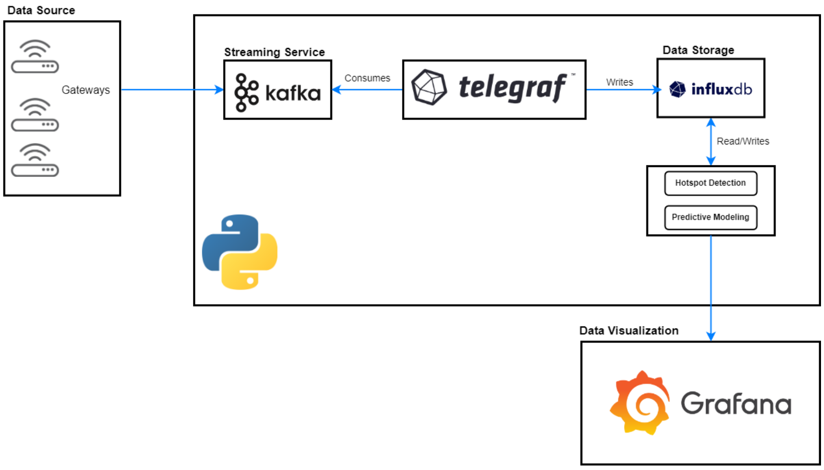


Figure 4.7: Pipeline

The computer utilized has the following specifications:

- CPU: i7-1270P
- RAM: 32GB

These are useful as several benchmarks later on, will be done with these specifications in consideration.

4.6 Models Execution Time

As it was mentioned on 5, the events are expected to be sent from the gateways every 12 seconds. For this reason is important to measure models LSTM models scalability as this type of architecture is more computationally expensive than the FFNN.

The figure 4.8 legend represents the model parameters starting by the dimension of the matrix of inputs of the model, the number of Layers and at last, the number of neurons. The execution time scales linearly, additionally is not just the time the model takes to predict that should be considered, as there's also the latency of queries and preparation of the data

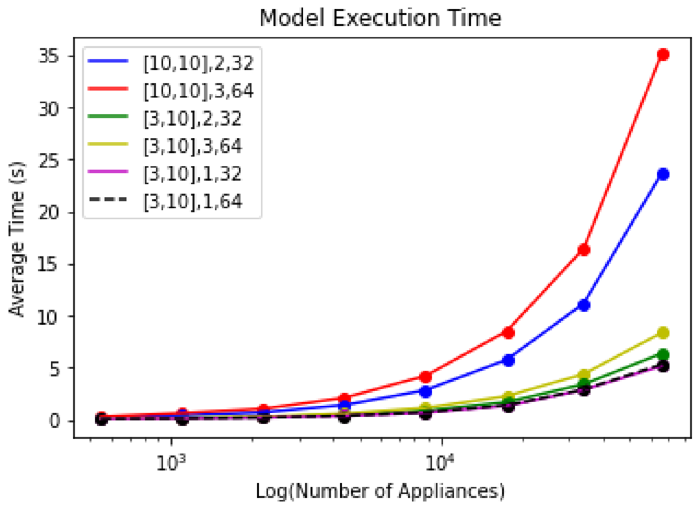


Figure 4.8: Different LSTM models Execution time

for the model should also be considered. For these additional operations, although the time consumption should be less than the time the model takes to predict, it is better to assume a worst scenario. Query times will depend on the data needed for the model, so this will be considered further on the model results. In order to ensure that the prediction time is shorter than the interval between events and that the values forecasted hold relevant information. There are several alternatives:

- Reduce the number of inputs, these could be the number of features, but also the number of past observations.
- Predict farther into the future.
- Use a simpler model.
- Aggregate the data to bigger time windows and predict the average consumption for the next time window.

Of the several alternatives presented, for the results of LSTM model, the number of past observations considered was three, as it also aligns with the with the Partial Auto Correlation results 3.12b, where only the past three events, showed to have some correlation, and only the past event showed a high correlation. Since only the past event has a very high correlation, a FFNN will also be trained, which will rely only on the past event.

CHAPTER 5

Results

Integrate basic detection and prediction solutions for unwanted pattern detection in both time and space, optimizing power grids management, ideally preventive. – The detection method will focus on the identification of areas with high variation of power consumption as well areas of high consumption. – The prediction of appliances energy consumption. • Provide a dashboard to visualize multiple metrics (power consumption, variation of consumption) spatially at multiple scales.

This section presents the findings from the analyses performed in this study, focusing on the two objectives of this dissertation, unwanted pattern detection in both time and space and Visualization.

For abnormal pattern detection, two methods were developed, one focusing on the space dimension, Hotspot detection, whereas for time dimension, the prediction of Active Power, models were trained to predict the feature ActPow. For hotspot Detection and Active Power tests were made to assess the models performance.

For hotspots detection the algorithm is the one presented at 4.4 . For Active Power prediction, the prediction of the feature ActPow, both LSTM and FFNN models were utilized on the dataset

For the visualization a dashboard was created.

5.1 Hotspot Detection

The algorithm for detection is highly scalable, for the case of 10 000 it has an execution time in the interval of [4, 5] where almost 4 of those seconds are spent on queries, resulting in the Database being the bottleneck, so it can be applied to a much larger number of appliances before surpassing the 12 seconds mark, which is the time it usually takes to receive a new event.

As labeled data was not available to test the hotspot detection method, the method to test the hotspot detection was to artificially generate hotspots and apply the model on the artificially generated data. The hotspot detection method was validated at the postcode level

Table 5.1: Codified Labels

Label	Representation
0	Coldspot
1	Coldspot candidates
2	Normal
3	Hotspot Candidates
4	Hotspots

of hierarchy, for each test, a total of 1413 appliances were considered each with a different postcode. The postcodes and their spatial coordinates were obtained through [110], and the values for ActPow and Power Variation were artificially generated. In all tests the same postcodes were used for consistency.

Using artificially generated data hotspots and coldspots will present ideal scenarios for analysis.

Two cases were considered for amount of hotspots and coldspots injected, the first one 1% of postcodes were considered as hotspots and coldspots, each with 0.5%, and the second with 2% of the data, were hotspots and coldspots represented 0.5% each. It is important to keep the percentage low, as it is important to keep the number of hotspots and coldspots low, in order to not have a lot of them, and it is also important to see the behavior of the model as different percentages of hotspots and coldspots were injected.

The postcodes were labeled according to the following table 5.1:

For the evaluation of the hotspot detection algorithm the comparison between the number of hotspots and coldspots detected by the algorithm and the true number of hotspots and coldspots injected will be made. For the results from the model to be considered good, it should detect most of the true injected hotspots and coldspots.

The data generated for the Active Power and Power Variation is different as they have different ranges, and it is easier to generate directly the values for Power Variation for the Normal Distribution, ensuring it passes the Shapiro-Wilk test of normality,

5.1.1 Active Power Hotspot Detection

Since ActPow feature has a range of [0, 100], the data generated for coldspots is going to be on the low end, whereas hotspots is going on the high end, which will make it easier to interpret.

For the Non Normal data, data was generated using a uniform distribution, in the range [30,70], and hotspots and coldspots were injected also using uniform distributions with a range of [80,100] and [0, 20] respectively.

For the Normal data, since the normality test is done on the hotspot and coldspot candidates samples, data was generated for these samples following a normal distribution, $g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{(\frac{-(x-\mu)^2}{2\sigma^2})}$.

For coldspots candidates, the function $g(x)$, had $\sigma = 10$, and $\mu = 25$,, and the injected spots, used a uniform distribution in the range [0, 5].

For hotspots candidates, the function $g(x)$, had $\sigma = 10$, and $\mu = 75$, and the injected spots, used a uniform distribution in the range [95, 100].

The following table 5.2 presents the results obtained, the percentage of data the hotspots and coldspots amount to, as well the number of appliances given a certain label, furthermore it also shows the number of hotspots and coldspots injected as true hotspots and true coldspots respectively.

Table 5.2: Active Power Hotspot Detection results

Data Distribution	Percentage (%)	Coldspots	Label 1	Label 2	Label 3	Hotspots	True Coldspots	True Hotspots
Non Normal	1	10	177	1030	186	10	7	7
Non Normal	2	10	187	1023	183	10	14	14
Normal	1	14	183	1023	183	10	7	7
Normal	2	13	184	1023	183	10	14	14

Analyzing the table 5.2, some early conclusions can be drawn which is that the model with the current threshold, in the case of 7 hotspots and coldspots, correctly identified the true hotspots and coldspots, but it also detected false positives. Whereas in the case with 14 coldspots and hotspots it failed to detect some cases.

For the case of Normal Distribution data, in the case where 1% of the data were considered as spots, it actually detected 1 more coldspots compared to the case with 2% of spots injected. The same as in the Non Normal data occurred where in case with 7 true hotspots and coldspots, it detected false positives besides the true spots and in the case with 14 true spots and coldspots failed to detect 1 coldspot and 4 hotspots.

Analyzing the Non Normal data scenario, it is possible to conclude that increasing the number of injected hotspots and coldspots did not result in an increase of the number of hotspots and coldspots detected by the algorithm, but it slightly decreased the number of postcodes labeled with 2.

This means that the threshold that classifies possible candidate hotspots needs to be tuned to allow a higher number of candidates which then results in a higher number of hotspots. This change would also increase the number of hotspots and coldspots in the case of 1%, which means that the quantile used for hotspots and coldspots, 0.95 and 0.05, needs to be increased and reduced respectively.

5.1.2 Power Variation Hotspot Detection

Since Power Power variation has a range of [-100, 100], so new data needs to be generated.

For the Non Normal data, data was generated using a uniform distribution, in the range [-70,70], and hotspots and coldspots were injected also using uniform distributions with a range of [80,100] and [-100, -80] respectively.

For the Normal data, since the normality test is done on the hotspot and coldspot candidates sample, data was generated for these samples following a normal distribution, $g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{(\frac{-(x-\mu)^2}{2\sigma^2})}$.

For coldspots candidates, the function $g(x)$, had $\sigma = 20$, and $\mu = -55$,, and the injected spots, used a uniform distribution in the range [-100, -90].

For hotspots candidates, the function $g(x)$, had $\sigma = 20$, and $\mu = 55$, and the injected spots, used a uniform distribution in the range [90, 100].

The following table 5.3 presents the results obtained for Power Variation, the percentage of data the hotspots and coldspots amount to, as well the number of appliances given a certain label, furthermore it also shows the number of hotspots and coldspots injected as true hotspots and true coldspots respectively.

Table 5.3: Power Variation Hotspot Detection results

Data Distribution	Percentage (%)	Coldspots	Label 1	Label 2	Label 3	Hotspots	True Coldspots	True Hotspots
Non Normal	1	8	274	821	302	8	7	7
Non Normal	2	7	285	794	316	11	14	14
Normal	1	11	276	821	292	18	7	7
Normal	2	16	276	794	307	20	14	14

Similar to the Active Power results, on the Non Normal data scenario, for 7 true coldspots and hotspots, the algorithm identified false positives, in this case it was only 1 for each of the cases. In the case with 14, it failed to detect 7 coldspots and 3 hotspots.

In the normal data scenario, it is possible to see that it detected false positives in both cases, but the number of hotspots and coldspots increased with the rise of the true coldspots and hotspots.

The results of table 5.3 also show that the increase on the number of true hotspots and coldspots, led to a increase of candidate hotspots and coldspots which also resulted on an overall increase of the number of hotspots and coldspots detected by the algorithm, with the exception of the coldspots on the case of Non Normal data.

Overall it is possible to see the adaptability of the algorithm to the number of hotspots and coldspots, where it shows a tendency to detect a higher number of hotspots and coldspots as the injected hotspots and coldspots increase.

It is also shown that it can detect hotspots and coldspots, although not perfectly, as the thresholds need to be refined considering a multitude of different scenarios, ideally real life situations.

As in a real life scenario the appliances are used for space heating in the winter season, and the rest of the year this feature is not utilized which could require a different set of parameters for the algorithm, furthermore, using different thresholds for Active Power hotspot detection and Power Variation hotspots detection might yield better results.

5.2 Active Power Prediction

Now working with the temporal scale the results of the different architectures will be evaluated, since this is a regression and not a classification problem, the smaller the Loss value is the better the model.

The data for training, validation and testing was the one that resulted from 4.2.2.

Table 5.4: Results obtained for the different LSTM architectures

Number of Layers	Hidden Layer Neurons	Output Activation Layer	Training Loss (MSE)	Validation Loss (MSE)
1	32	ReLU	7.66	6.53
1	32	ELU	7.88	5.91
1	64	ReLU	6.53	5.69
1	64	ELU	3.92	4.12
2	32	ReLU	5.25	4.75
2	32	ELU	7.42	5.70
2	64	ReLU	3.18	2.78
2	64	ELU	3.08	2.74

5.2.1 Long Short-Term Memory

The following table 5.4 provides the Training and Validation Loss for the different LSTM architectures.

Analyzing the results of the table 5.4 it is possible to see that the model with 2 LSTM layers achieved better results on both training and validation loss, when compared to the model with 1 LSTM layer. Furthermore increasing the number neurons of LSTM layers, also increased the performance of the models on both training and loss error. The Output Activation Function achieved better results overall when compared to the function ReLU.

So the best models are the ones with 2 layers and 64 neurons, diving into more details of the models, Model ELU the one with 1 layer, 64 neurons the activation function ELU.

5.2.2 Feedforward Neural Networks

The following table 5.5 presents the results for the different FFNN architectures.

Table 5.5: Results obtained for the different FFNN architectures

Hidden Layer Activation Function	Hidden Layer Neurons	Output Activation Layer	Training Loss (MSE)	Validation Loss (MSE)
Hard Sigmoid	32	ReLU	22.63	14.54
Hard Sigmoid	32	ELU	23.49	14.96
Hard Sigmoid	64	ReLU	20.05	13.34
Hard Sigmoid	64	ELU	21.62	14.55
Tanh	32	ReLU	24.31	15.68
Tanh	32	ELU	22.44	15.64
Tanh	64	ReLU	22.70	14.65
Tanh	64	ELU	22.69	15.66

Analyzing the results, Hard Sigmoid function utilized on the Hidden Layer achieved better results in Validation Loss when compared to the corresponding models with Tanh Hidden Layer Activation, on Training Loss the results were the same with the exception of the case of 32 Hidden Layer Neurons, and the function ELU on the Output Activation Layer.

Overall increasing the number of Hidden layer neurons also resulted in better performance on the models, with the exception of the architecture with Tanh function on the Hidden Layer and ELU function on the Output Activation Function.

Furthermore on the Output Activations Functions, ReLU also achieved better performance, when compared to the function ELU, with the only exception being on the architecture with Hidden Layer Activation Function Tanh and 32 neurons in the hidden layer.

5.3 Detailed Results for the Best Model

In order to select the best model the criteria was the model with the minimum Validation Loss as that was the one that was able to achieve a better generalization for prediction. From both LSTM and FFNN, the one who had the best performance was was the model LSTM with 2 Layers, 64 Neurons and Output Activation Layer ELU with a validation loss of 2.74.

The other metrics for the LSTM model with 2 Layers, 64 Neurons and Output Activation Layer ELU are presented in the table 5.6.

Table 5.6: Results for the best performant model

Metric	Value
Training Loss (MSE)	3.08
Training RMSE	1.76
Training MAE	0.76
Validation Loss	2.74
Validation RMSE	1.66
Validation MAE	0.67
Test Loss (MSE)	9.52
Test RMSE	3.08
Test MAE	1.08

So analyzing the details of Model ELU it is possible to see that the error on the test data is much bigger which is expected, although the results are good, with MAE of 1.08%, meaning that on average the prediction has an error of 1.08%, as Active Power comes in percentage. On testing results, there are some values where the error of the prediction is much bigger, this is evident as MSE is much bigger than the squared value of MAE.

The results obtained for the prediction were good, as the error was low, LSTM obtained a Mean Average Error of 1.08% which is pretty good, considering the scale of 100%. Unfortunately the prediction is not scalable due to a bottleneck in the database, more specifically the query latency, since more columns (representing the features) need to be queried for prediction. The query latency is around 6 seconds at best for a 10 000 thousand of appliances, with the addition of the model prediction and detection it would be next to 12 seconds at best, resulting on only a small amount of information gained, since it would compute the hotspot detection at the time the next event reached, when compared to the real time detection which is [4, 5] seconds later.

Doing both Prediction of Hotspots and Real-Time Detection of hotspots, would slow down both, when compared to only using one of them, as the traffic in the database would increase, and therefore slow down queries even more.

These tests were made on the local machine, which also has an impact on the Database performance. Nevertheless InfluxDB offers Clustering which is a commercial Product and well Cloud Database service which come with a new Engine on the version 3.0 [111] of InfluxDB. This new engine achieves better performance compared to the OSS version which was the version utilized for the tests. So these commercial products might allow a deployment of the prediction model with high scalability as the bottleneck is on the database. If the model used for prediction starts to be the one hindering performance, parallelization can be safely implemented to achieve better latency results although this would require more resources.

So overall the error of the model for prediction is low, which is good, but scalability is an issue. As although a few seconds are gained, there is also the risk of error, that comes with the prediction.

5.4 Visualization

For the visualization, a dashboard was created in Grafana. The main dashboard provides an overall view of the World Map. Where the user can select different topologies to visualize the consumption and power variation. In the main dashboard it is also possible to visualize the historical values of Power Consumption and Power Variation, both on kWh and percentage.

For the visualization data was generated using the same method that was used for the Active Power hotspot detection results analysis, with the difference of not guaranteeing that the data generated was normal or non-normal.

As for the hierarchy the user can select the different ones using the dropdown box, the ones available are country, city, area (general areas represented by first two digits of the postcode) and the postcode, and the world map will display their locations and label for the respective plot. The histograms will show the count of the labels which will represent a summary of the labels the user is visualizing.

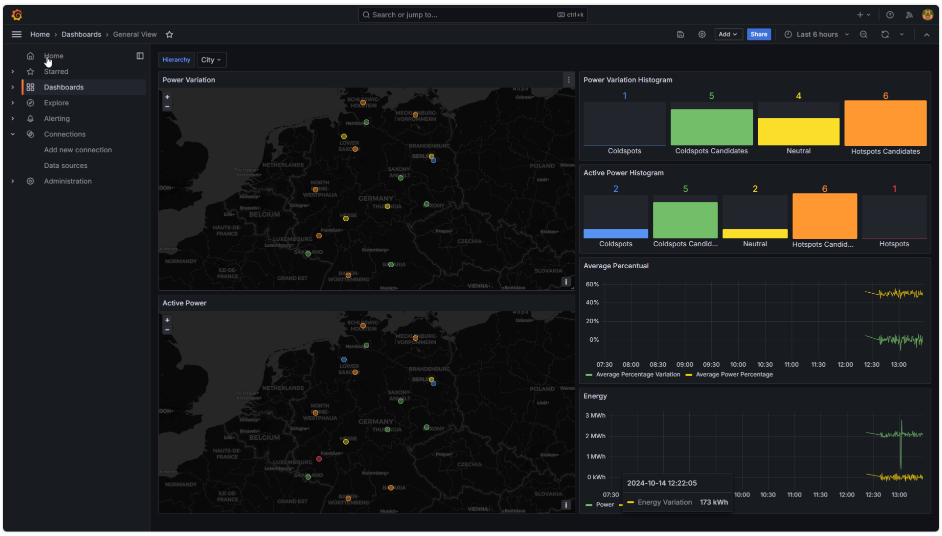


Figure 5.1: Overview of the different cities

Going from Figure 5.1 to 5.3, it's possible to see the topology, as the number of nodes increases the lower the level of topology. This allows the user to verify the root causes of a hotspot or coldspot at higher levels of hierarchy, by stepping down the hierarchy level, and spatially locate major contributors for a either hotspot or coldspot at a more general level. The opposite can also be said, the user can check if an area ends has a big impact on a bigger scale.

Figure 5.2: Overview of the different Postcode Areas

Figure 5.3: Overview of the different Postcodes

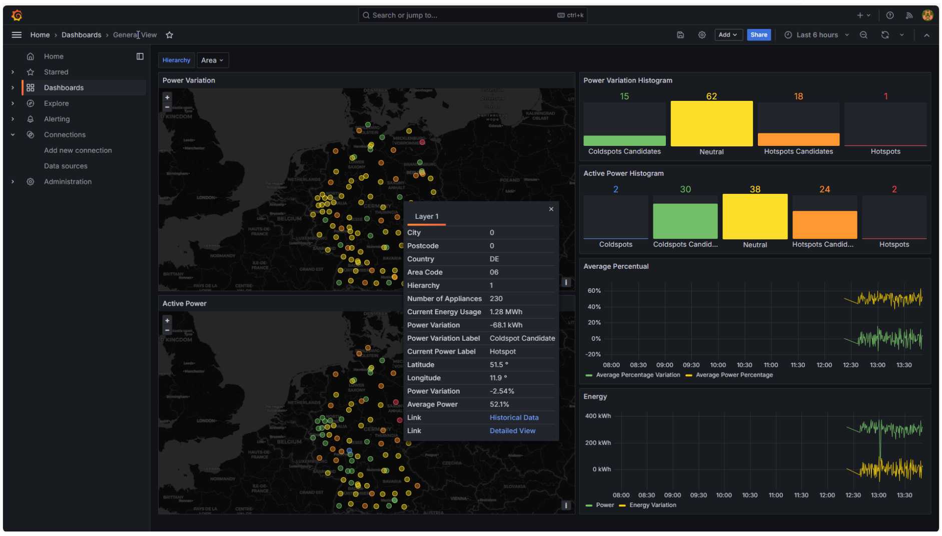


Figure 5.4: Node Pop Up information

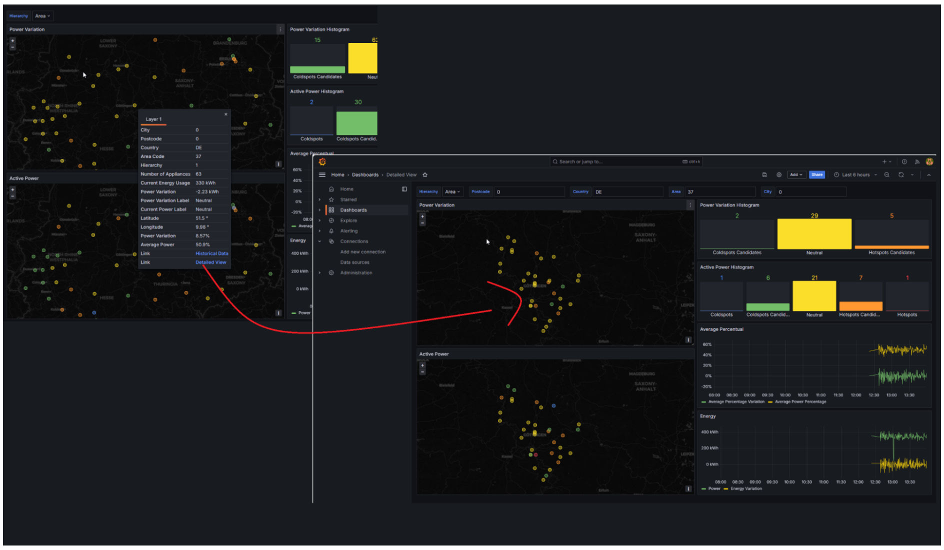


Figure 5.5: Detailed View

Clicking on a node 5.4 of the map, will open the following pop up, demonstrated on the 5.4, this contains Information about the node, as the Number of Appliances evaluated in the node, the Power Consumption and Variation, coordinates, and the information pertaining the location, as the appliances are grouped together at higher levels, depending on the hierarchy some of the camps will have the value 0, since they don't share a location for that level of hierarchy.

Clicking on Historical Data, will update the Historical Plot, to represent that nodes values historically. Even though the data is artificially generated, higher levels of topology will be more susceptible to small variations in historical values. As the historical data is the average of the appliances in a common area.

Whereas clicking on Detailed View will lead to another dashboard, with the same layout, but with the difference of only showing postcodes in the area the node is representing, as represented in figure 5.5.

This other dashboard provides a more detailed view, focusing only on the smaller areas that belong to the area clicked, and make it easier to track root causes of hotspots and coldspots in a higher level of topology.

CHAPTER 6

Conclusion

In this dissertation, the primary objectives were to integrate detection and prediction tools for optimizing power grid management, with an emphasis on identifying high-consumption and high-variation areas, predicting appliance-level consumption, and enabling user-friendly data visualization.

Numerical experiments are presented for the hotspot detection algorithm. It was shown that the algorithm developed is capable of detecting hotspots, not only that, but it is also capable of detecting coldspots. The hotspot detection algorithm demonstrated capabilities to handle different percentages of outliers, although some of the thresholds still need to be refined, as different scenarios might consider 1% or 2% of the all cases, a large number of outliers, which in this case the thresholds should be more restrictive, or the other way around, a bigger percentage is needed. Scalability also plays a major role in the algorithm evaluation, it should be able to compute thousands of appliances before the next event, where the algorithm also succeeded.

As for the prediction, the results obtained were satisfactory, the model with minimum error, the LSTM with 64 neurons, 2 layers, and ELU activation function had a Mean Absolute Error of 1.08% which is good, given how low it is, but scalability issues are present, it can be solvable by allocating more resources to the database, there is no guarantee, nevertheless it was proven that prediction can be made with low margin of error.

The integration of detection and prediction tools for power grid management optimization were met successfully at least for the real time hotspot detection, whereas for prediction progress was made as it had a low error, but it has scalability issues.

The visualization was also successful achieved, being able to visualize the appliances spatially at different levels, going from a generalized view to a more granulated one. It is also possible to inspect specific regions, making it easier to track the origin of hotspots and how they affect a major zone. Furthermore it is also possible to visualize historical data for the zones and number of appliances for a given location.

The solution developed can therefore help Power Grids manage their electricity, by detecting areas of high variation and high consumption, and understanding the areas who require energy. From BOSCH perspective it is also helpful to understand appliances usage spatially, which can be extended further to other features which might yield relevant information about.

6.1 Future Work

Refine hotspot detection algorithm thresholds, as in this thesis, it was tested and the thresholds provide good results on the tested cases. In a real world scenario, to deploy this solution in a production environment, these thresholds might need to be adjusted for the production in order to suit BOSCH needs.

For the deployment of prediction of Active Power of appliances, BOSCH needs to improve the database latency performance, this can be achieved with Cloud databases and the InfluxDB clustering.

In this dissertation a general approach was taking in the hierarchy as postcodes might vary from country to country, but the solution could be improved further, by creating an hierarchy with more levels, this could be achieved by creating another database where spatial data and hierarchies are stored according to the different postcodes formats.

References

[1] *Bosch thermotechnology sales*, https://www.bosch-presse.de/pressportal/de/en/bosch-thermotechnology-posts-record-sales-revenues-of-4-5-billion-euros-for-2022-251520.html, [webpage][Accessed: 2023-11-16].

[2] "Europe heatpumps rollout." (), [Online]. Available: %5Curl%7Bhttps://build-up.ec.europa.eu/system/files/2023-05/ECF-Europes-Leap-to-Heat-Pumps-Report_FINAL_April-2023.pdf%7D.

[3] *German energy act*, https://www.bundesregierung.de/breg-en/news/new-building-energy-act-2185010, [webpage][Accessed: 2023-11-17].

[4] *EU Smart grids and meters*, https://energy.ec.europa.eu/topics/markets-and-consumers/smart-grids-and-meters_en, [webpage] [Accessed 28-11-2023].

[5] *EU Smart grids and meters*, https://digital-strategy.ec.europa.eu/en/policies/eu-policy-digitalisation-energy, [webpage] [Accessed 28-11-2023].

[6] J. Zheng, D. W. Gao, and L. Lin, "Smart meters in smart grid: An overview," in *2013 IEEE Green Technologies Conference (GreenTech)*, 2013, pp. 57–64. DOI: 10.1109/GreenTech.2013.17.

[7] G. R. Barai, S. Krishnan, and B. Venkatesh, "Smart metering and functionalities of smart meters in smart grid - a review," in *2015 IEEE Electrical Power and Energy Conference (EPEC)*, 2015, pp. 138–145. DOI: 10.1109/EPEC.2015.7379940.

[8] I. Diahovchenko, M. Kolcun, Z. Čonka, V. Savkiv, and R. Mykhailyshyn, "Progress and challenges in smart grids: Distributed generation, smart metering, energy storage and smart loads," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, pp. 1319–1333, 4 Dec. 2020, ISSN: 2228-6179. DOI: 10.1007/s40998-020-00322-8.

[9] S. Wasoontarajaroen, K. Pawasan, and V. Chamnanphrai, "Development of an iot device for monitoring electrical energy consumption," in *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2017, pp. 1–4. DOI: 10.1109/ICITEED.2017.8250475.

[10] S. Gan, K. Li, Y. Wang, and C. Cameron, "Iot based energy consumption monitoring platform for industrial processes," in *2018 UKACC 12th International Conference on Control (CONTROL)*, 2018, pp. 236–240. DOI: 10.1109/CONTROL.2018.8516828.

[11] S. Goudarzi, M. H. Anisi, S. A. Soleymani, M. Ayob, and S. Zeadally, "An iot-based prediction technique for efficient energy consumption in buildings," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2076–2088, 2021. DOI: 10.1109/TGCN.2021.3091388.

[12] A. V. Borisovskaya and A. M. Turlikov, "Reducing energy consumption in the iot systems with unlimited number of devices," in *2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, 2021, pp. 1–6. DOI: 10.1109/WECONF51603.2021.9470596.

[13] G. Bedi, G. K. Venayagamoorthy, and R. Singh, "Development of an iot-driven building environment for prediction of electric energy consumption," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4912–4921, 2020. DOI: 10.1109/JIOT.2020.2975847.

[14] *Smart Grids in Germany*, https://www.handelskammer.se/sites/default/files/smart_grids_in_germany_-_web.pdf, [Online; accessed 28-11-2023].

[15] *Consumers Energy: Investing in a Reliable Future*, https://energy.ec.europa.eu/topics/markets-and-consumers/smart-grids-and-meters_en, [webpage] [Accessed 15-05-2023].

[16] *Germany Postal Codes*, https://kids.kiddle.co/Postal_codes_in_Germany, [webpage] [Accessed 17-05-2023].

[17] *France Postal Codes*, https://maps-france.com/maps-france-regions/france-postcode-map, [webpage] [Accessed 17-05-2023].

[18] H. Peter W, D. Pfeiffer, and H. Oshitani, "Prospects for emerging infections in east and southeast asia 10 years after severe acute respiratory syndrome," *Emerging Infectious Diseases*, vol. 19, 6 Jun. 2013, ISSN: 1080-6040. DOI: 10.3201/eid1906.121783.

[19] R. Maciejewski, S. Rudolph, R. Hafen, *et al.*, "A visual analytics approach to understanding spatiotemporal hotspots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 2, pp. 205–220, 2010. DOI: 10.1109/TVCG.2009.100.

[20] P. Mishra, C. M. Pandey, U. Singh, A. Gupta, C. Sahu, and A. Keshri, "Descriptive statistics and normality tests for statistical data.," *Annals of cardiac anaesthesia*, vol. 22, pp. 67–72, 1 2019, ISSN: 0974-5181. DOI: 10.4103/aca.ACA_157_18.

[21] M. Baker, A. Y. Fard, H. Althuwaini, and M. B. Shadmand, "Real-time ai-based anomaly detection and classification in power electronics dominated grids," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 4, no. 2, pp. 549–559, 2023. DOI: 10.1109/JESTIE.2022.3227005.

[22] A. Mahi-al-rashid, F. Hossain, A. Anwar, and S. Azam, "False data injection attack detection in smart grid using energy consumption forecasting," *Energies*, vol. 15, no. 13, 2022, ISSN: 1996-1073. DOI: 10.3390/en15134877. [Online]. Available: https://www.mdpi.com/1996-1073/15/13/4877.

[23] Z. Y. Wu, A. Chew, X. Meng, *et al.*, "Data-driven and model-based framework for smart water grid anomaly detection and localization," *AQUA - Water Infrastructure, Ecosystems and Society*, vol. 71, no. 1, pp. 31–41, Dec. 2021, ISSN: 2709-8028. DOI: 10.2166/aqua.2021.091. eprint: https://iwaponline.com/aqua/article-pdf/71/1/31/1101230/jws0710031.pdf. [Online]. Available: https://doi.org/10.2166/aqua.2021.091.

[24] S. Saleem, M. Aslam, and M. Shaukat, "A review and empirical comparison of univariate outlier detection methods," 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237612582.

[25] F. Wang, W. Li, S. Wang, and C. R. Johnson, "Association rules-based multivariate analysis and visualization of spatiotemporal climate data," *ISPRS International Journal of Geo-Information*, vol. 7, no. 7, 2018, ISSN: 2220-9964. DOI: 10.3390/ijgi7070266. [Online]. Available: https://www.mdpi.com/2220-9964/7/7/266.

[26] G. A. Pagani and M. Aiello, "The power grid as a complex network: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 11, pp. 2688–2700, 2013, ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2013.01.023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437113000575.

[27] P. D. Domański, "Study on statistical outlier detection and labelling," *International Journal of Automation and Computing*, vol. 17, pp. 788–811, 6 Dec. 2020, ISSN: 1476-8186. DOI: 10.1007/s11633-020-1243-2.

[28] A. Gnilke and T. G. M. Sanders, "Distinguishing abrupt and gradual forest disturbances with modis-based phenological anomaly series," *Frontiers in Plant Science*, vol. 13, May 2022, ISSN: 1664-462X. DOI: 10.3389/fpls.2022.863116.

[29] F. Amin and G. S. Choi, "Hotspots analysis using cyber-physical-social system for a smart city," *IEEE Access*, vol. 8, pp. 122 197–122 209, 2020. DOI: 10.1109/ACCESS.2020.3003030.

[30] R. Chen, M. Chen, W. Li, and N. Guo, "Detecting community structure of urban hotspot regions," in 2020, pp. 265–278. DOI: 10.1007/978-981-15-3707-3_26.

[31] M. S. Tabarej and S. Minz, "Rough-graph-based hotspot detection of polygon vector data," *Multimedia Tools and Applications*, vol. 83, pp. 16 683–16 710, 6 Jul. 2023, ISSN: 1573-7721. DOI: 10.1007/s11042-023-16246-4.

[32] J. S. Pengxiang Zhao Xintao Liu and M. Chen, "A network distance and graph-partitioning-based clustering method for improving the accuracy of urban hotspot detection," *Geocarto International*, vol. 34, no. 3, pp. 293–315, 2019. DOI: 10.1080/10106049.2017.1404140. [Online]. Available: https://doi.org/10.1080/10106049.2017.1404140.

[33] M. Ghahramani, M. Zhou, and C. T. Hon, "Mobile phone data analysis: A spatial exploration toward hotspot detection," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 351–362, 2019. DOI: 10.1109/TASE.2018.2795241.

[34] S. Gregory, B. J. L. Berry, and D. F. Marble, "Spatial analysis: A reader in statistical geography," *The Geographical Journal*, vol. 134, 2 1968, ISSN: 00167398. DOI: 10.2307/1792496.

[35] S. Pourbahrami, M. A. Balafar, L. M. Khanli, and Z. A. Kakarash, "A survey of neighborhood construction algorithms for clustering and classifying data points," *Computer Science Review*, vol. 38, p. 100 315, 2020, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2020.100315. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013720304159.

[36] S. Bloemheuvel, J. van den Hoogen, and M. Atzmueller, "Graph construction on complex spatiotemporal data for enhancing graph neural network-based approaches," *International Journal of Data Science and Analytics*, vol. 18, pp. 157–174, 2 Aug. 2024, ISSN: 2364-415X. DOI: 10.1007/s41060-023-00452-2.

[37] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, p. 420, 6 Nov. 2021, ISSN: 2662-995X. DOI: 10.1007/s42979-021-00815-1.

[38] "Annual scholarly publications on artificial intelligence." (), [Online]. Available: %5Curl%7Bhttps://ourworldindata.org/grapher/annual-scholarly-publications-on-artificial-intelligence%7D.

[39] A. Jadon, A. Patil, and S. Jadon, *A comprehensive survey of regression based loss functions for time series forecasting*, 2022. arXiv: 2211.02989 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2211.02989.

[40] S. Sun, Z. Cao, H. Zhu, and J. Zhao, *A survey of optimization methods from a machine learning perspective*, 2019. arXiv: 1906.06821 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1906.06821.

[41] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. DOI: 10.1109/TIT.1967.1053964.

[42] P. Sasikumar and S. Khara, "K-means clustering in wireless sensor networks," in *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 140–144. DOI: 10.1109/CICN.2012.136.

[43] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000, ISSN: 0163-5808. DOI: 10.1145/335191.335388. [Online]. Available: https://doi.org/10.1145/335191.335388.

[44] Z. Cheng, C. Zou, and J. Dong, "Outlier detection using isolation forest and local outlier factor," in *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, ser. RACS '19, Chongqing, China: Association for Computing Machinery, 2019, pp. 161–168, ISBN: 9781450368438. DOI: 10.1145/3338840.3355641. [Online]. Available: https://doi.org/10.1145/3338840.3355641.

[45] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, 1996, pp. 226–231.

[46] R. Huang, T. Huang, R. Gadh, and N. Li, "Solar generation prediction using the arma model in a laboratory-level micro-grid," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, 2012, pp. 528–533. DOI: 10.1109/SmartGridComm.2012.6486039.

[47] G. L. Vara Prasad, B. Ravi Teja, S. Govathoti, and S. Dhanikonda, "Leveraging arma and armax time-series forecasting models for rainfall prediction," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2023, pp. 353–357. DOI: 10.1109/ICACCS57279.2023.10113031.

[48] V. Vapnik, “An overview of statistical learning theory,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999. DOI: 10.1109/72.788640.

[49] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[50] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, Mar. 2012, ISSN: 1556-4681. DOI: 10.1145/2133360.2133363. [Online]. Available: https://doi.org/10.1145/2133360.2133363.

[51] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 1 2001, ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[52] S. Zagoruyko and N. Komodakis, *Wide residual networks*, 2017. arXiv: 1605.07146 [cs.CV].

[53] G. Bebis and M. Georgiopoulos, “Feed-forward neural networks,” *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994. DOI: 10.1109/45.329294.

[54] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation,” ACM, Jun. 2007, pp. 473–480, ISBN: 9781595937933. DOI: 10.1145/1273496.1273556.

[55] O. Shamir, “Without-replacement sampling for stochastic gradient methods,” in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/c74d97b01eae257e44aa9d5bade97baf-Paper.pdf.

[56] “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” Feb. 2016.

[57] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” Jul. 2016.

[58] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, “Understanding and improving layer normalization,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/2f4fe03d77724a7217006e5d16728874-Paper.pdf.

[59] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 448–456. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html.

[60] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” Sep. 2016.

[61] S. Ioffe, “Batch renormalization: Towards reducing minibatch dependence in batch-normalized models,” Feb. 2017.

[62] “Improving neural networks by preventing co-adaptation of feature detectors,” Jul. 2012.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” vol. 15, no. 1, pp. 1929–1958, Jan. 2014, ISSN: 1532-4435.

[64] “A theoretically grounded application of dropout in recurrent neural networks,” Dec. 2015.

[65] “Regularizing and optimizing lstm language models,” Aug. 2017.

[66] C. Wei, S. Kakade, and T. Ma, “The implicit and explicit regularization effects of dropout,” in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 13–18 Jul 2020, pp. 10 181–10 192. [Online]. Available: https://proceedings.mlr.press/v119/wei20d.html.

[67] R. J. Williams and D. Zipser, “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks,” *Neural Computation*, vol. 1, no. 2, pp. 270–280, Jun. 1989, ISSN: 0899-7667. DOI: 10.1162/

neco.1989.1.2.270. eprint: https://direct.mit.edu/neco/article-pdf/1/2/270/811849/neco.1989.1.2.270.pdf. [Online]. Available: https://doi.org/10.1162/neco.1989.1.2.270.

[68] J. Connor, R. Martin, and L. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994. DOI: 10.1109/72.279188.

[69] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Scalable recurrent neural network for hyperspectral image classification," *The Journal of Supercomputing*, vol. 76, pp. 8866–8882, 11 Nov. 2020, ISSN: 0920-8542. DOI: 10.1007/s11227-020-03187-0.

[70] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, 2015. arXiv: 1511.08458 [cs.NE].

[71] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[72] K. Tornai, L. Kovács, A. Oláh, *et al.*, "Classification for consumption data in smart grid based on forecasting time series," *Electric Power Systems Research*, vol. 141, pp. 191–201, Dec. 2016, ISSN: 03787796. DOI: 10.1016/j.epsr.2016.07.018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378779616302735.

[73] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1643–1647. DOI: 10.1109/ICACCI.2017.8126078.

[74] "An efficient hybrid model for appliances classification based on time series features," *Energy and Buildings*, vol. 266, p. 112 087, Jul. 2022, ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2022.112087.

[75] M. Xia, H. Shao, X. Ma, and C. W. de Silva, "A stacked gru-rnn-based approach for predicting renewable energy and electricity load for smart grid operation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7050–7059, 2021. DOI: 10.1109/TII.2021.3056867.

[76] R.-G. Wang, W.-J. Ho, K.-C. Chiang, *et al.*, "Analyzing long-term and high instantaneous power consumption of buildings from smart meter big data with deep learning and knowledge graph techniques," *Energies*, vol. 16, p. 6893, 19 Sep. 2023, ISSN: 1996-1073. DOI: 10.3390/en16196893.

[77] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5. DOI: 10.1109/WTS.2018.8363930.

[78] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, ser. MLSDA'14, Gold Coast, Australia QLD, Australia: Association for Computing Machinery, 2014, pp. 4–11, ISBN: 9781450331593. DOI: 10.1145/2689746.2689747. [Online]. Available: https://doi.org/10.1145/2689746.2689747.

[79] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5. DOI: 10.1109/WTS.2018.8363930.

[80] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for iot time series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 112–122, 2022. DOI: 10.1109/TSMC.2020.2968516.

[81] H. Güneyli and S. M. S. Ahmed, "Detecting abnormal seismic activity areas of anatolian plate and deformation directions using python geospatial libraries," *Heliyon*, vol. 9, e14394, 3 Mar. 2023, ISSN: 24058440. DOI: 10.1016/j.heliyon.2023.e14394. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405844023016018.

[82] H.-S. Kim, C.-G. Sun, M.-G. Lee, and H.-I. Cho, "Multivariate geotechnical zonation of seismic site effects with clustering-blended model for a city area, south korea," *Engineering Geology*, vol. 294, p. 106 365, Dec. 2021, ISSN: 00137952. DOI: 10.1016/j.enggeo.2021.106365. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0013795221003768.

[83] M. Shariati, T. Mesgari, M. Kasraee, and M. Jahangiri-rad, "Spatiotemporal analysis and hotspots detection of covid-19 using geographic information system (march and april, 2020)," *Journal of Environmental Health Science and Engineering*, vol. 18, pp. 1499–1507, 2 Dec. 2020, ISSN: 2052-336X. DOI: 10.1007/s40201-020-00565-x. [Online]. Available: https://link.springer.com/10.1007/s40201-020-00565-x.

[84] K. B. Gurumoorthy, S. P. Vimal, N. S. Kumar, and M. Kasiselvanathan, "Air pollution hotspot detection and identification of their source trajectory," *Journal of Physics: Conference Series*, vol. 1917, p. 012 029, 1 Jun. 2021, ISSN: 1742-6588. DOI: 10.1088/1742-6596/1917/1/012029. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1917/1/012029.

[85] B. Su, H. Chen, K. Liu, and W. Liu, "Rcag-net: Residual channelwise attention gate network for hot spot defect detection of photovoltaic farms," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–14, 2021. DOI: 10.1109/TIM.2021.3054415.

[86] U. M. Butt, S. Letchmunan, F. H. Hassan, M. Ali, A. Baqir, and H. H. R. Sherazi, "Spatio-temporal crime hotspot detection and prediction: A systematic literature review," *IEEE Access*, vol. 8, pp. 166 553–166 574, 2020. DOI: 10.1109/ACCESS.2020.3022808.

[87] M. H. Sazlı, "A brief review of feed-forward neural networks," *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006. DOI: 10.1501/commua1-2_0000000026.

[88] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[89] X. H. Le, H. Ho, G. Lee, and S. Jung, "Application of long short-term memory (lstm) neural network for flood forecasting," *Water*, vol. 11, p. 1387, Jul. 2019. DOI: 10.3390/w11071387.

[90] J. Mostafa, S. Wehbi, S. Chilingaryan, and A. Kopmann, "Scits: A benchmark for time-series databases in scientific experiments and industrial internet of things," in *Proceedings of the 34th International Conference on Scientific and Statistical Database Management*, ser. SSDBM '22, Copenhagen, Denmark: Association for Computing Machinery, 2022, ISBN: 9781450396677. DOI: 10.1145/3538712.3538723. [Online]. Available: https://doi.org/10.1145/3538712.3538723.

[91] A. Struckov, S. Yufa, A. A. Visheratin, and D. Nasonov, "Evaluation of modern tools and techniques for storing time-series data," *Procedia Computer Science*, vol. 156, pp. 19–28, 2019, 8th International Young Scientists Conference on Computational Science, YSC2019, 24-28 June 2019, Heraklion, Greece, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.08.125. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919310439.

[92] I. Sychev, A. Abdelkader, W. Kozak, R. Bonetto, and F. H. P. Fitzek, "Closed loop benchmark for timeseries databases," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6. DOI: 10.1109/CCNC46108.2020.9045409.

[93] A. Khelifati, M. Khayati, A. Dignös, D. Difallah, and P. Cudré-Mauroux, "Tsm-bench: Benchmarking time series database systems for monitoring applications," *Proc. VLDB Endow.*, vol. 16, no. 11, pp. 3363–3376, Jul. 2023, ISSN: 2150-8097. DOI: 10.14778/3611479.3611532. [Online]. Available: https://doi.org/10.14778/3611479.3611532.

[94] S. Di Martino, L. Fiadone, A. Peron, A. Riccabone, and V. N. Vitale, "Industrial internet of things: Persistence for time series with nosql databases," in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2019, pp. 340–345. DOI: 10.1109/WETICE.2019.00076.

[95] B. Shah, P. M. Jat, and K. Sashidhar, *Performance study of time series databases*, 2022. arXiv: 2208.13982 [cs.DB].

[96] F. Puentes, M. D. Pérez-Godoy, P. González, and M. J. D. Jesus, "An analysis of technological frameworks for data streams," *Progress in Artificial Intelligence*, vol. 9, pp. 239–261, 3 Sep. 2020, ISSN: 2192-6352. DOI: 10.1007/s13748-020-00210-6.

[97] L. Hernández, C. Baladrón, J. M. Aguiar, *et al.*, "A study of the relationship between weather variables and electric power demand inside a smart grid/smart world framework," *Sensors*, vol. 12, pp. 11 571–11 591, 9 Aug. 2012, ISSN: 1424-8220. DOI: `10.3390/s120911571`.

[98] S. M. Stigler, "Francis galton's account of the invention of correlation," *Statistical Science*, vol. 4, no. 2, pp. 73–79, 1989, ISSN: 08834237. [Online]. Available: `http://www.jstor.org/stable/2245329` (visited on 01/19/2024).

[99] P. Wei, Z. Lu, and J. Song, "Variable importance analysis: A comprehensive review," *Reliability Engineering System Safety*, vol. 142, pp. 399–432, 2015, ISSN: 0951-8320. DOI: `https://doi.org/10.1016/j.ress.2015.05.018`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0951832015001672`.

[100] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data Mining and Knowledge Discovery*, vol. 2, pp. 345–389, 4 1998, ISSN: 13845810. DOI: `10.1023/A:1009744630224`.

[101] G. Louppe, "Understanding random forests: From theory to practice," Jul. 2014.

[102] W. Paja, "Feature selection methods based on decision rule and tree models," in *Intelligent Decision Technologies 2016*, I. Czarnowski, A. M. Caballero, R. J. Howlett, and L. C. Jain, Eds., Cham: Springer International Publishing, 2016, pp. 63–70, ISBN: 978-3-319-39627-9.

[103] S. Chiang, J. Zito, V. R. Rao, and M. Vannucci, "Time-series analysis," in *Statistical Methods in Epilepsy*, Chapman and Hall/CRC, 2024, pp. 166–200.

[104] J. H. F. Flores, P. M. Engel, and R. C. Pinto, "Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–8. DOI: `10.1109/IJCNN.2012.6252470`.

[105] E. Tipton, K. Hallberg, L. V. Hedges, and W. Chan, "Implications of small samples for generalization: Adjustments and rules of thumb," *Evaluation Review*, vol. 41, no. 5, pp. 472–505, 2017, PMID: 27402612. DOI: `10.1177/0193841X16655665`. [Online]. Available: `https://doi.org/10.1177/0193841X16655665`.

[106] S. S. SHAPIRO and M. B. WILK, "An analysis of variance test for normality (complete samples)†," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, Dec. 1965, ISSN: 0006-3444. DOI: `10.1093/biomet/52.3-4.591`. eprint: `https://academic.oup.com/biomet/article-pdf/52/3-4/591/962907/52-3-4-591.pdf`. [Online]. Available: `https://doi.org/10.1093/biomet/52.3-4.591`.

[107] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, *A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru*, 2024. arXiv: `2305.17473 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2305.17473`.

[108] M. Cao, R. Yao, J. Xia, K. Jia, and H. Wang, "Lstm attention neural-network-based signal detection for hybrid modulated faster-than-nyquist optical wireless communications," *Sensors*, vol. 22, p. 8992, 22 Nov. 2022, ISSN: 1424-8220. DOI: `10.3390/s22228992`.

[109] *Tensorflow*, `https://www.tensorflow.org/install/pip?hl=pt-br`, [webpage][Accessed: 2023-05-17].

[110] *Geonames*, `https://www.geonames.org`, [webpage][Accessed: 2023-05-23].

[111] *Influxdb 3.0*, `https://www.influxdata.com/lp/oss-vs-new-engine/`, [webpage][Accessed: 2023-06-09].