Masters Program in Geospatial Technologies

Sharing ML models using IoT Communities of Interest based on Data Similarity and Location.

Michael Sirya Mwaruah.

Dissertation submitted in partial fulfilment of the requirements for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS Information Management School
Universitat Jaume I
Westfälische Wilhelms-Universität Münster

Sharing ML models using IoT Communities of Interest based on Data Similarity and Location.

Dissertation supervised by:

Dr. Sergi Trilles Oliver, PhD
Universität Jaume I
Castellón, Spain.

Co-supervised by:

Prof. Dr. Marco Octávio Trindade Painho, PhD
NOVA Information Management School
Lisbon, Portugal.

Prof. Dr. Ana Cristina Costa, PhD
NOVA Information Management School
Lisbon, Portugal.

I affirm that the thesis titled " Sharing ML models using IoT Communities of Interest based on Data Similarity and Location." represents my original work, and I have not utilized any sources or aids beyond those explicitly mentioned. Any content or quotations from external sources, including electronic media, have been appropriately credited and referenced. I consent to having my thesis undergo plagiarism checks to ensure its originality and agree to its inclusion in a database for this purpose.

Michael Sirya Mwaruah
Castellon de la Plana
20 February 2024.

Abstract

The increasing number of Internet of Things devices, propelled by technological advancements and widespread Internet coverage, has led to an unprecedented surge in data generation. This influx of data, often characterized as Big Data, poses significant challenges in terms of handling, processing, and extracting meaningful insights.

The rise of Artificial Intelligence is crucial for the managment of Big Data, establishing a foundation for the perfect symbiosis of Internet of Things and Artificial Intelligence. Internet of Things devices generate enough data to feed Machine Learning a subset of Artificial Intelligence. Machine Learning specializes in recognizing patterns, discerning intricate trends and anomalies, streamlining data analysis through automation, and exhibiting scalability to effortlessly accommodate expanding data quantities. Machine Learning's strengths in predictive analytics and real-time processing makes it highly suitable for prompt decision-making.

Despite promising prospects, creating and deploying Machine Learning models in numerous heterogeneous Internet of Things devices and ecosystems presents a formidable and competitive task.Similarly, developing a universal ML model capable of encompassing all IoT devices worldwide is an impractical endeavor since each IoT device comes with its unique characteristics and functionalities, making a one-size-fits-all model unfeasible. Therefore, this study proposes a novel solution in which Machine Learning models are shared among Internet of Things devices based on their similarity in purpose, domain, and context. This strategy leverages the concept of Communities of Interest within the Social Internet of Things framework.

The main goal of this master thesis is to develop an efficient method for sharing Machine Learning models across Internet of Things devices. To achieve this, the research work proposes a novel approach focused on distributing Machine Learning models among Internet of Things Communities of Interest based on the similarity of Internet of Things data streams and geospatial components such as location and elevation. To validate this approach, the study adopted a cluster-based strategy to form Internet of Things Communities of Interest. Initially, a thorough similarity analysis of IoT weather sensor data streams was conducted using both Dynamic Time Warping and Spearman's correlation methods.

Evaluation of the similarity results revealed that Spearman's correlation performed better than Dynamic Time Warping, producing higher-quality and more coherent clusters. Thus, the study proceeded with K-means clustering using the outcomes of Spearman's correlation analysis and goespatial data to form clusters, guided by the optimal number of clusters, four, determined through the elbow method.

These clusters formed the foundation for Internet of Things - Communities of Interest, essential for the development, validation, testing, and sharing of Machine Learning models. Evaluation of Machine Learning model performance during the sharing and testing phases revealed that the majority of the Machine Learning models performed better when trained, tested, and shared within the same Community of Interest dataset. On the contrary, models trained on a different Community of Interest exhibited poorer performance when tested on members of another Community of Interest.

The findings of this study demonstrate that it is possible to delineate geospatial zones based on the inherent similarity of Internet of Things data streams, and to craft and validate Machine Learning models tailored to the unique characteristics of each zone. It also establishes that it possible to leverage geospatial components for sharing and reusing pre-trained Machine Learning models among Internet of Things devices.

Acknowledgements

First of all, I would like to express my sincere thankful gratitude to Ph.D. Sergi Trilles, supervisor, for his helpful advice and feedback towards the whole work, from topic proposal stage until the conclusion. Giving my special thanks to Prof. Marco Painho and Prof. Ana Christina Costa as well for being co-supervisors of this work.

In addition,I express my thanks to the colleagues of this master programme in Geospatial Technologies, for all in Universitat Jaume I, Universität Münster, and Universidade NOVA de Lisboa, for sharing thoughts and knowledge during the course.

I convey my grateful appreciation to Erasmus Mundus Programme for the scholarship as being financial support during this master programme.

I would like to express my thanks to my family for always being supportive despite the distance.

Last but not least, I gratefully acknowledge the Most High God for His unwavering protection and blessings, which have safeguarded my health and safety throughout this Master program.

Contents

List of Figures

List of Tables

Acronyms

Abbreviation	Meaning
IoT	Internet of Things
AI	Artificial Intelligence
ML	Machine Learning
CoI	Community of Interest
SIoT	Social Internet of Thing
CoIoTIA	Communities of Internet of Things in Artificial Intelligence
AIoT	Artificial Intelligence Internet of Things
IoV	Internet of Vehicles
DTW	Dynamic Time Warping
MODH	Modified Hausdorf
LCSS	Longest Common Sub-Sequence
HMM	Hidden Markov Models
SOM	Self Organizing Maps
SVM	Support Vector Machines
RFID	Radio Frequency Identification
SVOs	Social Virtual Objects
RWOs	Real World Objects
SVOR	Social Virtual Object Root
SN	Social Network
eLSA	extended local similarity analysis
CDIISN	Community Detection In an Intergrated Social Network.
DCIM	Dynamic Community of Interest Model
DSPL	Dynamic Software Product Lines
CBR	Case-Based Reasoning
HPM	Hierarchical Pattern Matching
VPN	Virtual Private Network
LSTM	Long Short Term Memory
DL	Deep Learning
RMSE	Root Mean Square Error
SSE	Sum of Squares Error
WCSS	Within-Cluster Sum of Squares
MSE	Mean Squared Error
MLaaS	Machine Learning as a Service
CP-ABPRE	Ciphertext Policy Attribute Based Proxy Re-encryption
EI	Edge intelligence
CPA	Chosen Plaintext Attacks
NCV	Nested cross-validation

1. Introduction

1.1 Context and Motivation

Today, rapid advances in technology have led to the widespread adoption of Internet of Things (IoT) systems in various areas such as smart homes, healthcare, agriculture, and transportation Granell et al. [2020], Trilles et al. [2015, 2020, 2017]. According to Karie et al. [2020], the number of interconnected IoT devices will increase to 38.6 billion by 2025 and is expected to reach approximately 50 billion by 2030. The advent of IoT ushers in an era where all objects in our surroundings will be connected to the Internet, enabling seamless communication between them with minimal human intervention [Atlam et al., 2018].

However, this proliferation of IoT devices comes with a challenge: the huge amounts of data they generate. Managing and analyzing these data, especially when real-time analysis is needed for tasks such as health monitoring, emergency response, security, and smart assistants, has become increasingly complex [Yu and Wang, 2020].

According to Adi et al. [2020] the emergence of Machine Learning (ML) is crucial to processing and analyzing IoT data due to the unprecedented scale and complexity of information generated by interconnected devices. Traditional data analysis methods struggle to handle the sheer volume, velocity, and variety of IoT data. ML algorithms excel in discerning patterns, identifying anomalies, and extracting meaningful insights from this data deluge [Trilles et al., 2024, Hammad et al., 2023]. Using ML, organizations can predict future events, optimize resource utilization, enable real-time decision-making, and adapt to changing circumstances. Essentially, ML empowers IoT systems to derive actionable intelligence from massive datasets, unlocking the full potential of IoT in diverse applications and industries.

The rapid advancement of IoT devices enables them to undertake more intricate computational tasks. This has resulted in the emergence of Artificial Intelligence of Things AIoT, driven by the synergy of AI and 5G technology. This trend offers two key strategies: a centralized system based on cloud computing for data analysis and decision-making, and a decentralized approach, known as edge computing, capable of generating immediate responses at the data source [Pinyoanuntapong et al., 2022]. This progress, in turn, has facilitated the development of more sophisticated analysis algorithms within the edge computing layer.

Each strategy has distinct advantages and drawbacks. The centralized model [Corchado, 2020] offers simplicity during the implementation and deployment of the model, while the decentralized approach reduces the costs and time associated

with data transfer. The choice between these strategies depends on specific use-case requirements. For example, applications that require real-time decision making, such as health or safety systems, often benefit from the decentralized approach [Corchado, 2020]. In contrast, scenarios that require extensive computational power may favor the centralized strategy, aligning with the unique needs of the application.

In connection with the AIoT paradigm, a compelling concept known as TinyML has surfaced [Ray, 2022, Trilles et al., 2024]. TinyML focuses on creating ML models specifically optimized for execution on IoT devices. This approach enables ML models to operate directly where data is generated or where decisions need to be implemented. The ultimate goal is to empower IoT devices to autonomously make decisions, transforming them from mere messengers conveying information to the control center into active decision-makers [Ray, 2022].

In a diverse field such as IoT, characterized by a wide range of devices, purposes, and dynamic contexts, generating and managing specific models for each individual IoT device is a daunting task. Given the intricate and diverse nature of IoT devices and ecosystems, developing a universal ML model capable of encompassing all IoT devices worldwide becomes an impractical endeavor. Each IoT device comes with its unique characteristics and functionalities, making a one-size-fits-all model unfeasible. A potential solution lies in the ability to share ML models among IoT devices based on their similarity in purpose, domain, and context.

A possible solution is to share these models is using the concept of a Community of Interest (CoI) [Bao et al., 2013]. CoI is a strategic approach to facilitate the sharing of models among IoT devices. It originates from the Social Internet of Things (SIoT) realm Atzori et al. [2011], allowing IoT nodes to be grouped according to shared interests or purposes. SIoT extends the scope of IoT by enabling IoT devices (Social Objects) to establish autonomous relationships, similar to trust relationships between humans [Shahab et al., 2022]. Within a CoI, there are three types of social relationships an IoT device can benefit from: friendship, representing intimacy; CoI, signifying common purpose or knowledge; and social contact, denoting closeness and proximity. This concept is particularly valuable for improving cooperation and effectively delivering services through autonomous collaboration. CoI has previously been employed to establish trust between IoT devices, with each community being overseen by a trusted administrator managing membership [Djedjig et al., 2020].

Moreover, IoT devices within a CoI must adhere to acceptance conditions to maintain their participation. According to Sagar et al. [2021], considerations of temporal and geospatial components become essential when defining a CoI. The temporal aspect recognizes that devices may undergo changes over time, and the geospatial condition recognizes that the behavior of an IoT device can vary depending on its

location, especially in contexts such as the Internet of Vehicles (IoV) where devices are in constant motion [Adnan et al., 2019].

Therefore, it is imperative to incorporate temporal and geospatial components when defining and maintaining a CoI to determine the eligibility of IoT devices within the community [Sagar et al., 2021]. Leveraging these components in an ML parsing system is highly beneficial to improve adaptability in addressing the same underlying challenges. The term ML will be used to refer to the TinyML system to be shared in this project.

This research aims to establish a new method of sharing of ML models among IoT devices using CoIs based on the similarity of IoT data streams and geospatial components that is location and elevation. It leverages ML models for forecasting temperature through simulations using real-world IoT data from weather sensor networks to demonstrate the effectiveness and relevance of the proposed approach in addressing real-world challenges and advancing the field of IoT and ML.

To achieve this goal, the project will perform a comprehensive time series analysis of temperature data collected from weather sensors as IoT devices. The analysis will identify patterns of similarity in the data. It will then integrate temperature time-series similarity data with geospatial attributes such as location and elevation to execute geospatial clustering of the IoT devices. This approach will generate CoI, and organize sensors based on their similarities. As a result, a foundation and framework for seamless crafting, validation, testing, and experimenting with the sharing of TinyML models among interconnected IoT devices will be presented.

1.2 Main Objective

The main objective of this master thesis is to define a strategy to share and reuse pre-trained ML models among IoT devices using data similarity and geospatial components.

1.2.1 Specific Objectives

To provide a clear roadmap for achieving the broader goal of enabling the sharing and re-use of pre-trained ML models among IoT devices this study addresses this aim by, breaking down the overarching goal into smaller, manageable objectives that guide the research process. The specific objectives of this study are as follows:

1. Following the concept of CoIs, to establish geospatial zones based on the inherent similarity of IoT data streams.

2. Use these geospatial zones to design, validate, test, and experiment the sharing of ML models tailored to the unique characteristics of each geospatial zone.

1.3 Research Questions

Crucial for understanding how to enable the sharing and re-use of pre-trained ML models among IoT devices this study also addresses fundamental research questions. These questions provide clear direction for exploring the complexities of model sharing within IoT environments. This research aims to uncover meaningful findings and contribute to a better understanding of how to effectively distribute models in IoT systems by carefully examining the following reasearch questions:

1. Can IoT devices be grouped into CoI based on the similarity of temporal data and geospatial attributes?
2. Can grouping IoT devices into CoI based on their data similarity enable the sharing of Tiny ML models among the devices?

1.4 CoIoTIA project

This study is part of the broader national project, Communities of Internet of Things in Artificial Intelligence (CoIoTIA), aimed at enhancing the practical implementation of AIoT devices, particularly those equipped with edge computing capabilities. To realize this objective, a ML analysis platform will be developed, creating a decentralized repository hosting pre-trained models and the necessary tools for adaptability, illustrated in Figure 1.1.

The repository will be filled with micro ML models generated through local learning algorithms, distributed using the CoI paradigm derived from the SIoT. These models are customized for context-dependent IoT devices, considering factors like location and time. The study embraces a distributed IoT architecture, departing from the conventional centralized model reliant on cloud computing, opting instead for the computational power of edge/fog computing to provide scalability and low latency. In parallel, the ML model repositories reside in fog nodes (Figure 1.1), ensuring the publication, reuse, and reproducibility of integrated models. Simultaneously, these fog elements become integral components of a blockchain network, contributing to scalability, security, transparency, and traceability.

The research introduces an innovative approach to facilitate the sharing of ML models within IoT CoI, based on the similarity of IoT data streams and geospatial components (location and elevation). The goal is to devise a method for sharing models stored in fog nodes among IoT devices. The anticipated outcomes of the project are expected to exert a significant impact across various business sectors, including smart cities, smart homes, and industry, among others.

This study contributes to the project by proposing a novel method for sharing and reusing pre-trained ML models among IoT devices. This objective will be achieved

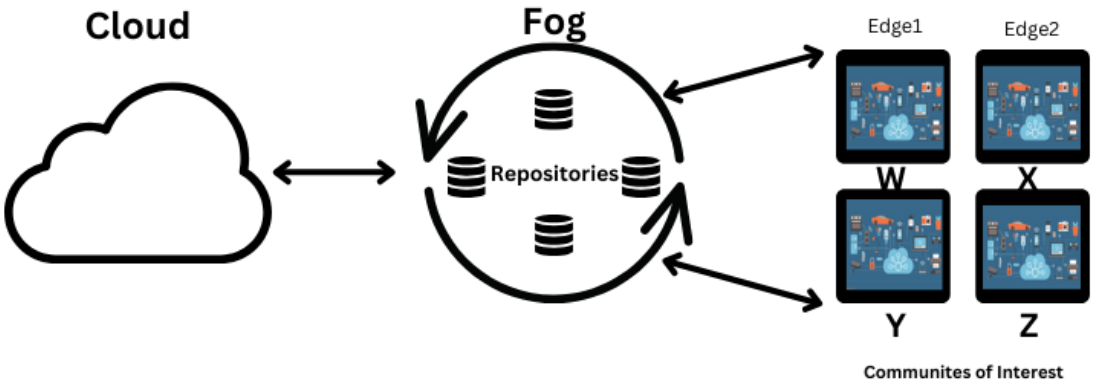


Figure 1.1.: Overall architecture of the project

by defining geospatial zones based on the inherent similarity of IoT data streams and location, and utilizing these zones to effectively design and validate machine learning models tailored to the unique characteristics of each geospatial zone. The experiments will involve the use of ML models to forecast temperature changes, conducted through simulations using real-world data from weather sensor networks. The aim of these experiments is to demonstrate the efficacy and relevance of this approach in addressing real-world challenges and advancing the fields of IoT and ML.

1.5 Document Structure

This master thesis report is divided into five chapters. Chapter 1, which is the current chapter, explains the motivation behind this work, the goals of the study, and the research questions.

Chapter 2 explains the background, giving contextual information and relevant concepts that set the stage for the research presented in this thesis. It provides an overview of topics related to this work, which are *IoT and ML*, *Social Internet of Things and Communities of Interest*, *Time Series – Similarity Analysis*, *Time Series – Clustering*, *Cluster Quality Analysis* and *Time Series Forescasting*

Chapter 3, explores the literature on existing work related to IoT, CoI, and Time Series Similarity Analysis and Clustering, as well as its applications.

Chapter 4 is the Methodology describing the systematic approach or framework used to conduct this research. It encompasses the principles, procedures, and techniques employed to gather data and analyze information.

Chapter 5 provides details about the implementation and development stage of the methodology. It includes code snippets to illustrate key components of the analysis and provided explanations for each snippet.

After that, Chapter 6 shows the development and experimental results of similarity analysis and clustering and the results of the ML model crafting, validation, testing and sharing. Followed by *Conclusions and Future Work*, the last section concludes the work and summarises the results, as well as suggests future development.

2. Background

This chapter provides an overview of the background and context that form the foundation of this thesis research. It delves into the historical, theoretical, and practical aspects that have shaped the subject matter of this study, laying the groundwork for the subsequent chapters.

2.1 IoT and ML

As technology has advanced, the exponential increase in the number of IoT devices has contributed substantially to the global surge in data production. By Karie et al. [2020], the number of connected devices in the IoT ecosystem is projected to reach 38.6 billion by 2025 and an estimated 50 billion by 2030. This proliferation of IoT devices not only has diversified the sources of data but has also significantly magnified the scale of data generated. From smart home devices and industrial sensors to wearable gadgets and connected vehicles, the IoT landscape spans a myriad of sectors, resulting in a diverse array of data sources. This surge not only encompasses building information models, parking transactions, and public transport transactions but also includes real-time health monitoring, environmental sensors, and more.

Despite this abundance, current applications often demonstrate a propensity to focus narrowly on specific use cases. As the IoT landscape continues to expand, addressing the challenge of effectively harnessing and leveraging this vast and diverse dataset for broader applications becomes increasingly crucial.

ML serves as a pivotal solution to the challenges posed by the influx of big data and the large datasets that emanate from the IoT. It is instrumental in harnessing the full potential of IoT by extracting meaningful insights and optimizing processes in the face of massive and diverse datasets [Adi et al., 2020].

By automating tasks such as feature extraction and dimensionality reduction, ML algorithms facilitate the processing and analysis of diverse and unstructured IoT data. Predictive modeling and anomaly detection enable proactive decision-making and identification of potential issues. Real-time processing, pattern recognition, and behavioral analysis capabilities allow immediate insights and adaptability in dynamic IoT environments [Adi et al., 2020].

In the world of CoI, the goal is to make devices more than just information carriers—they should be able to make decisions on their own. However, dealing with the diverse array of devices, purposes, and dynamic situations in the IoT makes it practically impossible to create and manage individual ML models for each device.

Similarly, creating a single ML model that can cover all IoT devices around the world is too complicated and unrealistic due to the wide variety and complexity of these devices and systems. A practical solution is to let the IoT device share ML models based on their similarities in purpose, domain, and context. This sharing can happen through a CoI, where devices with similar functions or environments team up. This simplifies the task of handling ML models and allows devices to collectively benefit from shared insights, making decision-making in the IoT world more effective and relevant [Bao et al., 2013].

This master thesis aims to take advantage of the CoI concept, where nodes or devices are grouped based on shared interests or purposes. The inspiration for this concept is drawn from the realm of the ***SIoT*** Atzori et al. [2011]. By applying the CoI approach, devices with similar functions or objectives can form collaborative communities, simplifying the complexity of managing and sharing ML models within the IoT ecosystem. This utilization of CoI not only streamlines decision-making processes among devices, but also enhances the efficiency of communication and collaboration within the IoT framework.

2.2 Social Internet of Things and Communities of Interest

The concept of SIoT comes from Atzori et al. [2011]. It introduces a social structure among objects in the IoT. Similar to how social networks connect individuals, SIoT aims to establish social relationships among objects. These relationships enable objects to interact, collaborate, and share information autonomously, without relying solely on human intervention.

CoI plays a significant role in the SIoT concept. Objects with similar characteristics and profiles can form communities based on shared interests, goals, or functionalities. These communities allow objects to exchange information, share best practices, and collaborate on solving common problems. For example, devices in the same local area network can establish social relationships to find solutions to common configuration issues. Similarly, objects visiting the same geographical area can form friendships to exchange useful information about the physical world.

By forming CoI, objects in SIoT can leverage collective intelligence and benefit from the knowledge and experiences of other objects with similar characteristics. This improves the general functionality, efficiency, and effectiveness of the IoT ecosystem.

In the field of IoT, characterized by numerous heterogeneous devices, varied purposes, and dynamic contexts, the individual creation and management of specific

models for each IoT device poses significant challenges. Given the intricate and diverse nature of IoT devices and ecosystems, developing a universal ML model capable of encompassing all IoT devices worldwide becomes an impractical endeavor. Each IoT device comes with its unique characteristics and functionalities, making a one-size-fits-all model unfeasible. To address this, a viable solution involves sharing of ML models among IoT devices that exhibit similarities in purpose, domain, and context.

2.3 Time Series – Similarity Analysis

Time series data represent a continuous flow of information generated by measuring various attributes such as sales, temperature, stocks, etc., at regular intervals. These data sets are typically indexed chronologically, making them valuable for applications such as weather forecasting, econometrics, earthquake prediction, signal processing, and other fields where analysing patterns and trends over time is crucial.

According to Esling and Agon [2012] a Time series T is an ordered sequence of n variables with real value.

$$T = (t1, ..., tn), ti \in R \tag{2.1}$$

Time series similarity measures are essential for tasks such as mining, retrieval, classification, and clustering. The goal is to determine to what degree a given time series resembles another one, which can provide valuable insights and enable various applications in different domains. According to Serrà and Arcos [2014] it is a core part of many systems, including case-based reasoning systems. Case-based reasoning (CBR) systems are a type of AI system that solves new problems by reusing solutions from similar past problems. CBR systems are particularly useful in domains where explicit rules or algorithms may be difficult to define or where there is a lack of complete domain knowledge Serrà and Arcos [2014]. They excel in situations where past experiences and solutions can be leveraged to solve new, similar problems. It has been successfully applied in various domains, including healthcare, engineering, finance, and customer support, among others.

According to Kianimajd et al. [2017] similarity in time series can be gauged based on the following aspects:

- Value: Time series exhibit similarity when the values of the Analysis Variable are approximately equal across time. For instance, a time series with values (1, 0, 1, 0, 1) is more similar to another time series with values (1, 1, 1, 1, 1) than it is to a series with values (10, 0, 10, 0, 10) due to the closer resemblance in values.

- Profile (Correlation): Time series display similarity if their values demonstrate synchronous increases and decreases, maintaining a roughly proportional relationship over time. For instance, a time series with values (1, 0, 1, 0, 1) is more akin to a time series with values (10, 0, 10, 0, 10) than it is to a time series with values (1, 1, 1, 1, 1) because of the correlated trends in their fluctuations.
- Profile (Fourier): Time series share similarity if they feature analogous smooth, periodic patterns in their values over time. These periodic patterns, also known as cycles or seasons, represent the durations of a pattern that repeats in subsequent periods. For example, businesses may observe recurrent patterns in total weekly sales, with the period starting on Monday and ending on Sunday.

In the investigation conducted by Abanda et al. [2019], there is a best distance measure for each case in theory and according to Aghabozorgi et al. [2015] the most prevalent shape-based methods commonly used to measure similarity in time-series clustering are ***Euclidean distance*** and ***Dynamic Time Warping***

Several research articles argue that the Euclidean distance metric is not well suited for time-series analysis. In summary, it is criticized for being insensitive to time shifts, essentially overlooking the temporal dimension of the data. If two time series are strongly correlated but one is shifted by just one time step, the Euclidean distance may inaccurately measure them as being more distant. One particular article by Keogh and Ratanamahatana [2005] contends that DTW serves as a significantly more robust distance measure for time series, enabling similar shapes to match even when they are out of phase along the time axis.

In their investigation, Zhu et al. [2018] delve into an examination and comparison of clustering techniques based on raw data and features. Their evaluation reveals that the feature-based clustering approach enhances the performance of time series models, particularly those integrating information from analogous time series and weather data. In particular, the use of ***Spearman correlation*** within the feature-based method consistently yields superior results on diverse metrics. The authors recognize the effectiveness of the feature-based approach in clustering and emphasize subsequent experiments centered on this method to further validate its performance.

2.3.0.1. Dynamic Time Warping

DTW stands out as a robust method in subsequence time series clustering due to its ability to compare time series data. It is described as an elastic measure that is well-suited for dealing with temporal drift and capturing similarity in shape, where the time of occurrence of patterns is not important Aghabozorgi et al. [2015]. DTW's key functionality involves aligning two time series, minimizing discrepancies by creating a warping path that traverses the distance between the corresponding

points in the series. In particular, the measurement of Euclidean distance is integral to this alignment process.[Zolhavarieh et al., 2014]

One of the main strengths of DTW lies in its ability to accommodate variations in the length and pace of time series, making it a potent tool for assessing similarity between subsequences in the context of subsequence time series clustering. The method's versatility in handling diverse temporal patterns contributes to its widespread use in this specific domain.

In comparison to Euclidean distance algorithms, the paper Zolhavarieh et al. [2014] emphasizes efficiency, highlighting its ability to quickly search and mine extensive time series datasets and in terms of time-series classification accuracy [Aghabozorgi et al., 2015]. This efficiency factor positions DTW as an invaluable asset for the analysis and clustering of massive time-series datasets.

The DTW between sequences x and y is formulated as the following optimization problem:

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{(i,j)\in\pi} d(x_i, y_j)^2} \tag{2.2}$$

where $\pi = [\pi_0, \ldots, \pi_K]$ is a path that satisfies the following properties:

- It is a list of index pairs $\pi_k = (i_k, j_k)$ with $0 \leq i_k < n$ and $0 \leq j_k < m$.
- $\pi_0 = (0, 0)$ and $\pi_K = (n - 1, m - 1)$.
- For all $k > 0$, $\pi_k = (i_k, j_k)$ is related to $\pi_{k-1} = (i_{k-1}, j_{k-1})$ as follows:
  - $i_{k-1} \leq i_k \leq i_{k-1} + 1$
  - $j_{k-1} \leq j_k \leq j_{k-1} + 1$

the equation 2.2 from:tslearn documentation depicts a DTW optimization problem.

To sum it up, DTW is computed by taking the square root of the sum of squared distances between each element in the time series X and its corresponding nearest point in the time series Y.

Consider two distinct curves, one red and the other blue, as shown in Figure 2.1 above, characterized by varying lengths. Despite both curves sharing a common pattern, the longer blue curve poses a challenge when employing a one-to-one Euclidean match, resulting in a misalignment where the tail of the blue curve is omitted (as depicted on the right). DTW addresses this issue by introducing a one-to-many matching approach. This ensures a precise alignment of the shared pattern,

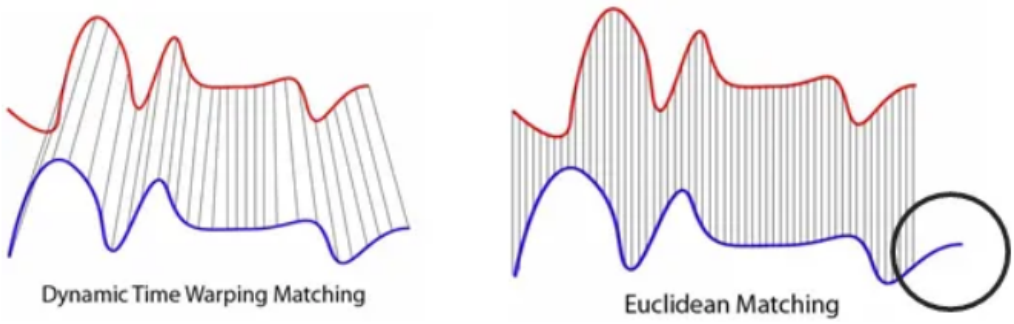


Figure 2.1.: Differences between DTW and Euclidean matching Source:Wikimedia Commons

eliminating any omission in both curves (illustrated on the left, Figure 2.1).

2.3.1 Spearman's Rank Correlation

Spearman's rank correlation assesses the intensity and direction of the connection or relationship between two variables that have been ranked. According to Gauthier [2001] it essentially gauges the degree of monotonicity in the relationship between these variables, indicating how effectively their association can be depicted using a monotonic function. See equation 2.3 [Gauthier, 2001].

$$\frac{r_s = 1 - 6\sum_{i=1}^{n} d_i^2}{n^3 - n} \tag{2.3}$$

where,
rs = Spearman Correlation coefficient
di = the difference in the ranks given to the two values for each item of the data
n = total number of observation

The Spearman Rank Correlation can take a value from $+1$ to -1 where,
A value of $+1$ means a perfect association of rank
A value of 0 means that there is no association between ranks
A value of -1 means a perfect negative association of rank

Spearman's rank correlation coefficient offers several advantages compared to the more commonly used Pearson's correlation coefficient. Being a non-parametric technique, it remains unaffected by the underlying data distribution. Since it operates on data ranks rather than actual values, it demonstrates robustness against outliers, and there is no requirement for data to be collected at regular intervals. In addition, it is applicable even with small sample sizes and is straightforward to implement. However, a drawback is the loss of information when converting data to ranks, and when dealing with normally distributed data,the corresponding statistical test for in-

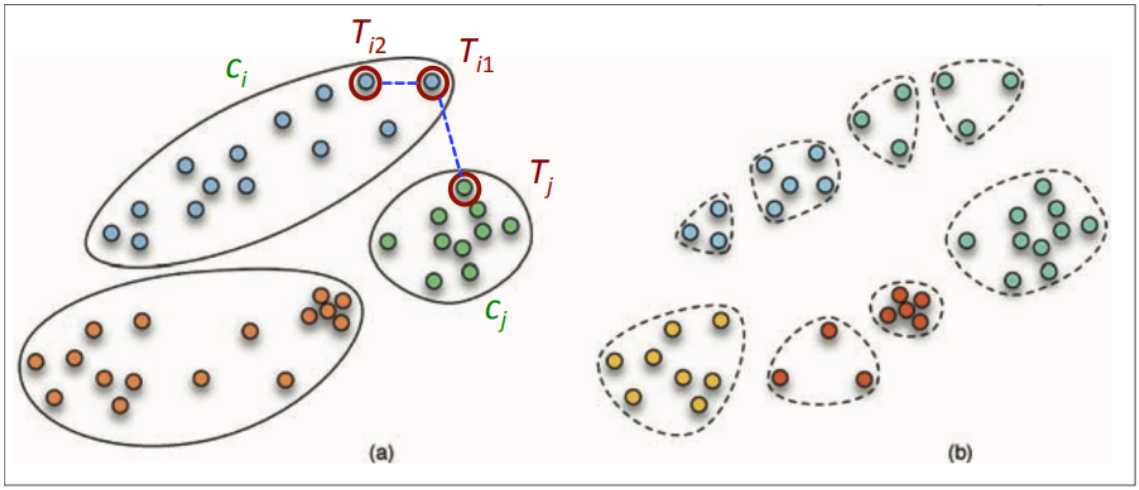
dependence may be less powerful than the Pearson correlation coefficient [Gauthier, 2001].

2.4 Time Series – Clustering

Clustering is classified as an - unsupervised learning problem that revolves around identifying similarities among various data points and grouping them -together. The primary objective is to organize the data points into clusters, where those within the same group share more similarities with each other than with those in other groups. This technique is a fundamental aspect of exploratory data mining and finds applications in diverse fields such as bioinformatics, pattern recognition, image analysis, machine learning, and more [Esling and Agon, 2012].

Given a time-series database DB and a similarity measure $D(Q, T)$, find the set of clusters $C = \{c_i\}$ where $c_i = \{T_k | T_k \in DB\}$ maximizes the distance between clusters and minimizes the variance within clusters.

More formally, for all i_1, i_2, j such that $T_{i_1}, T_{i_2} \in c_i$ and $T_j \in c_j$, it holds that $D(T_{i_1}, T_j) \gg D(T_{i_1}, T_{i_2})$.



(a) N = 3 and (b) N = 8

Figure 2.2.: Visual depiction of number of clusters when N is three and when N is eight respectively. Source:Esling and Agon [2012]

Figure 2.2 is a visual illustration of two different scenarios of the clustering equation when the number of clusters is three and when the number of clusters is eight.

2.4.1 Taxonomy of Time series Clustering

Upon reviewing the existing literature, it becomes apparent that many works related to clustering time-series can be categorized into three main groups: whole time-series clustering, subsequence clustering, and time point clustering [Keogh and Lin, 2005].

- ***Whole time-series clustering*** -involves clustering a collection of individual time-series based on their similarity. In this context, clustering entails the application of conventional (typically) clustering methods on discrete objects, where these objects are the time-series themselves [Esling and Agon, 2012].
- ***Subsequence clustering*** - entails clustering a set of subsequences extracted from a time-series using a sliding window. This method involves clustering segments derived from a single extended time-series [Esling and Agon, 2012].
- ***Time points clustering***- represents another category found in Ultsch and Morchen. It involves clustering time points based on a combination of their temporal proximity and the similarity of corresponding values. This approach shares similarities with time-series segmentation; however, it differs in that not all points need to be assigned to clusters, meaning that some may be treated as noise [Esling and Agon, 2012].

In essence, subsequence clustering is executed on an individual time series, but according to Keogh and Lin [2005] this form of clustering lacks significance. Similarly, time-point clustering is implemented on a solitary time-series and bears resemblance to time-series segmentation. The aim of time-point clustering is to identify clusters of time points rather than clusters of entire time-series data.

2.4.2 Temporal and Spatial Clustering

According to Esling and Agon [2012] there are three primary approaches to cluster time series: shape-based, feature-based, and model-based, as shown in Figure 2.3 below.

In the ***shape-based approach***, the focus is on matching the shapes of two time-series through non-linear stretching and contracting of the time axes. This method, often termed a raw-data-based approach, operates directly on raw time-series data. Shape-based algorithms typically utilize conventional clustering methods, adapted for time-series with modified distance/similarity measures.

The ***feature-based approach*** involves converting raw time-series into lower-dimensional feature vectors, followed by applying a conventional clustering algorithm to these extracted feature vectors. Typically, equal-length feature vectors are computed from each time-series, and the Euclidean distance is often used for measurement.

In the ***model-based methods***, raw time-series are transformed into the model parameters (parametric models for each time-series). A suitable model distance and a clustering algorithm, often conventional, are chosen and applied to the extracted model parameters[Warren Liao, 2005]. However, model-based approaches have been

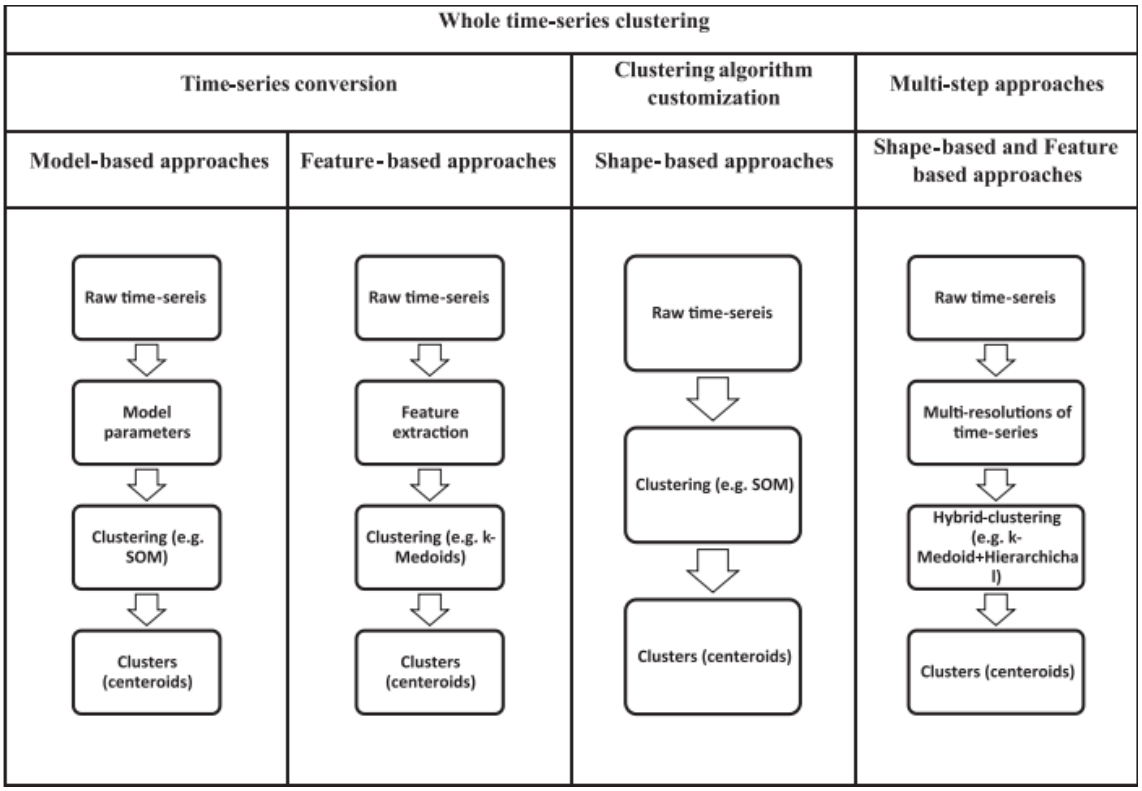


Figure 2.3.: Time series clustering approaches. Source: Esling and Agon [2012]

shown to face scalability issues [Aghabozorgi et al., 2015], and their performance tends to decrease when clusters are in close proximity to each other [Aghabozorgi et al., 2015].

For this master thesis since we will be dealing with (dynamic) temperature time series data, both shape-based and feature-based approaches will be applied and compared to discover which of these approaches best captures hidden patterns and relationships in raw time series data Warren Liao [2005]. The distance or similarity measure used could be based on the differences between temperature values at different time points. This is because shape-based and feature-based methods excel differently in capturing patterns and variations in time series data.[Aghabozorgi et al., 2015].

2.4.3 Time-Series Clustering Algorithms

Time-series data mining encompasses diverse tasks such as clustering of whole time series, anomaly detection, motif discovery, and query by content [Esling and Agon, 2012]. To address these tasks, various techniques such as K-means clustering, Self-Organizing Maps (SOM), Hidden Markov Models (HMM), Support Vector Machines (SVM), and decision forests are employed. Each algorithm exhibits unique strengths and limitations, and the selection of a particular algorithm is contingent on the specific demands and features of the time-series data under analysis.

The authors of Keogh and Lin [2005] point out that in existing literature, among clustering algorithms, K-means are widely used and deemed more effective than recently proposed distance measures for time series. However, they emphasize certain drawbacks associated with K-means, including its heuristic nature and susceptibility to the initial selection of centres.

Some researchers [Javed et al., 2021] explore the benefits of opting for the Kohonen self-organizing map in contrast to the K-means clustering algorithm. They claim that while K-means stands out as a widely acknowledged and superior clustering technique, it falls short in terms of interpretability and visualization capabilities when compared to SOM. SOM, in contrast, is distinguished by its unique data visualization features and is frequently employed for tasks such as data visualization, dimensionality reduction, and feature selection. However, it is highlighted that if the sole requirement is for classification or hard clusters, then K-means serves as a faster and equally accurate clustering algorithm.

In reference to the literature reviewed, this master's thesis plans to utilize the K-means algorithm for clustering time-series data. Following the application of this algorithm, the Silhouette score of the clustering outcomes from the similarity analysis data of both the DTW and Spearman Correlation method will be assessed and compared.

The cluster groups with the highest silhoutte score will be used to conduct further experiments in this Master Thesis. This will involve sharing ML models trained on sensor data from one station in each cluster group and then tested on each of the other clusters to assess the ML model performance in at least one sensor in each of the other clusters.

2.4.3.1. K-Means Algorithm

The K-means algorithm is a data mining technique that employs clustering methods. Clustering entails the segmentation of a dataset into clusters or groups sharing comparable traits. This algorithm operates through iterations, endeavoring to organize the dataset by assigning each data point to the closest cluster centroid Melo Riveros et al. [2019].

The objective function of K-means is to minimize the sum of the squared Euclidean distances between each data point and its nearest cluster center, also known as square-error distortion [Bação et al., 2005]. Input parameters include the preferred number of clusters and initial centroids, with the algorithm producing the ultimate centroids. Each data element is grouped into a cluster according to its similarity to the cluster centroid. The primary objective of the K-means algorithm is to minimize the distance between data points and their designated cluster cen-

troids [Melo Riveros et al., 2019].

2.5 Cluster Quality Analysis

There are various methods available to assess the quality of the clustering. These methods can be broadly classified into two groups based on the availability of ground truth.

Extrinsic methods, or ***supervised methods***, utilize the ground truth when available, comparing the clustering results against this known truth. On the other hand, ***Intrinsic methods***, or ***unsupervised methods***, assess the quality of clustering by examining the separation between clusters in the absence of ground truth.

Ground truth, in this context, serves as a form of supervision represented by ***cluster labels.*** Therefore, metrics for supervised algorithms, such as accuracy, R-square value, sensitivity, and specificity, are commonly used to evaluate their goodness of fit.

When employing unsupervised machine learning techniques and incorporating clustering algorithms such as K-Means, DBSCAN, or HDBSCAN,Esling and Agon [2012] the actual or true labeling of a dataset is unavailable. In this case, intrinsic methods are employed to assess the quality of clustering. Typically, these methods gauge the effectiveness of clustering by analyzing the degree of separation and compactness of the clusters. Inherent to many intrinsic methods is the utilization of a similarity metric to measure the relationships between objects within the dataset.

Several scholarly (Ashari et al. [2023]),(Shi [2021]), sources propose several techniques to assess the effectiveness of the K-means clustering algorithm in achieving optimal results. These methods include the Elbow, Silhouette, Davies-Bouldin, Calinski-Harabasz, and Rand-Index, along with the gap statistic method. These techniques serve as validation measures to ensure the quality of the clusters generated by the algorithm.

In their work, Aksan et al. [2021] highlights the diverse applications of evaluation measures. They specifically advocate employing the elbow method for determining the optimal number of clusters, whereas the silhouette coefficient and Calinski–Harabasz index are recommended as metrics for evaluating clustering performance.

In Shi [2021], the Elbow method and the Silhouette method stand out as the most prominent and widely recognized approaches. The Elbow method, considered the oldest, is mentioned to estimate the potential optimal cluster number. However, it is acknowledged to have subjective aspects due to the ambiguity of identifying the elbow point. The Silhouette method on the other hand is noted for its effective

performance in estimating the potential cluster quality.

In the perspective of Agoun et al., the elbow method focuses on evaluating the cohesion of the cluster, specifically assessing the position of the data points within a cluster. On the other hand, the silhouette score is highlighted as offering a more comprehensive assessment of cluster quality. This is achieved by considering both cohesion, which measures the proximity of data points within a cluster, and separation, which measures the distance between data points in one cluster and those in the nearest cluster.

Due to the distinct functionalities of these metrics, this master thesis aims to employ both the elbow method and the silhouette score to thoroughly evaluate cluster quality.

2.5.1 Silhoutte Score

The Silhouette method, a widely acknowledged technique with commendable efficacy to determine the optimal number of clusters, evaluates the clustering results based on the average distances between individual data points within the same cluster and the average distances between different clusters. According to Rousseeuw [1987] the key metric employed in this method is the silhouette coefficient (S), denoted by

$$\frac{b-a}{max(a,b)} \tag{2.4}$$

where :
a - signifies the mean intra-cluster distance,
b - represents the mean nearest-cluster distance.

The silhouette coefficient (S) is defined within the range of $-1 \leq S \leq 1$. A higher S value, closer to 1, signifies superior clustering, while proximity to -1 suggests that the sample may be more appropriately assigned to an alternative cluster. Notably, the Silhouette method is preferred for estimating the potential optimal cluster number. The silhouette index, derived from this method, proves effective in determining the optimal number of clusters across diverse scenarios.

2.5.2 Elbow Method

The Elbow method is regarded as the earliest approach to estimate the optimal cluster number for utilization in a K-means clustering algorithm [Shi et al., 2020]. The rationale behind the Elbow method is that as the number of clusters increases, the WCSS tends to decrease, leading to more compact clusters. The process involves plotting a curve of SSE against each cluster number, and experienced analysts are

tasked with estimating the optimal elbow point based on their curve analysis. However, at a certain point, adding more clusters provides diminishing returns in terms of reducing WCSS.

The Elbow point is where the reduction in WCSS slows down, indicating a balance between cluster compactness and avoiding overfitting. However, if the SSE curve exhibits a smooth pattern, analysts may struggle to clearly identify the Elbow, rendering the Elbow method less effective in determining the optimal cluster number [Shi et al., 2020]. The obtained cluster number through the Elbow method is subjective and visual, lacking a quantitative metric to explicitly indicate the optimum elbow point [Shi et al., 2020].

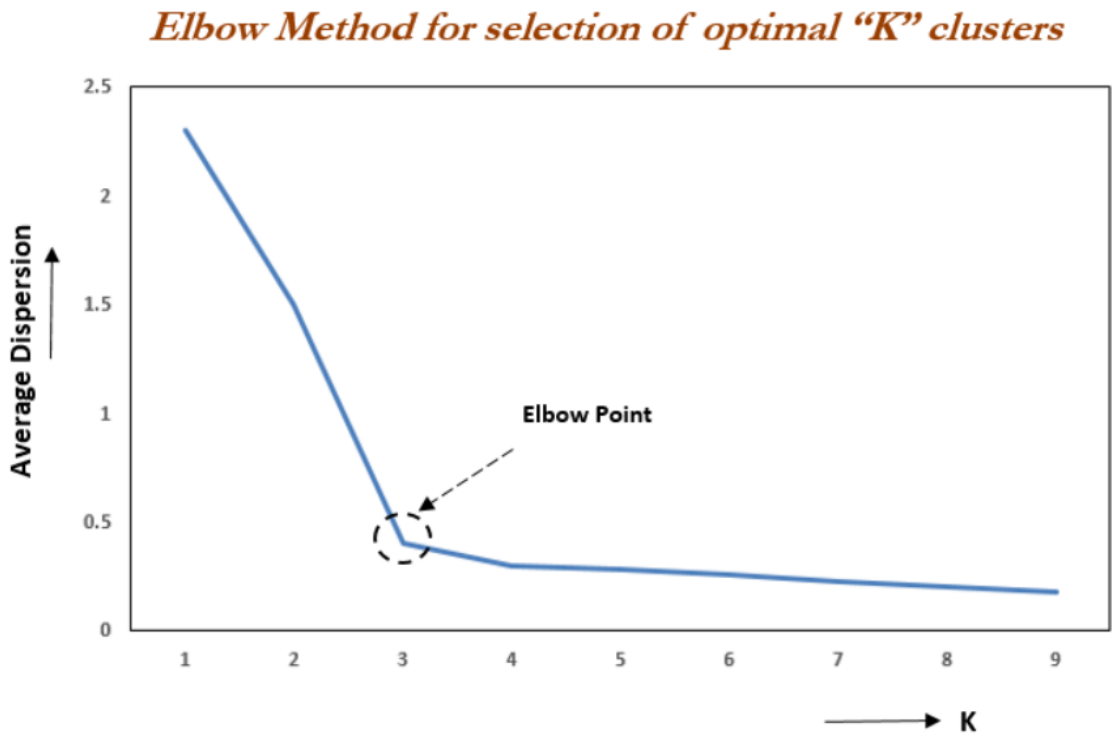


Figure 2.4.: Elbow plot revealing optimal cluster count. Source :Oreilly.com

2.6 Time Series Forescasting

Time series forecasting is a branch of predictive analytics that focuses on predicting future values based on past observations of a time-dependent variable. In a time series, data points are ordered chronologically, and the goal of forecasting is to make predictions about future values or trends. It is widely used in various fields, including finance, economics, weather forecasting, inventory management, and more.

Deterministic and statistical models have been created to predict and forecast outcomes over time, using multivariate metrics for variable selection. However, the intricate nature of the relationship among IoT sensor data poses challenges for straightforward inference [Bogado Machuca et al., 2023]. In addition, IoT devices

vary, with certain sensors experiencing faults that result in delayed or even missing data transmission. This complexity has led to the emergence of competitive alternatives in the form of data-driven methodologies, such as ML and deep learning, as they are better equipped to handle the complexities associated with IoT data [Bogado Machuca et al., 2023].

This master's thesis will employ conventional ML techniques, specifically Random Forest Regression, and a Deep Learning approach,LSTM namely Long Short-Term Memory, for predicting temperature time series values. The objective is to demonstrate that grouping time series data by assessing the similarity of incidence time series, enhances the overall performance of the forecasting models Bogado Machuca et al. [2023].

2.6.1 Long Short Term Memory

The LSTM is a type of recurrent neural network (RNN) architecture. LSTM models, are specifically tailored for time series forecasting, leveraging memory cells, which effectively capture both long and short dependencies Bogado Machuca et al. [2023]. LSTM cells consist of input (i), forget (f), and output (o) gates, playing roles in incorporating new information into the cell state (C), discarding less relevant information from memory, and controlling the output prediction (h). Recurrent Neural Networks such as LSTM utilize sequential information, where the output is influenced not only by current inputs but also by preceding ones. For instance, the input at a given point x_t is a value $x_t - n$ from the same series, with n representing the look-back period. The collaborative function of these gates allows the network to learn and store information, both short-term and long-term, pertinent to the sequence.

The computation of LSTM cell states is performed as follows Bogado Machuca et al. [2023]:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{2.5}$$

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{2.6}$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{2.7}$$

$$\hat{C}_t = \tanh(W_C h_{t-1} + U_C x_t + b_C) \tag{2.8}$$

$$C_t = ft \odot C_{t-1} + it \odot \hat{C}_t \tag{2.9}$$

$$h_t = ot \odot \tanh(C_t) \tag{2.10}$$

where : W , U and b_q are weights matrices and bias respectively, the subscript q can be either for input gate i, output gate o, forget gate f, or memory cell c

depending on what is being calculated. The $\odot$ symbol represents the Hadamard entrywise product. Vectors i_t, f_t, and o_t denote the input, forget, and output gates, respectively. Vector C_t represents the current cell state, and vector $\hat{C}$ represents the new candidate value for the cell state.The function $\sigma(\cdot)$ is a Sigmoid function and modulates equations 2.4 to 2.6 between 0 and 1.

2.6.2 Random Forest

The random forest is a ML technique that employs a set of decision trees to enhance flexibility, accuracy, and accessibility of the output. It combines Breiman's bagging sampling approach and random selection of features to construct a collection of decision trees with controlled variation. Each decision tree in the ensemble is constructed using a sample based on training data replacement, and the class label of an unlabeled instance is determined by majority voting of the classifiers.Fawagreh et al. [2014]

It outperforms the decision tree algorithm, which tends to have a lower accuracy. In essence, the random forest method boosts the effectiveness of decision trees, serving as an excellent algorithm capable of handling both classification and regression tasks. As a supervised learning algorithm, the random forest utilizes the bagging method within decision trees, which leads to improved model accuracy.

2.6.3 Perfomance Evaluation

Perfomance evaluation metrics are crafted to gauge the model's proficiency in predicting future values, providing quantitative insights into its accuracy and ability to capture inherent patterns and trends within the data. Evaluation metrics for time series models serve to measure the model's overall performance and effectiveness in forecasting.

RMSE is an extended form of MSE (Mean Squared Error) and is a widely adopted technique for assessing model performance. It serves to measure the dispersion of data points around the optimal line, representing the standard deviation of the mean squared error. A lower RMSE value signifies that the data points are closer to the best-fit line, indicating improved model performance.[Chai and Draxler, 2014]

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \tag{2.11}$$

where:

$\hat{y}_i$: Predicted value for observation i

y_i : Actual value for observation i

N : Number of observations

3. Related Works

This section categorizes the literature reviews into distinct thematic areas, namely *Social Internet of Things and Communities of Interest*, *Time Series Similarity Analysis and Clustering*, *Clusted-Based Time Series Forecasting* and *Machine Learning Model Testing and Sharing*.

3.1 Social Internet of Things and Communities of Interest

In the realm of IoT, how objects relate to each other has been a topic in research. One study by Kosmatos Kosmatos et al. [2011] explores social networks-like relationships, exploring various architectural models of IoT such as RFID, smart objects, and social perspectives. In this framework, objects can join communities, create interest groups, and work together. However, the paper does not detail the methods for establishing these desired social networks among objects.

Another group of researchers focuses on IoT object relationships by introducing Social Virtual Objects SVOs, which represent real-world objects RWOs in an edge-cloud environment. These researchers propose a platform called SVOR Farris et al. [2015] (Social Virtual Object Root) which serves as a central coordinator in the proposed platform, facilitating the dynamic definition of IoT areas or CoI. Its key role lies in managing social relationships among Social Virtual Objects SVOs in the edge-cloud. SVOR dynamically establishes and adapts social connections among SVOs based on varying permission levels, responding to changing interactions within the IoT. By efficiently addressing resources and providing access keys, SVOR enables effective interaction and service provision among IoT objects.

In an alternative study, the concept of friendship selection is introduced within the context of the SIoT. This addresses the challenge posed by the abundance of objects in the IoT by developing strategies to improve network navigability through the selective establishment of links Nitti et al. [2015]. The study suggests that the principles derived from friendship selection and enhanced network navigability could be applied to dynamically group IoT objects for collaborative purposes.

Another paper by Misra et al. [2012] introduces a method for detecting communities within an integrated architecture of the IoT and Social Network SN. Employing a graph mining approach to address complexities within the network, the proposed scheme, known as Community Detection in an Integrated IoT and SN -(CDIISN), categorizes nodes/actors in complex networks into fundamental nodes and IoT nodes, subsequently applying a community detection algorithm. Addition-

ally, the paper explores the significance of community detection within an integrated setting and discusses potential future applications of this approach.

Yue et al. [2014] suggests adopting a community-centric strategy to cluster IoT devices into communities. The criteria for organizing these IoT devices revolve around their shared interests in data and information. Users with comparable interests are assembled into communities, fostering streamlined data dissemination and sharing for enhanced efficiency.

The paper by Barthwal et al. [2013] introduces two graph clustering algorithms designed for community detection within integrated IoT and social networks. The first algorithm is tailored for undirected graphs like those found on Facebook and Google+, while the second algorithm is specifically designed for directed graphs, such as those on Twitter. Both algorithms work by categorizing nodes into basic and IoT nodes and utilize the Community Detection in Integrated IoT and the SN-CDIISN algorithm to identify clusters of nodes where the connections within a cluster are denser than connections between clusters. The algorithms incorporate the concept of friendship in social networks, taking advantage of the metric of mutual friends for community extraction. After identifying communities, the algorithms take into account an access control policy based on these communities, determining resource sharing among nodes. In essence, the proposed graph clustering algorithms focus on recognizing communities in integrated IoT and social networks, emphasizing the role of friendship and mutual connections among nodes.

The primary objective of Lianhong Ding et al. [2010] is to propose a platform that combines the Internet, the IoT and social networks, in order to promote the advancement of IoT and social networking. The platform is designed to enable scientists to analyse the behaviors of both objects and individuals as data. Although the paper lacks specific details on the methodology or algorithms employed for clustering, it highlights the integration of the Internet, IoT, and social networks into a unified virtual platform. This integration is intended to establish meaningful relationships among information, objects, and people, although the specific clustering techniques are not explicitly outlined in the provided excerpts.

Aldelaimi et al. [2020] proposed a Dynamic Community of Interest Model (DCIM) that enables IoT objects to socialize and form communities based on common interests. The model allows objects to join or leave communities, with criteria such as the number of objects allowed, object types, and duration of membership. The DCIM model has been evaluated through simulation, demonstrating its effectiveness in detecting common interests and facilitating the formation of dynamic communities. The model provides a framework for IoT objects to build communities and enhance interactions in various scenarios, such as smart cities, smart homes, and

universities.

Researchers in Atzori et al. [2011] paper delve into the creation of social connections among objects within the SIoT. The paper highlights that objects can form relationships through shared physical proximity and collaborative activities, mirroring the way humans build connections through shared experiences. These relationships are categorized into location-based application profiles or situation-based application profiles. Furthermore, the paper proposes the exchange of social profiles between devices possessing similar characteristics, enabling them to autonomously share knowledge and address challenges. It continues to suggest that the social structure within SIoT can be shaped to ensure effective navigation of the network and facilitate the discovery of objects and services. This parallels the functionality of human social networks. Notably, the paper does not explicitly mention the grouping of devices into CoI or dynamic areas.

In Shahab et al. [2022] the authors discuss coordination strategies in a collaborative system where social entities, places, and data interact seamlessly. This implies that the grouping and coordination of IoT devices within a social network framework can facilitate more efficient and effective interactions and collaborations. The paper does not provide specific technical implementations or algorithms for device grouping, but recognizes the importance of social relationships and coordinated interactions among devices in the SIoT context.

The paper by Bao et al. [2013] is mainly focused on the development and assessment of a trust management protocol for the CoI-based SIoT system. Although it notes the dynamic nature of nodes, allowing them to join and depart while forming CoIs it does not delve into the intricate details of node functionalities.

Each node in the system is identified by a distinct address, and the formation of CoIs is dependent on the social networks of entity owners, where nodes sharing common interests and robust social bonds become part of the same community. It elaborates how the nodes within the same CoI can establish trust agreements due to shared interests but refrains from providing explicit insights into the specific mechanisms employed by nodes to organize themselves within the CoI. In essence, the paper centres its attention on the trust management protocol design and evaluation, leaving the detailed characterization and organizational aspects of the nodes within the CoI less expounded.

The paper by Achtaich et al. [2018] outlines the process of fleet formation, depicting fleets as Dynamic Software Product Lines (DSPL) operating at the domain level. Described as unique products within this framework, each fleet shares common characteristics with others while tailoring its features to meet the specific demands of the served customer base. Fleet formation involves the creation of assets and the

derivation of features, aligning with individual customer requirements.

The same paper underscores the dynamic nature of fleets, emphasizing that their composition is subject to change over time. Devices forming a fleet in one configuration may not necessarily persist in the same capacity in subsequent fleet instances. Some devices may transition to being part of the contextual landscape, while others may assume roles in the broader environment.

The table provided below 3.1 offers a comprehensive overview of different community formation techniques. This comparative analysis serves as a valuable resource for researchers and practitioners seeking to design and deploy IoT systems capable of forming robust and adaptable CoI tailored to specific application requirements.

Table 3.1.: Comparison of CoI Concepts

SIoT Work	**CoI Concept**	**Comparison with Thesis**
Kosmatos et al. [2011]	Objects can join communities, create interest groups, and work together, but lacks detailed methods for establishment.	Unlike Kosmatos et al., this study provides a structured framework for CoI formation, potentially leading to more efficient collaboration and resource sharing among objects.
Farris et al. [2015]	Introduces Social Virtual Object Root (SVOR) for managing social relationships among objects.	Unlike Farris et al., this study emphasizes data-driven criteria for CoI formation, potentially leading to more effective utilization of resources and improved model sharing.
Nitti et al. [2015]	Introduces friendship selection to improve network navigability, suggesting dynamic group formation.	Unlike Nitti et al., this thesis offers a systematic approach for forming CoIs based on objective criteria, potentially leading to more efficient model sharing across IoT devices.

Continued on next page

Table 3.1 – continued from previous page

SIoT Work	CoI Concept	Comparison with Thesis
Misra et al. [2012]	Proposes CDIISN for detecting communities within integrated IoT and Social Network, without detailed methods.	Unlike Misra et al., this study offers a streamlined approach for CoI formation, potentially avoiding the complexity of integrated network architectures.
Yue et al. [2014]	Suggests a community-centric strategy based on shared interests, without detailed methods for establishment.	Unlike Yue et al., this thesis provides a systematic framework for CoI formation, potentially leading to more effective utilization of resources and improved model sharing.
Barthwal et al. [2013]	Introduces graph clustering algorithms for community detection, without specific integration with IoT.	This study focuses on data similarity and geospatial factors for CoI formation without detailed algorithms. Unlike Barthwal et al., this study offers a straightforward approach for forming CoIs, potentially avoiding the complexity of integrated network algorithms.
Lianhong Ding et al. [2010]	Proposes integrating IoT, social networks, and the Internet for meaningful relationships, lacking specific CoI methods.	Unlike Ding et al., this thesis provides a systematic framework for CoI formation, potentially leading to more efficient resource sharing among objects.
Aldelaimi et al. [2020]	Introduces DCIM for dynamic CoI formation based on common interests.	Unlike Aldelaimi et al., this provides a systematic framework for CoI formation, potentially leading to more effective model sharing across IoT devices.

Continued on next page

Table 3.1 – continued from previous page

SIoT Work	CoI Concept	Comparison with Thesis
Atzori et al. [2011]	Discusses creating social connections among objects through shared physical proximity and collaborative activities, without specific CoI methods.	This thesis focuses on data similarity and geospatial factors for CoI formation, providing a more structured approach compared to physical proximity. Unlike Atzori et al., this research offers a systematic framework for CoI formation, potentially leading to more efficient model sharing among IoT devices.
Shahab et al. [2022]	Discusses coordination strategies in SIoT, without specific CoI methods.	Unlike Shahab et al., this study provides a systematic framework for CoI formation, potentially leading to more efficient model sharing among IoT devices.
Bao et al. [2013]	Develops trust management protocol for dynamic CoI formation, but lacks detailed methods for CoI establishment.	Unlike Bao et al., this research offers a systematic framework for CoI formation, potentially leading to more efficient model sharing across IoT devices.
Achtaich et al. [2018]	Describes dynamic fleet formation at the domain level, without specific CoI methods.	Utilizing data similarity and geospatial factors for CoI formation, offering a more structured approach compared to fleet formation alone. Unlike Achtaich et al., this research provides a systematic framework for CoI formation, potentially leading to more efficient model sharing among IoT devices.

3.2 Time Series Similarity Analysis and Clustering

Several studies have utilized time series clustering to reveal the inherent patterns in their time series data. The researchers in Jastrzebska et al. [2022] introduce a novel concept-based strategy for assessing time series similarity and applying it to time series classification. The primary goal is to overcome the lack of transparency and interpretability observed in existing methods for evaluating time series similarity. The authors employ fuzzy sets to represent concepts and utilize linguistic labels to

express data in natural language.

The proposed approach entails the creation of global and local models for time series based on abstract concepts. Subsequently, the evaluation of similarity between time series involves a comparison of their respective local models. The authors substantiate the efficacy of their method by achieving highly favourable results in time series classification tasks, surpassing other contemporary approaches.

A group of researchers Xia et al. [2011] utilized an extended local similarity analysis (eLSA) method to examine time series data related to microbial communities and gene expression. This technique facilitated the detection of statistically significant local patterns and potentially time-delayed associations within both the microbial community and gene expression datasets.

In this paper Van Onsem et al. [2022], researchers introduce a streamlined anomaly detection method termed Hierarchical Pattern Matching (HPM), designed for real-time monitoring of device metrics to prevent downtime and data loss by continuously observing critical device metrics. HPM employs time series similarity analysis and establishes a lightweight hierarchical structure called a Time Series Identity Tree, enabling the retention of extensive metric history without necessitating a large memory footprint. During similarity analysis, incoming subsequences are compared with stored patterns in the Time Series Identity Tree. If a match is identified, it is considered normal behavior, and no anomaly is triggered. In cases where no match is found, signalling an anomaly, an alarm is activated.

Scholars in Wang et al. [2006] devised an approach to cluster time series data by focusing on their structural attributes, with the aim of improving the sensitivity to missing or noisy data while reducing dimensionality. The approach performs a similarity analysis by extracting overarching features from the time series, encompassing elements such as trend, seasonality, and periodicity. These features serve as the foundation for measuring the similarity between time series and establishing the groundwork for clustering.

The global features extracted are then inputted into clustering algorithms, including hierarchical clustering and self-organizing map SOM clustering. Similarity analysis, based on these extracted features, aids in the identification of meaningful clusters, effectively capturing the inherent characteristics of the time series data. The proposed method exhibits promising outcomes in terms of clustering accuracy and its ability to handle time series data of varying lengths.

Utilizing datasets sourced from the University of California Riverside archive for a time series similarity analysis incorporating three distinct distance measures, namely the Euclidean distance, the DTW and the shape-based distance. The researchers

in Javed et al. [2020] gauged the similarity among time series data to perform clustering. Their goal was to group time series data into meaningful clusters based on their similarity. This approach served to assess the effectiveness of various clustering methods and distance measures in the context of time series data.

The researchers in Razaque et al. [2022] conducted a time series similarity analysis by introducing an algorithm called the Novel Matrix Profile (NMP), which incorporates features from existing algorithms such as STAMP and STOMP. The NMP algorithm calculates and stores information for a search for all-pair similarity, offering utility in various data mining tasks. The results of the time series analysis using the NMP algorithm were then applied to enhance the efficiency and effectiveness of data mining tasks in the healthcare domain.

In Maurya et al. [2016], the authors suggest an enhanced algorithm that leverages grid correlation and attraction calculations for precise clustering of time-series data within the Smart Grid. This modification allows for the efficient analysis of extensive energy consumption data in the Smart Grid, ultimately leading to enhanced energy management, improved demand response strategies, and more effective appliance diagnostics.

Researchers in Bornemann et al. [2018], performed a time series similarity analysis by representing data changes as time series and clustering them based on their similarities. They introduce a transformation framework designed to aggregate sets of changes into numerical time series at varying resolutions. Time series clustering is then applied to the transformed data, utilizing different similarity measures and clustering algorithms. The outcomes of the clustering process are employed to unveil patterns, identify outliers, and offer insights across diverse domains. The paper illustrates the application of this framework in uncovering patterns in IMDB voting behaviour.

The paper by Alwan et al. [2022] uses a time series similarity analysis, using measures such as DTW, to compare the shapes or features of time series and identify malfunctioning sensor nodes. Based on the results of this analysis, the time series are then grouped into different clusters according to their patterns. The research shows that time-series clustering is adept at detecting both continuous and emerging faults in sensor nodes.

Scholars in Robinson et al. [2021] performs an analysis of time series similarity by comparing spectral values within a building footprint to those in the surrounding area across different time points. The results of the similarity analysis are clustered using k-means clustering on time series data derived from remotely sensed imagery. The combined approach of time series similarity analysis and clustering, as implemented in the Temporal Cluster Matching (TCM) model, facilitates the detection of

building changes by comparing spectral values inside and outside the footprint and pinpointing dissimilarities in their distributions relying on the assumption that the colours and textures of a developed structure will differ from those of its immediate surroundings.

Another paper by Bonacina et al. [2020] employs natural visibility graphs for time series similarity analysis, creating a weighted graph that represents signal similarities. Community detection algorithms are then applied to identify clusters of similar time series. The clustering results are leveraged for feature subset selection, effectively reducing the dataset's dimensionality by 74.4%. Applied to a cogeneration plant's condition monitoring system, the method outperforms standard time series clustering, providing insights into system behaviour, relationships between components, and enhancing information content about signal roles within the network. The approach proves valuable for diagnosing exceptional events, explaining causal mechanisms, and responding to urgent events.

According to Ergüner Özkoç [2021] and numerous reviewed literature time-series similarity analysis and clustering play a crucial role in understanding, organizing, and extracting meaningful information from time-series data, leading to improved decision-making, problem-solving, and data-driven insights.

3.3 Clusted-Based Time Series Forecasting

Many studies employ cluster-based techniques to tackle the complexity posed by diverse devices, behaviors, and data streams in training ML models. Researchers use clustering methods to group similar time series data based on their similarities, aiming to enhance the effectiveness of ML models. This approach enables the training of models by capturing patterns and dependencies in time series data through the utilization of clusters, ultimately resulting in enhanced accuracy in predictive forecasting.

The research conducted by Bogado Machuca et al. [2023] explores the application of cluster-based LSTM models to enhance the precision of dengue cases forecasts by integrating information from similar time series and weather data. The study addresses the issue of heterogeneous behaviors and the limited accuracy of LSTM models, particularly in regions with sparse data. The authors suggest a clustering analysis across time series using scores to evaluate the quality of clustering in 217 cities in Paraguay. They compare various clustering techniques, both raw and feature-based, and conclude that hierarchical clustering combined with Spearman correlation proves to be the most effective approach. The paper highlights a notable enhancement in model accuracy by $19.48 \pm 18.80\%$ achieved through the utilization of clustered models.

In their study, Zhu et al. [2018] employ time series clustering to identify anomalies in flight vibration, addressing the challenge posed by uncorrelated features in time series data. The approach utilizes Spearman's rank correlation coefficient and a hierarchical clustering method to group related time series. The clustering outcomes reveal the aggregation of monotonically similar series, effectively removing unrelated ones. This technique is applied to mitigate the impact of uncorrelated features on the prediction model. The clustering results, integrated into the prediction model, notably decrease the RMSE of predicted outcomes, as evidenced in the experimental analysis conducted on COMAC's C919 flight test data.

In their investigation, Kim et al. [2023] utilize Euclidean distances and DTW distance as metrics for assessing similarity in time-series data. The clustering process employs the K-means method, with households being categorized into clusters to capture distinct characteristics in electricity usage data over time. Prior to forecasting, clustering is conducted to group households exhibiting similar electricity usage patterns, resulting in enhanced forecasting accuracy. The findings indicate that the approach of clustering households and forecasting electricity usage for each cluster outperforms the prediction of total electricity usage for all households without clustering. In addition, the study affirms that incorporating exogenous variables such as cooling degree day, humidity, and insolation contributes to improved forecasting performance compared to relying solely on electricity consumption data.

The approach described in Laurinec and Lucká [2018] involves a preprocessing step for time series data, encompassing normalization and the computation of diverse model-based time series representations. Subsequently, consumer clustering is performed using either K-means or K-medoids, and forecasts are generated based on the centroids of these clusters. The final forecasts, derived from the centroids, are adjusted by applying the saved normalization parameters for each consumer. This clustering-based methodology significantly enhances forecasting accuracy, particularly for residential consumers, and offers greater scalability compared to a fully disaggregated approach. Notably, this scalability is achieved by training the model on clusters rather than individual consumers. The evaluation results, conducted on smart meter datasets from residences in Ireland and Australia, as well as factories in Slovakia, underscore the effectiveness of the clustering-based method in improving the accuracy of electricity load forecasting for individual consumers.

The study conducted by Tadayon and Iwashita [2021] employs DTW for distance-based similarity analysis and introduces two distinct feature extraction methods for time series data to facilitate feature-based similarity analysis. Prior to time series forecasting, K-means clustering is applied, aiming to enhance both the prediction time and forecasting performance of the neural network. The research introduces various neural network architectures utilizing the LSTM algorithm for dynamic mea-

surements in time series forecasting, exploring the impact of techniques such as anomaly detection and clustering on forecasting accuracy. The findings suggest that clustering not only improves overall prediction time but also enhances the forecasting performance of the neural network. The paper highlights that feature-based clustering surpasses distance-based clustering in terms of speed and efficiency.

Researchers in Daskalov and Nikolov [2017] employ a time series similarity analysis method that involves segmenting the initial data series into subseries and organizing them into clusters based on their shapes. Each cluster retains the relative differences between consecutive values for its subseries, which are then averaged to generate a consolidated series for each cluster. This methodology is utilized to discern patterns and similarities within the time series data, leading to the creation of clusters that represent akin shapes in the dataset.

As per Daskalov and Nikolov [2017], the clustering technique employed centers around grouping subseries into clusters in a manner that minimizes the distance between subseries within a group while maximizing distances between subseries in different groups. Various clustering algorithms, including K-means, ISODATA, hierarchical clustering, and self-learning neural networks, can be applied for this purpose. Clustering performed prior to forecasting serves to identify and capture patterns and similarities in the time series data, which can then be leveraged for predictions based on the cluster centers. This approach facilitates the recognition of distinct shapes and patterns in the data, contributing to more precise predictions.

3.4 Machine Learning Model Testing and Sharing

There are numerous other methods proposed for the sharing of ML models, however the main advantage of the objective and method proposed in this thesis, lies in its customized approach to addressing the unique challenges of IoT environments. By incorporating data similarity analysis and geospatial integration, this method offers context-aware model selection, localized deployment, resource-efficient sharing, and dynamic adaptation, thereby improving performance and efficiency compared to traditional methods. If proven viable, this method has the potential to be applied across numerous other IoT scenarios, offering benefits such as optimized resource utilization and improved adaptability to dynamic environmental conditions.

AI has significantly transformed various aspects of our lives, influencing both human interactions with the Internet and computational devices, as well as the way devices engage with us. This impact extends to industrial and socioeconomic domains where ML applications are gaining prominence. The IoT is a pivotal element in facilitating these interactions by providing contextual information that,

when processed, enhances intelligence in various processes. However, delivering ML applications in IoT encounters challenges due to the inherent complexity of ML operations, the multitude and diversity of IoT devices, and the need for online interoperability. Consequently, there is a pressing need to harness the potential of AI in IoT devices. However, developing ML models for the numerous IoT devices is a daunting, costly, and competitive endeavor, prompting scholars to explore avenues for sharing ML models among these devices.

The study conducted by Mira et al. [2023] introduces a platform for ML as a Service (MLaaS), specifically designed to offer intelligent applications within the IoT domain. This platform is structured to provide services that include DL training and inference online, the conversion and sharing of ML models, and the verification of zero-knowledge models using blockchain technology. By tackling the inherent intricacies of ML operations and ensuring online interoperability with IoT devices, the platform aims to augment IoT capabilities with enhanced intelligence.

The research by Resifi et al. [2022] addresses the challenges associated with deploying DL models in resource-constrained environments such as the IoT, where DL models demand significant computational resources. The authors suggest two strategies to overcome this challenge: sharing the DL model between the cloud and the device, and optimizing model execution through early exiting, a method where the entire model does not need to run for certain inputs. These approaches are automatically optimized to determine the most effective points for sharing and early exiting based on input, offering a versatile solution applicable across various scenarios and providing a viable option for local execution of DL models.

In their work Zhou et al. [2021], researchers propose a Ciphertext Policy Attribute Based Proxy Re-encryption CP-ABPRE scheme that incorporates accountability to address security and privacy concerns in the sharing of edge intelligence (EI) models for the IoT. The authors intend to facilitate the sharing of ML models by enabling users to delegate access rights, incorporating unique IDs for traceability, and conducting a security analysis and performance evaluation to showcase the efficacy of their proposed scheme. The CP-ABPRE framework ensures adaptable data access and security while establishing accountability for both edge nodes and users involved in EI model sharing. Furthermore, the scheme is designed to be resilient against Chosen Plaintext Attacks (CPA), providing robust security without compromising efficiency, even with additional features.

The research paper by González-Soto et al. [2024] introduces a collaborative and decentralized ML framework specifically designed for IoT devices with limited resources. The focus is on sharing ML models using random-based protocols and decentralized prototype sharing protocols. This approach involves sharing local models

among computing elements within the network, enhancing both the diversity and quantity of available data to improve the overall performance of the network model. The study assesses the effectiveness of these sharing protocols and underscores the significance of sharing parameters in decentralized ML. The findings reveal promising results in terms of accuracy, training time, and robustness when compared to traditional centralized approaches. The paper contributes valuable insights into the influence of data sharing protocols on ML performance, emphasizing the critical role of selecting optimal sharing parameters to achieve optimal performance and resource efficiency in decentralized ML frameworks.

The Nguyen et al. [2021] paper introduces a novel ecosystem for trading ML models on a secure Blockchain-based network, with a specific emphasis on promoting collaborative training and data/model exchange for IoT devices. The proposed system facilitates the sale of contributions in training ML models, allowing buyers to acquire these models. All transaction details are securely recorded in a tamper-proof distributed ledger.

In this ecosystem, participants share updated model weights with the marketplace. An aggregator then consolidates these locally trained models to create global models. The results of this approach demonstrate competitive runtime performance, accompanied by a 15% reduction in execution costs. The system ensures fairness in terms of incentives for participating individuals.

This section, presents comparative analysis of various ML sharing methods, aiming to explain their strengths, weaknesses, and applicability in diverse contexts. The table below 3.2 compares ML sharing mechanisms. This comparative examination,seeks to contribute to the advancement of ML methodologies while creating a deeper understanding of the trade-offs inherent in different sharing paradigms.

Table 3.2.: Comparison of Machine Learning Sharing Approaches

Reference	**Approach**	**Key Contributions**	**Benefits of This Thesis**
Mira et al. [2023]	MLaaS for IoT	DL training and inference online, model conversion and sharing, verification of zero-knowledge models using blockchain	This study introduces a novel CoI-based approach for efficient ML model sharing, tailored specifically for IoT environments.
Resifi et al. [2022]	DL model optimization for IoT	Strategies for model sharing between cloud and device, model execution optimization through early exiting	This thesis offers a structured framework for CoI formation and model sharing, optimizing resource utilization in IoT deployments.
Zhou et al. [2021]	CP-ABPRE for EI model sharing	Security and privacy framework for sharing edge intelligence models in IoT	This research emphasizes contextual factors like data similarity and geospatial components, enhancing model sharing efficacy.
González-Soto et al. [2024]	Decentralized ML framework for IoT	Random-based and decentralized prototype sharing protocols for enhancing model diversity and quantity	This study leverages advanced clustering and similarity analysis, ensuring more effective grouping and sharing among IoT devices.
Nguyen et al. [2021]	Blockchain-based ML model trading	Secure ecosystem for collaborative training and data exchange among IoT devices	This thesis provides a structured CoI formation approach, enabling transparent and efficient model trading within IoT networks.

4. Methodology

The literature review serves as the foundational framework for this methodology. The methodology chosen to support this research is influenced by both the achievements and limitations identified in existing research, as discussed in Section 2 the background.

The main objective is to use CoIs to develop, train, validate, and share ML models among IoT devices. These CoIs will be formed based on the similarity of the IoT data streams and the only additional input is geographic factors such as location and elevation, which influence the ML model outcomes to varying degrees. The ultimate goal is to present an innovative and reusable approach for sharing ML models, with the aim of reducing cost, improving competitiveness and simplifying the process of creating ML models for the multitude of IoT devices generating data. Through out this study the term ML is used in place of TinyML models which can be generated using Tensorflow lite.

This study opted to use a cluster-based methodology and selected optimal measures for similarity analysis and evaluation to establish CoI, which facilitates the development, validation, testing, and sharing of ML models. The methodology used in this study is illustrated in Figure 4.1.

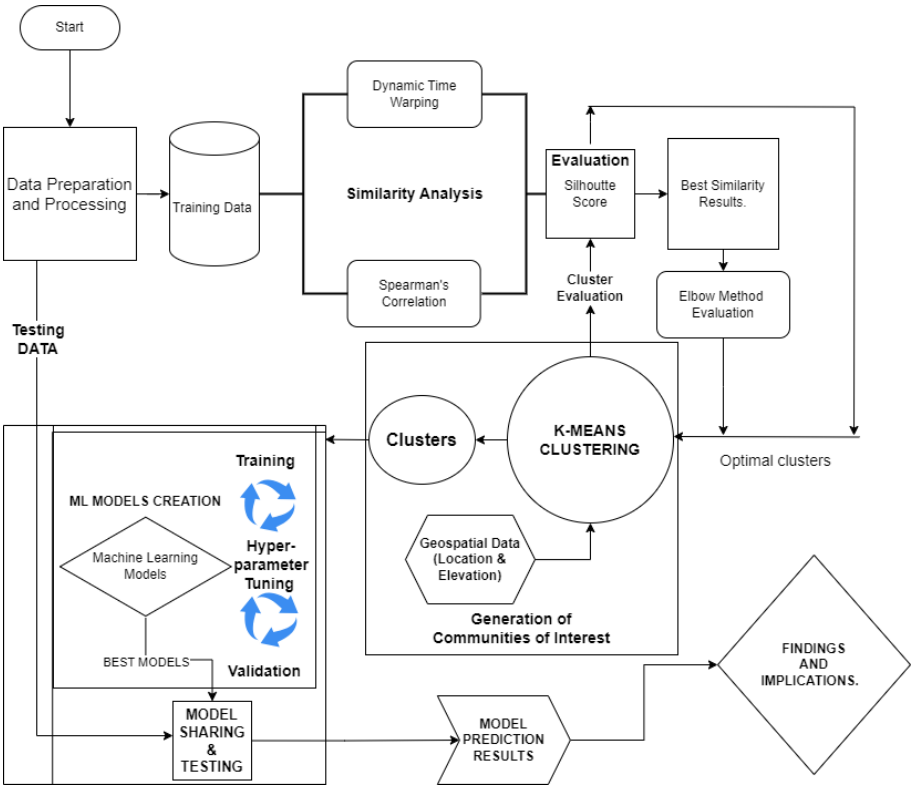


Figure 4.1.: The workflow to share ML models among IoT devices based on CoI.

4.1 Study Area

This study leveraged the data from weather sensor network to simulate real-world scenarios involving IoT devices. The weather sensor map plays an important role in this study by providing a spatial representation of 43 AVAMET weather sensors distributed across Castelló province, Valenciana Community, Spain (Figure 4.2). These sensors were carefully selected based on their consistent data streaming reliability throughout the years 2021 and 2022, ensuring the integrity of the dataset. While there are additional weather sensors in the region, they have been excluded from the study map due to inconsistencies in data streaming, rendering them unreliable for research purposes.

Figure 4.2.: Distribution of Avamet weather sensors in Castellon Province, Spain.

The map serves as a foundational visualization tool for investigating how location or geospatial factors influence the similarity of sensor data streams and the formation of CoI. By visualizing correlations of sensor data with geographic attributes such proximity to coastlines, or urbanization levels, the study aims to uncover spatial patterns in sensor data similarity and identify clusters or communities of sensors with similar data profiles.

This study will use the stations with the highest silhouette scores to train ML models. This is because these stations are likely to share common features or patterns that distinguish them from stations in other clusters. Stations with high

silhouette scores within a cluster are more representative of the cluster's overall characteristics and can be considered as prototypical members of that cluster.

4.2 Exploratory Data Analysis

To unravel the inherent patterns, trends, and anomalies encapsulated within the temporal evolution of the temperature readings, a comprehensive analysis was undertaken to identify anomalies and missing data. This exploration delves into the multifaceted aspects of the dataset, encompassing seasonality, long-term trends, and potential irregularities, with the aim of discerning meaningful insights into the dynamic behavior of temperature variations over time.

Data Visualization

To enhance the comprehension of patterns, trends, and insights within the temperature data, a comprehensive temperature time series plot was generated for the entire dataset, along with a randomly chosen subset of stations. This visualization aims to simplify the presentation of intricate data, making it more accessible for easier understanding. The objective is to foster a nuanced and well-informed interpretation of the temperature dataset, supporting thorough analysis and informed decision-making processes (Figure 4.3).

Randomly selected stations have been included to ensure a representative sample, allowing for a more holistic exploration of temperature variations across different locations. The purpose these visualizations play is to facilitate a clearer understanding of the temperature data and contribute to a robust analytical approach.

Data Distribution

This study employs a box plot to succinctly summarize the temperature distribution data of Almenara - Comunitat de Regants, a station exhibiting the highest variance of 117.07 degrees within the dataset. By focusing on this station, the analysis aims to capture a broad spectrum of temperature fluctuations, encompassing both typical ranges and extreme values. The utilization of the box plot facilitates the visualization of the temperature distribution, offering insights into the central tendency, variability, and presence of outliers. This approach enables the identification of patterns, anomalies, and potential drivers of temperature variability, laying the foundation for hypothesis generation and further in-depth analysis.

The right-leaning median line see Figure 4.4 suggests that the majority of temperature readings cluster towards the lower end of the distribution, with fewer high values, indicating positive skewness. Despite the presence of high temperatures, they are relatively close to the upper quartile (Q3), implying a narrower spread of high values compared to lower values. This consistency in high temperature readings,

without extreme outliers, suggests stability influenced by factors like geographical location and climate conditions.

To enrich the depth and relevance of the research findings and to contributing to a better understanding of temperature dataset dynamics the station with the highest mean temperature Castellon de la Plana-IES Vicent Sos Baynat was selected for plotting a box plot.

The resulting box plot displaying whiskers on both the left and right sides see Figure 4.5, coupled with a median line leaning towards the second quartile or to the left shows a symmetrical spread of temperature readings around the median suggesting a balanced variability in both lower and higher temperature values. The left-leaning median indicates a central tendency towards lower temperatures within the dataset, reflecting a notable shift towards cooler conditions. The absence of outside points or outliers underscores the uniformity and stability of the temperature data, devoid of any extreme values.

Data Seasonality

An investigation into seasonal patterns and fluctuations is conducted to reveal seasonal variations within the dataset. This analysis of seasonality is carried out on two stations: Almenara - Comunitat de Regants, which exhibits the highest variability, and Castellon de la Plana-IES Vicent Sos Baynat, which has the highest mean temperature see Figure 4.6. The identification of seasonal trends and patterns at these stations aids in evaluating long-term climate trends.

Upon comparing the seasonality between the two stations, it becomes apparent that the dataset from Almenara - Comunitat de Regants displays irregular patterns, prompting a need for closer scrutiny. This comparative approach and data examination contribute to a deeper comprehension of climate variability, the detection of abnormal patterns or datasets, and informs subsequent stages of the research, including hypothesis formulation and in-depth analysis.

The abnormal seasonal plot observed for Almenara - Comunitat de Regants explains why it exhibits a high positively skewed variance. This skewness indicates a tendency towards higher variability and the presence of extreme values in the dataset. This data exploration not only enhances our understanding of climate dynamics at the Almenara station but also provides valuable context for interpreting and analyzing the rest of the datasets. These insights informed adjustments to data analysis and cleaning techniques, hypothesis formulation, and decision-making processes, ultimately contributing to more accurate and comprehensive research outcomes.

	Artana	**Benassal**	**La Mata**	**La Pobla-Tornesa**	**Les-Useres**
count	8760	8760	8760	8760	8760
mean	16.2	13.2	7.42	13.51	16.29
std	6.92	6.86	9.26	8.41	6.31
min	0.0	-3.80	-1.9	-2.5	0.0
max	38.1	35.3	38.0	38.2	36.3

Table 4.1.: Summary statistics for temperature data at different stations.

Monthly Temperature Trends

This thesis computed the mean monthly temperature during exploratory data analysis to help in identifying seasonal patterns, and provide insight into how temperatures fluctuate throughout the year. Secondly,to aid in anomaly detection by highlighting any unusual deviations from the typical temperature patterns, which could indicate extreme weather events or errors in data collection.

The mean monthly temperatures were also used to offer a concise summary of the data, facilitating easier visualization and interpretation of temperature trends over time (Figure 4.7). By comparing mean temperatures across different months, long-term trends were identified. These mean temperatures serve as valuable features for further analysis such as predictive modeling or time series forecasting.

Data Description

To provide a summary of the essential statistics of the data and to aid in understanding the characteristics of the data. This study computed key descriptive statistics, including count, mean, standard deviation, minimum and maximum values, as well as percentiles such as the median and quartiles. See Figure 4.1

This information is essesntial to quickly grasp the central tendency, variability, and distribution shape of the dataset, to facilitate informed decision-making in data analysis and interpretation. This was done to identify outliers, assess data quality, and gain insights into the underlying patterns and trends within the data.

4.2.1 Data Preparation

In this study, the data source employed was AVAMET, the Meteorological Association of Valenciana, which supplied weather sensor data for the years 2021 and 2022 stored initially in text files which were then converted to CSV format. The dataset contained data from over 120 weather stations with vital meteorological parameters, including temperature, humidity, wind speed, precipitation, and atmospheric pressure.

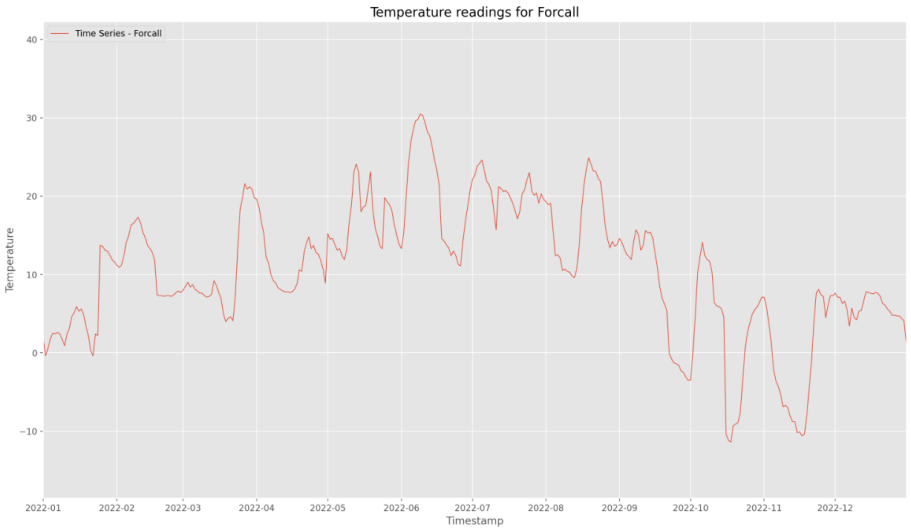
To narrow the focus of the analysis to temperature data, a data extraction process was executed to isolate and retain only the temperature data from both the

2021 and 2022 datasets. These combined measures ensured the creation of a robust and standardized dataset, laying the foundation for subsequent analyses and interpretations in this study.

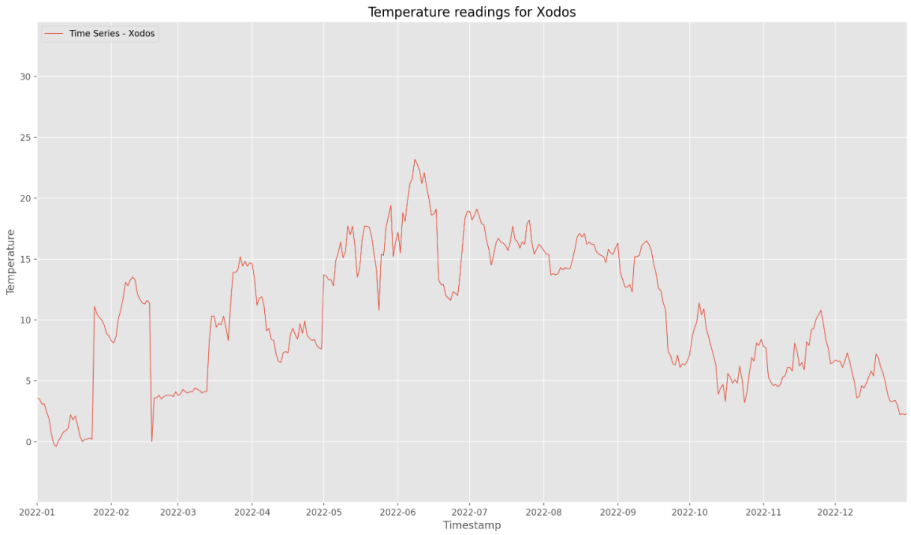
A series of comprehensive data molding and transformation techniques were applied. To deal with irregular time stamps in the data, the dataset was standardized by homogenizing and resampling the time stamps to 10-minute intervals. This was done to enhance the quality, consistency, and usability of weather data, to ultimately improve the accuracy and reliability of analyses and forecasts based on that data.

Next, weather stations with consistent data streaming patterns for both 2021 and 2022 were identified by filtering out stations that had at least 90% data availability and no more than 10% missing values, ensuring reliable and continuous data streams.To uphold the dataset's completeness,addressing missing data through effective data imputation techniques was essential. This study employed methods such as mean, median, and forward fill to fill in missing data points effectively, ensuring completeness and accuracy in the dataset. An indispensable refinement step involved data normalization to detect and manage outliers efficiently.

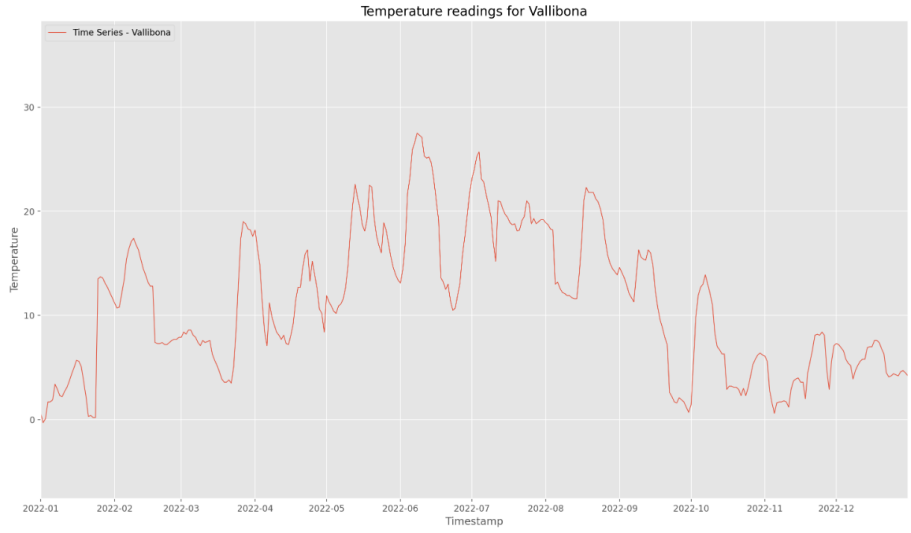
After careful data preparation, the study narrowed down to 43 Avamet weather sensors for both the 2021 and 2022 datasets. Subsequently, the station names were correlated with their respective station codes and geospatial data, including location and elevation. Ensuring the reliability of these sensors was paramount, given that the 2021 dataset would be employed for developing and training forecast models, while the 2022 dataset would be used for evaluating the prediction accuracy of these models. These datasets will mimic real-world IoT device scenarios.



(a) Forcall



(b) Xodos



(c) Vallibona

Figure 4.3.: Temperature time series Forcall 4.3a, Xodos 4.3b and Vallibona stations 4.3c.

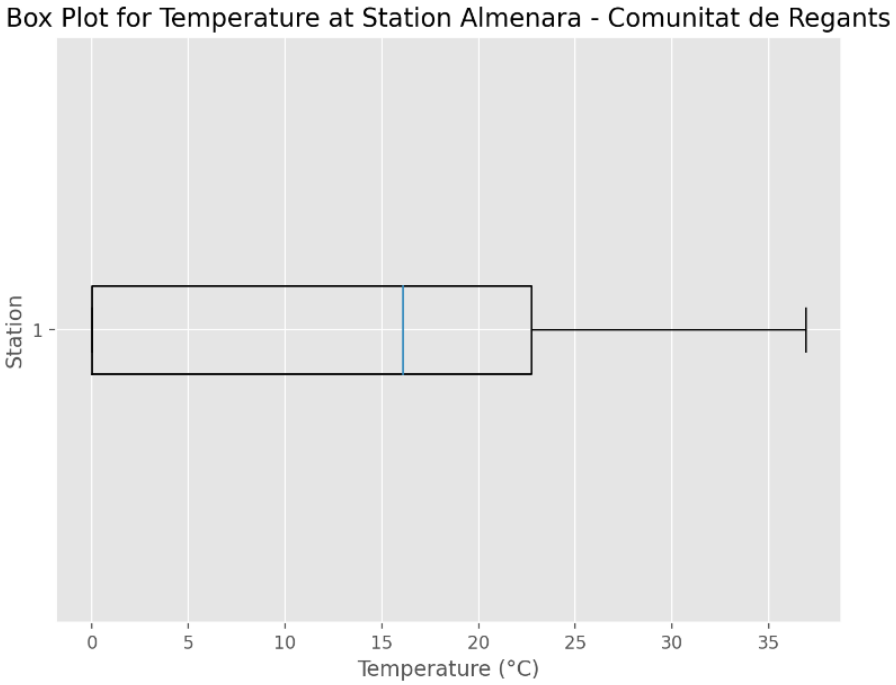


Figure 4.4.: Temperature distribution for Almenara - Comunitat de Regants.

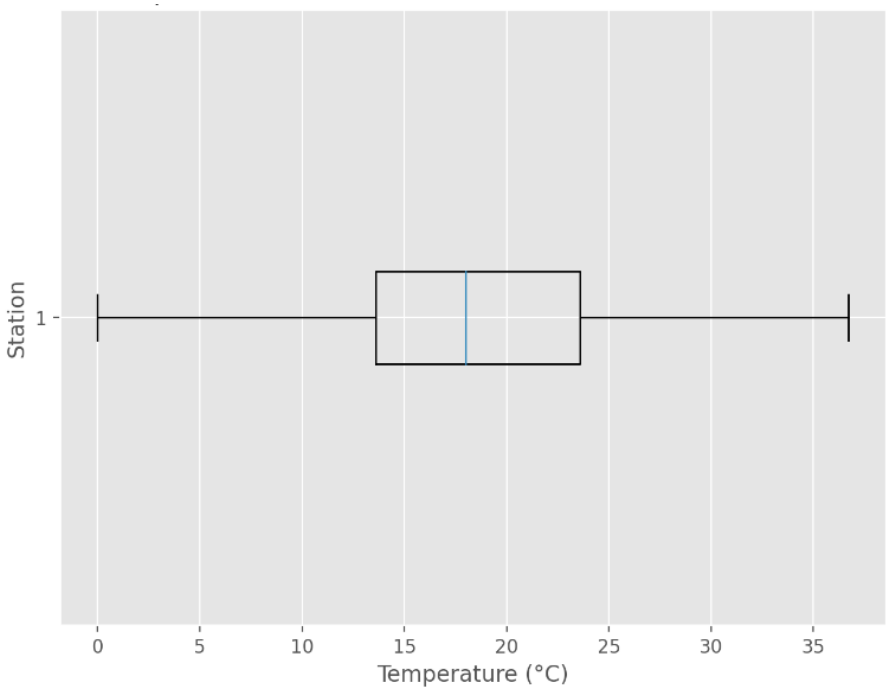


Figure 4.5.: Temperature distribution for Castellon- IES Vicent Sos Baynat.

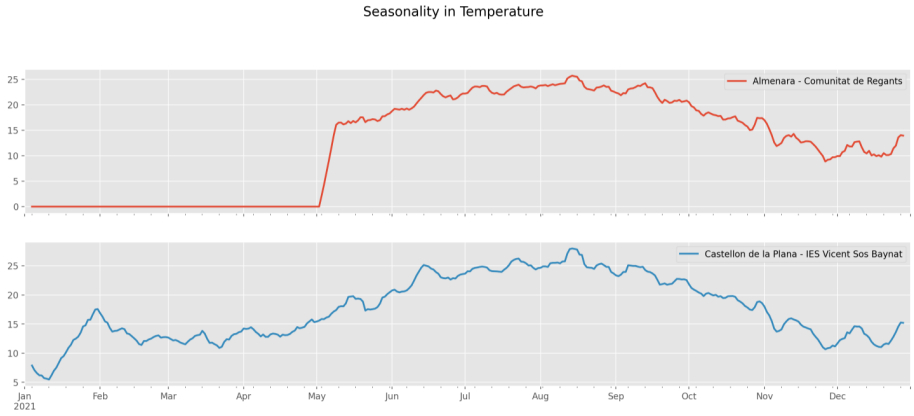


Figure 4.6.: Seasonality plot for Almenara - Comunitat de Regants, Castellon de la Plana-IES Vicent Sos Baynat stations

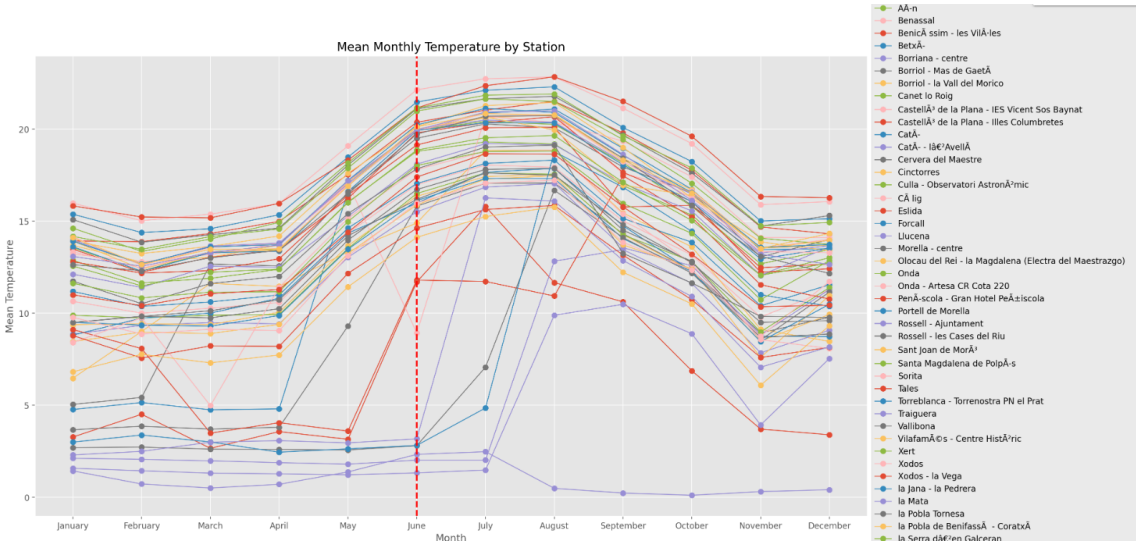


Figure 4.7.: Mean Monthly Temperature.

5. Development

This chapter provides details about the implementation of the methodology. It includes code snippets to illustrate key components of the analysis and provide explanations for each snippet.

5.1 Similarity Analysis

Following extensive data preparation and standardization, this research utilized two distinct similarity analysis metrics, as recommended by numerous research articles mentioned in the related works, to assess the similarity of temperature time series data. The research applied DTW, an approach based on shape, and Spearman's correlation, which is a feature-based approach.

Dynamic Time Warping

DTW was selected for this study based on its recognition by Aghabozorgi et al. [2015] and several other scholarly articles as one of the most widely adopted shape-based methods for similarity analysis. According to Keogh and Ratanamahatana [2005], DTW stands out as a notably robust distance measure for time series, allowing similar shapes to align, even when they are out of phase along the time axis. This analysis was conducted on google colab environment and the code used in this thesis for the DTW is made available on Github[1].

To assess the intrinsic similarities among a multitude of time series, each representing the temperature observations from distinct weather stations. The DTW algorithm operates by constructing a cost matrix, wherein each element encapsulates the cost of optimally aligning specific observations from different stations.

As the matrix is systematically populated, the DTW distance, located at the matrix's conclusion, captures the dissimilarity between any two stations' temperature profiles. This not only facilitates the identification of highly similar or dissimilar station pairs but also allows for the construction of a holistic similarity matrix encompassing all 43 stations with lower values indicating greater similarity. DTW incorporates a backtracking step to unveil the optimal alignment path, revealing how each point in one sequence corresponds to points in the other.

This snippet of code in Listing 5.1 demonstrates the implementation of DTW in python. The code imports necessary libraries such as NumPy for numerical operations and the fastdtw module for efficient DTW calculation. It defines a function 'dtw distance' that takes two time series as input and returns their DTW distance.

[1]Github: `https://github.com/MikeSirya/Master-Thesis.git`

This analysis was conducted in Google Colab environment and the code used in this thesis is made available on Github.

Listing 5.1: Python code for Dynamic Time Warping Similarity Analysis.

```
import pandas as pd
import numpy as np
from fastdtw import fastdtw
import seaborn as sns
import matplotlib.pyplot as plt

distance between two columns
def calculate_dtw_distance(column1, column2):
    valid_indices = np.isfinite(column1) & np.isfinite(
       column2)
    distance, _ = fastdtw(column1[valid_indices], column2[
       valid_indices])
    return distance

num_columns = len(seasonality_df.columns)
dtw_distance_matrix = np.zeros((num_columns, num_columns))

for i in range(num_columns):
    for j in range(i + 1, num_columns):
        dtw_distance_matrix[i, j] = calculate_dtw_distance(
           seasonality_df.iloc[:, i], seasonality_df.iloc
           [:, j])
        dtw_distance_matrix[j, i] = dtw_distance_matrix[i,
           j]

dtw_distance_df = pd.DataFrame(dtw_distance_matrix, index=
   seasonality_df.columns, columns=seasonality_df.columns)

distance matrix
fig = sns.clustermap(
    dtw_distance_df,
    annot=True,
    annot_kws={"size": 10},
    linewidths=0.4,
    figsize=(15, 10),
    cmap='viridis',  # Change color map to 'viridis', you
       can choose other colormaps
)

```

```
plt.setp(
    fig.ax_heatmap.xaxis.get_majorticklabels(),
    rotation=90,
)
plt.show()
```

Spearman's Correlation

Spearman's correlation is particularly valuable when assessing the monotonic relationship between two variables. Unlike Pearson's correlation, Spearman's correlation does not assume linearity and is particularly robust in the presence of outliers. In the context of weather station temperature data analysis, Spearman's correlation was utilized to discern the degree and direction of monotonic associations between the temperature observations from different stations.

The Spearman's correlation coefficient is calculated by first ranking the values in each time series. The correlation is then computed based on the ranks rather than the original values, making it less sensitive to extreme values and better suited for capturing non-linear relationships.

When applied to the temperature time series it provides insights into the consistency and direction of temperature trends across different locations. A positive Spearman's correlation suggests a monotonic increasing relationship, while a negative correlation indicates a monotonic decreasing relationship. A correlation close to zero implies a lack of monotonic association.

Listing 5.2: Python code for Spearmans Correlation Similarity Analysis.

```
seasonality_corr = seasonality_df.corr(
    method="spearman"
)

correlation_matrix_csv = 'correlation_matrix.csv'
seasonality_corr.to_csv(correlation_matrix_csv)

fig = sns.clustermap(
    seasonality_corr,
    annot=True,
    annot_kws={"size": 10},
    linewidths=0.4,
    figsize=(15, 10),
)

plt.setp(
    fig.ax_heatmap.xaxis.get_majorticklabels(),
```

```
    rotation=90,
)
plt.show()
```

The code snippet in listing 5.2 calculates the Spearman correlation matrix in python. The ‘corr‘ method is used with the parameter ‘method=”spearman”‘ to compute the correlation matrix based on Spearman’s rank correlation coefficient, which measures the strength and direction of monotonic relationships between variables. To visualize the correlation matrix, the seaborn ‘clustermap‘ function is employed. The function generates a hierarchical clustering heatmap of the correlation matrix, with annotations showing the correlation coefficients. This analysis was conducted on google colab environment and the code used in this thesis is made available on Github.

5.1.1 Performance Evaluation

In the methodological framework of this Master thesis, the evaluation of similarity methods and subsequent clustering efficacy is comprehensively undertaken through the application of the Silhouette Score. Before delving into the clustering phase, DTW and Spearman’s correlation are independently scrutinized.

5.1.1.1. Silhouette Score

This study employs the Silhouette Score as a pivotal metric to quantify the would be coherence of clusters to be formed by clustering the similarity analysis results of each method applied to the temperature time series data.

A higher Silhouette Score is indicative of more distinct and well-defined clusters, shedding light on the inherent effectiveness of each similarity method in capturing nuanced temporal patterns. This pre-clustering assessment allows for the identification of the superior similarity method, laying the groundwork for subsequent analyses. The K-means algorithm is then employed to group these stations into distinct clusters, considering the similarities revealed by the chosen method (DTW or Spearman’s correlation).

Post-clustering, the Silhouette Score is once again employed to assess the quality of the obtained clusters, providing a quantitative measure of the clustering efficacy based on the similarity information extracted from the temperature time series data.

5.1.1.2. Elbow Method

This Master thesis, integrated the elbow method in the determination of the optimal number of clusters for the forthcoming K-means clustering analysis. Prior to the application of K-means to the similarity matrix, derived from either DTW or Spearman’s correlation, a systematic exploration of various cluster counts is con-

ducted.

The elbow method involves running K-means for a range of cluster values and plotting the resultant sum of squared distances against the number of clusters. This plot exhibits a discernible "elbow" point where the reduction in the sum of squared distances diminishes, indicating the optimal number of clusters. The elbow point signifies the balance between capturing meaningful patterns within the data and avoiding overfitting by introducing unnecessary clusters.

Listing 5.3: Python code for Elbow Method Implementation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

correlation_matrix_csv = 'correlation_matrix.csv'
seasonality_corr = pd.read_csv(correlation_matrix_csv,
    index_col=0)

inertia = []
max_clusters = 10

for k in range(1, max_clusters + 1):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(seasonality_corr)
    inertia.append(kmeans.inertia_)

plt.plot(range(1, max_clusters + 1), inertia, marker='o')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.show()
```

The python code snippet in listing 5.3 implements the elbow method to ascertain the optimal number of clusters (k) for K-means clustering based on the inertia values. The code calculates the inertia for varying values of k, ranging from 1 to a predefined maximum. Utilizing K-means clustering for each k, the code computes the inertia, representing the sum of squared distances of samples to their closest cluster center, and appends these values to a list. Subsequently, a plot of the elbow curve is generated, illustrating the relationship between the number of clusters and inertia. This analysis was conducted on google colab environment and the code used in this thesis is made available on Github.

5.1.2 K-Means Clustering

In this study, K-means clustering assumes a crucial role following the completion of preliminary similarity analyses facilitated by DTW and Spearman's correlation. Having conducted DTW to capture nuanced temporal variations and Spearman's correlation to discern monotonic relationships among the temperature time series from multiple weather stations, the subsequent application of K-means clustering aims to distill these intricate patterns into discernible clusters.

The process unfolds by selecting a predetermined number of clusters using the Elbow method, denoted as K, and initializing cluster centroids. Leveraging the insights gleaned from the earlier similarity analyses, additional data containing the location and elevation of the weather sensors was added, K-means then iteratively assigns each weather station's temperature profile to the cluster with the closest centroid, fostering the grouping of stations that exhibit similar temporal and geospatial characteristics.

The algorithm refines these assignments and updates the centroids until convergence is achieved, yielding a final partitioning of weather stations into distinct temperature behavior groups. This sequential approach, encompassing similarity analysis with DTW and Spearman's correlation, merging with geospatial data, followed by K-means clustering, synergistically empowers the study to explore, categorize, and interpret the diverse temperature dynamics exhibited across the weather station network. Figure 5.1 shows the steps and procedures undertaken in implementing the K-means algorithm.

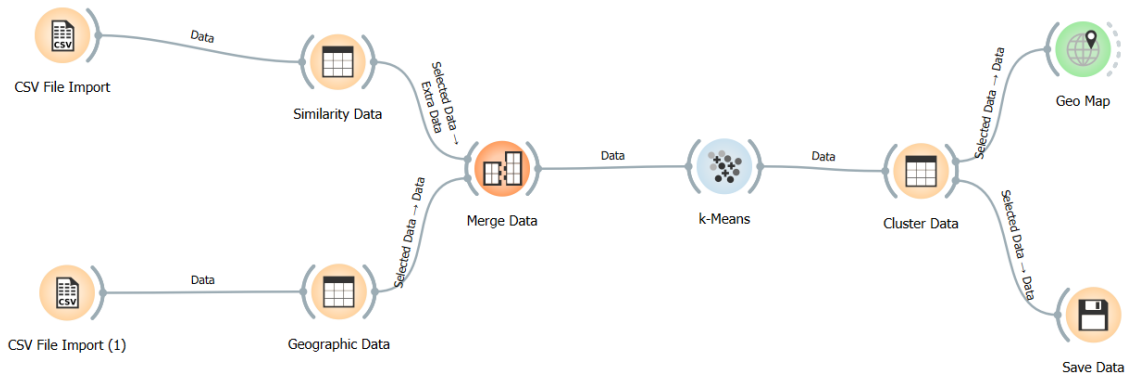


Figure 5.1.: K-Means Clustering flow Chart

5.2 Machine Learning Model Development

The process of modeling the ML models in this study began after the K-means clustering grouped the IoT devices into CoI. Two suitable ML algorithms were chosen. For each CoI formed, data from one station with the highest Silhoutte score was chosen for the training of the ML models such that there was a ML model for

each of the CoI. Then models were then trained on portions of the dataset, and their performance evaluated on, a separate validation set using RMSE. Hyperparameter tuning and cross-validation were employed to optimize the model, and their generalization was validated on an independent dataset the 2022 Castellon weather sensor data. The iterative nature of this process involved continuous refinement to ensure the model effectively addresses the problem and generalizes well to the new, unseen data.

5.2.1 Long Short Term Memory

LSTM is a specialized recurrent neural network architecture designed to address the challenge of capturing and learning long-range dependencies in sequential data. LSTM models incorporate memory cells equipped with input, forget, and output gates. Figure 5.2 shows how gates can regulate the flow of information into, out of, and within the memory cells, enabling the network to selectively store or discard information.

The unique architecture of LSTMs allows them to effectively handle the problem of vanishing gradients, a common issue in deep networks. LSTMs maintain an internal state, facilitating the retention of relevant information over extended sequences. This ability to capture and remember long-term dependencies makes LSTMs particularly valuable in applications such as natural language processing, speech recognition, and time series analysis. The success of LSTMs lies in their capacity to model intricate patterns in sequential data, contributing significantly to the advancement of DL techniques for various real-world tasks.

5.2.2 Random Forest

Random Forest, an ensemble learning method, operates by constructing a multitude of decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Each tree is built on a random subset of the training data and a random subset of features, utilizing bootstrapped sampling to introduce diversity among the trees. The randomness extends to feature selection for each split in a tree, preventing the dominance of a single feature and enhancing overall robustness. The final prediction in classification tasks is determined by a majority vote from all trees, while regression tasks rely on the average prediction.

This ensemble approach mitigates overfitting, as errors from individual trees are balanced out by the collective decision. Random Forest also leverages out-of-bag samples to estimate model performance without requiring a separate validation set. The algorithm provides insights into feature importance, aiding in the interpretation of the model. Renowned for its versatility, Random Forest finds application

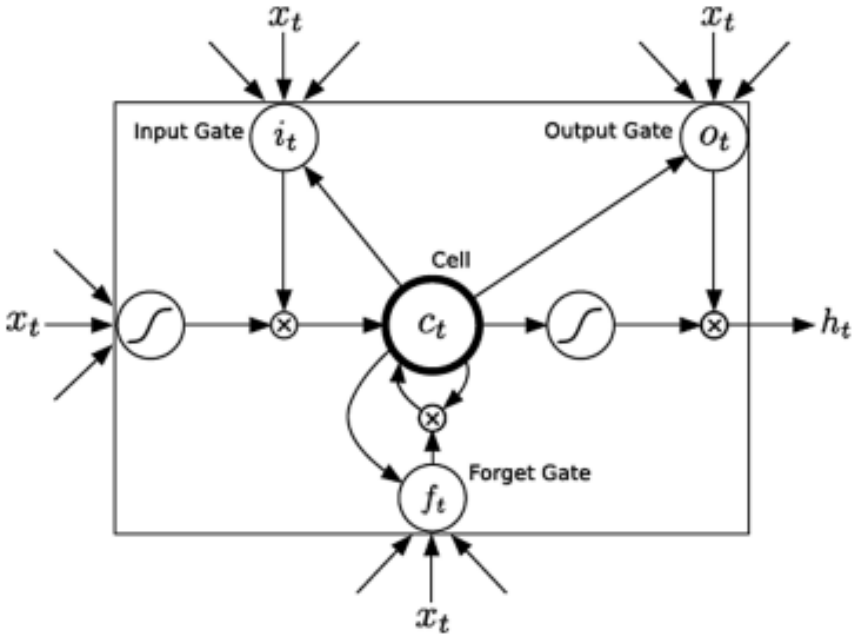


Figure 5.2.: Long Short Term Memory Cell. Source :Zhu et al. [2018]

in diverse domains due to its effectiveness in handling high-dimensional data and delivering robust predictions.

Listing 5.4: Python code for Random Forest Implementation

```
1
2  import pandas as pd
3  import numpy as np
4  from sklearn.model_selection import TimeSeriesSplit,
       GridSearchCV
5  from sklearn.ensemble import RandomForestRegressor
6  from sklearn.metrics import mean_squared_error
7
8  # Features (X) and Target Variable (y) using df_to_X_y
9  WINDOW_SIZE = 5
10 X, y = df_to_X_y(temp, WINDOW_SIZE)
11
12 # Nested Cross-Validation
13 tscv_outer = TimeSeriesSplit(n_splits=5)
14 tscv_inner = TimeSeriesSplit(n_splits=3)
15
16 param_grid = {'n_estimators': [10, 50, 100], 'max_depth': [
       None, 10, 20]}
17
18 rf_model = RandomForestRegressor()
```

```
19
20 grid_search = GridSearchCV(estimator=rf_model, param_grid=
       param_grid, scoring='neg_mean_squared_error', cv=
       tscv_inner)
21
22 for train_outer, test_outer in tscv_outer.split(X):
23     X_train_outer, X_test_outer = X[train_outer], X[
           test_outer]
24     y_train_outer, y_test_outer = y[train_outer], y[
           test_outer]
25
26     # Flatten the X_train_outer array
27     X_train_outer_flat = X_train_outer.reshape(
           X_train_outer.shape[0], -1)
28
29     grid_search.fit(X_train_outer_flat, y_train_outer)
30
31     best_model = grid_search.best_estimator_
32
33     # Flatten the X_test_outer array for prediction
34     X_test_outer_flat = X_test_outer.reshape(X_test_outer.
           shape[0], -1)
35
36     y_pred_outer = best_model.predict(X_test_outer_flat)
37
38     mse = mean_squared_error(y_test_outer, y_pred_outer)
39     print(f'Mean Squared Error for this fold: {mse}')
```

The python code snippet in listing 5.4 illustrates the implementation of a Random Forest Regressor model utilizing nested cross-validation for time series forecasting. Initially, the feature matrix (x) and target variable (y) are prepared using a custom function with a defined window size. Subsequently, nested cross-validation is established with two levels of time series splits to ensure robust evaluation and hyperparameter tuning. Within each outer fold, a grid search is conducted to identify the optimal hyperparameters for the Random Forest model based on negative mean squared error. The best model obtained from the inner cross-validation loop is then evaluated on the testing data for the outer fold, and the mean squared error is computed as a performance metric.

5.2.3 Model Implementation and Evaluation

5.2.3.1. Training Data

After the K-means clustering, the assessment of the resultant clusters was conducted using the silhouette score. Within each cluster, stations with the highest silhouette scores were identified.

The selection of stations with the highest silhouette scores for training ML models within each community of interest was important for several reasons. Firstly, high silhouette scores indicated that the data points within these stations were well-clustered and distinct from other clusters, ensuring that the training data were representative of the underlying patterns within each community. This enhanced the models' ability to accurately capture the characteristics of each community and generalize well to unseen data.

Training models on stations with high silhouette scores focused the efforts on the most informative data points, leading to more efficient model training processes thus optimizing computational resources. This made the models less prone to overfitting, as they were less likely to capture noise or irrelevant features in the data, thus improving the models' accuracy, generalization, and interpretability in addressing the CoI.

5.2.3.2. LSTM Model Parameters

In the implementation of the LSTM-based time series forecasting model, specific parameters were selected to optimize model performance and efficiency, further augmented by the integration of TensorFlow Lite to extend its functionality to IoT devices.. The LSTM layer is configured with 64 units, striking a balance between capturing intricate temporal dependencies in the data and computational efficiency.

Following the LSTM layer, a Dense layer with Rectified Linear Unit (ReLU) activation is introduced to enable the model to learn complex relationships. The output layer utilizes linear activation, aligning with the regression nature of time series forecasting. The Adam optimizer is chosen for efficient gradient-based optimization, with Mean Squared Error (MSE) as the loss function and Root Mean Squared Error (RMSE) as the evaluation metric for prediction accuracy.

In the implementation of our LSTM-based time series forecasting models,this study opted for simplicity in parameter choices, given the experimental nature of our research.The LSTM model underwent 10 epochs of training, a sensible choice aimed at preventing overfitting while allowing the model to learn from the data considering the experimental context, balancing the need for model complexity with the risk of memorization of noise in the training set.

A batch size of 32 was chosen, prioritizing computational efficiency while still

enabling the models to benefit from stochastic gradient descent. Empirical observations and experimentation confirmed that these parameter choices strike a suitable balance between model performance and computational resources for our experimental setup.

The experimental nature of the models and knowledge from related works [Bogado Machuca et al., 2023] guided the selection of these specific numbers for the LSTM hyperparameters to optimize both model performance and efficiency.

The implementation incorporated nested cross-validation with TimeSeriesSplit to robustly evaluate the models' performance while preserving temporal dependencies in both the outer and inner loops ensuring reliable assessment of the models' generalization ability in the context of this study's experimental investigations.

5.2.3.3. Random Forest Model Parameters

In the configuration of the Random Forest for time series forecasting, careful consideration has been given to the selection of parameters, with a deliberate emphasis on striking a delicate balance between computational efficiency and model expressiveness considering the experimental nature of the models. The choice of n estimators and max depth as the key hyperparameters stems from a thoughtful evaluation of the model's complexity and its ability to capture temporal dependencies within the dataset.

The parameter n estimators determines the number of decision trees in the forest. Through a grid search exploring values of 10, 50, and 100, a range of ensemble sizes was examined. This parameter significantly influences the trade-off between model performance and computational efficiency. While a higher number of estimators can enhance predictive capacity, potentially capturing more nuanced patterns in the data, it may also incur increased computational costs. The selection of these specific values reflects a pragmatic approach, allowing for comprehensive exploration while mitigating excessive computational burdens.

The max depth parameter sets the maximum depth of each individual tree in the forest. Considering None, 10, and 20, the grid search evaluates different levels of tree complexity. Deeper trees can capture more intricate patterns in the data but risk overfitting, especially with limited samples. By including these values in the grid search, we aimed to strike a balance between model expressiveness and generalization capability, given the experimental context of the models. The goal was to prevent overfitting while extracting meaningful patterns from the time series data, ensuring that the model's predictions are robust and reliable.

The selection of these specific numbers for the Random Forest hyperparameters was informed by a combination of empirical experimentation, insights from related

works [Tyralis and Papacharalampous, 2017], and the experimental nature of the models. These choices aimed to optimize both model performance and efficiency for the given time series forecasting task.

5.2.3.4. Model Cross-Validation Strategies

Cross Validation is systematic evaluation of a model's performance to ensure its effectiveness in making predictions on new, unseen data. The primary objective of Cross validation is to ensure that a model not only learns patterns effectively from the training data but also generalizes well to new, unseen instances. The systematic partitioning of the dataset into distinct subsets for training, validation, and testing, validation techniques is to prevent overfitting. Overfitting occurs when a model becomes too tailored to the peculiarities of the training data and fails to generalize to diverse or unseen examples. Through iterative model evaluation during the training phase, validation helps identify the optimal set of hyperparameters, enhancing the model's adaptability and predictive accuracy

5.2.3.5. Three-Way Hold Out Cross Validation

According to Berrar [2019] and several other scholarly articles, the holdout - method stands out as one of the simplest data resampling strategies. Given that the primary objective of this thesis is to determine the optimal approach for sharing ML models among IoT devices, little emphasis is put on the choice of the cross-validation method. The simplicity and ease of implementation of the holdout method are favored.

This study employed the three-way holdout method for model evaluation. The dataset was strategically partitioned into three distinct subsets: a training set, a validation set, and a testing set (Figure 5.3). The training set was utilized for the initial training of the ML model, allowing it to learn patterns and relationships within the data. The validation set plays a crucial role in hyperparameter tuning, as the model's hyperparameters are adjusted based on its performance on this independent subset.

This separation helps prevent overfitting hyperparameters to the testing set. Once the hyperparameters are optimized, the model's final evaluation is conducted on the testing set, ensuring an unbiased assessment of its generalization performance to previously unseen data. This method provides a systematic approach to balancing the training, hyperparameter tuning, and final evaluation stages, contributing to the robustness and reliability of the model assessment in the context of this research.

5.2.3.6. Nested Cross Validation

Researchers in Bergmeir and Benítez [2012] advocate for the utilization of k-fold cross-validation in the evaluation of time series models, emphasizing its ability to

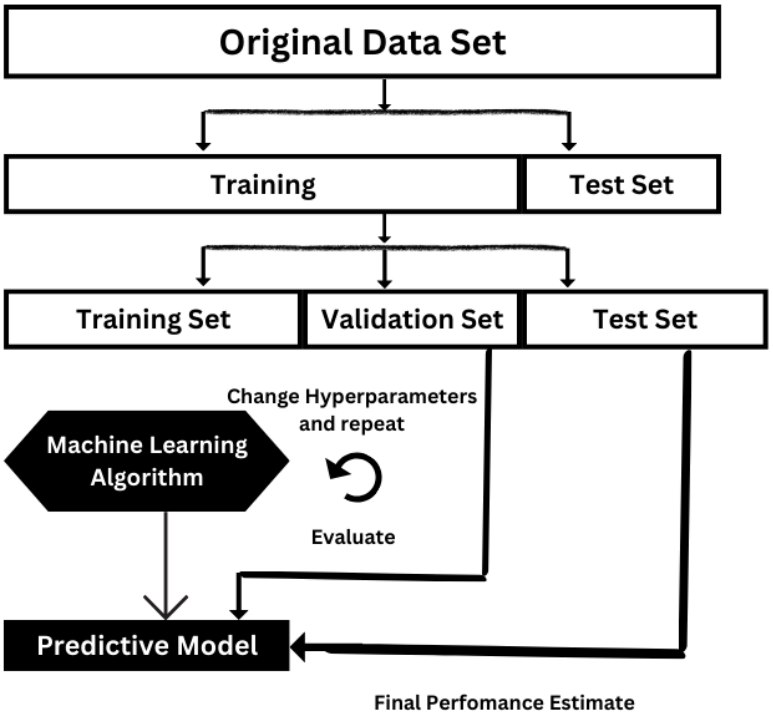


Figure 5.3.: Three Way Hold Out Cross validation flow chart.Source:Sebastianraschka.com

contribute to a more resilient model selection process. This approach is favored for its capacity to leverage all available information, thereby mitigating theoretical challenges associated with alternative methods. In a separate study, Bates et al. [2023] suggests nested cross-validation (NCV) as an attractive option for generating confidence intervals for prediction error. The research highlights NCV's consistent superiority in coverage compared to confidence intervals derived from naive cross-validation methods, making it a robust choice for obtaining reliable estimates of prediction error.

For this reasons this study utilizes NCV a variation of k-fold cross-validation. Traditional k-fold cross-validation involves splitting the dataset into k folds and using one fold for testing and the remaining k-1 folds for training in each iteration, nested cross-validation adds an additional layer of cross-validation for hyperparameter tuning.

In NCV , the outer loop performs the typical k-fold cross- validation to assessthe model's performance on different subsets of the data. Within each outer fold, there is an inner loop that is responsible for hyperparameter tuning. The inner loop typically uses another round of k-fold cross-validation to evaluate different hyperparameter configurations. See Figure 5.4

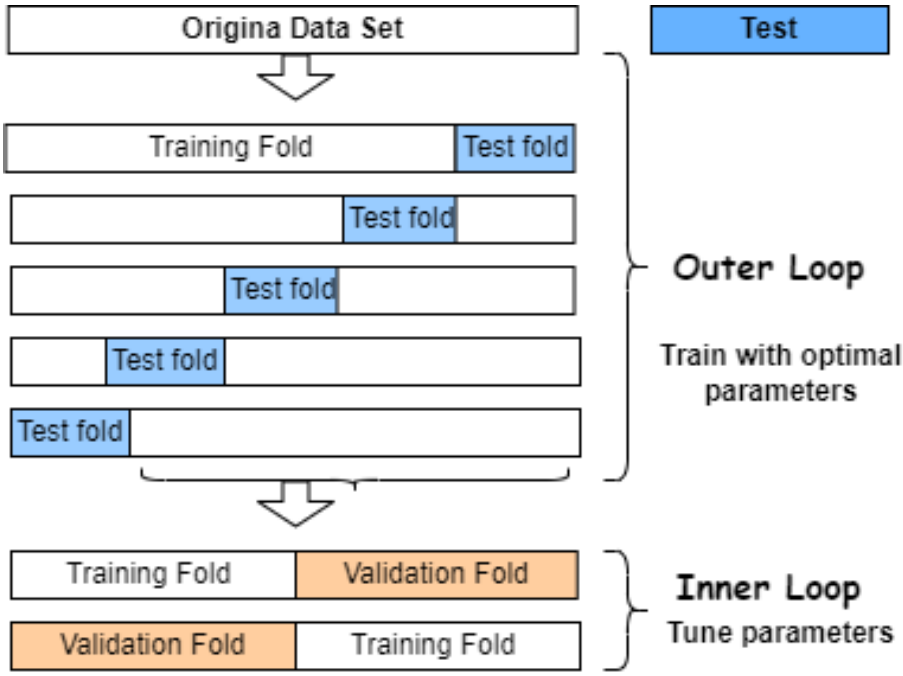


Figure 5.4.: Nested Cross validation flow chart.Source:Sebastianraschka.com

Nested cross-validation obtains a more reliable estimate of a model's performance by including both an outer loop for overall model evaluation and an inner loop for hyperparameter tuning. This helps to reduce the risk of overfitting to a specific set of hyperparameters and provides a more unbiased performance estimate.

5.2.3.7. Testing Data

Following the completion of model training, validation, and testing phases, the models underwent further evaluation on distinct datasets. These assessments encompassed testing on a station within the same cluster as the model's training data, selected based on the second-highest silhouette score within that cluster.See Table 5.1

Testing was extended to stations from other clusters, with the choice of testing stations determined by their possession of the highest silhouette scores within their respective clusters. This consistent reliance on silhouette scores for both training and testing station selection underlines the commitment to ensuring representative and well-defined datasets for model training and evaluation. It is important to note that the 2022 dataset was used for this testing phase.

5.2.3.8. Model Perfomance Evaluation using Root Mean Square Error

To evaluate the ML models this study employed RMSE. It calculates the average magnitude of errors between the model's predicted values and the actual observed values. It involves squaring the differences between predictions and true values, averaging these squared errors, and taking the square root of the result. This process

Table 5.1.: Testing Data Stations

Station Code	**Station Name**	**Cluster**	**Silhouette**	**Elevation**
c04m055e02	Xodos	C1	0.703216	1074
c05m040e13	Castelló - IES Vicent Sos Baynat	C2	0.715614	22
c03m100e02	Sant Mateu	C3	0.719095	330
c01m061e01	Forcall	C4	0.712083	692
c01m091e01	Portell de Morella	C1	0.703188	1074
c05m085e03	Orpesa Torre Bellver	C2	0.715601	22
c03m070e01	la Jana - la Pedrera	C3	0.719037	330
c02m042e02	Catí	C4	0.710289	661

ensures that both overestimations and underestimations contribute positively to the overall error measure.

The lower the RMSE, the better the model's performance, indicating smaller errors in predicting continuous outcomes. RMSE is particularly suitable for regression tasks where accuracy in predicting specific values is paramount. Its application provides a quantitative and interpretable assessment of the model's ability to make accurate predictions, facilitating the comparison and selection of models based on their predictive performance.

6. Results

This chapter will showcase the outcomes of conducting time series similarity analysis, evaluating results, implementing K-means clustering, or establishing CoI based on similarity data and geospatial components. The subsequent steps involve the development, validation, testing, and sharing of ML models.

These experiments aim to substantiate the effectiveness of the newly proposed method for sharing ML models among IoT devices. The primary goal is to substantiate that sharing ML models among CoI formed by grouping together IoT devices with similar data streams and geographical proximity is not only feasible but also advantageous. The chapter aims to provide empirical evidence supporting the effectiveness and viability of the proposed method in the context of IoT model sharing.

6.1 Time Series Similarity Analysis Results

In this study Time series similarity analysis was conducted with the primary objective of identifying and recognizing of patterns or trends embedded in time series data, enabling the extraction of valuable insights. The reason being this technique is instrumental in clustering and classification tasks, facilitating the grouping of time series data into clusters based on similarity for enhanced understanding of distinct categories.

6.1.1 Dynamic Time Warping

The DTW similarity analysis was conducted on temperature data collected from the 43 weather sensors. The primary objective was to assess the temporal patterns and similarities in temperature trends across these stations.

The results of the analysis are encapsulated in a similarity matrix, (see Figure 6.1) a comprehensive representation of pairwise DTW distances between the time series of each station. The similarity matrix provides a quantitative measure of the temporal proximity or dissimilarity between station pairs. The corresponding DTW similarity matrix heat map visually enhances the interpretation of these results, utilizing color gradients to highlight patterns within the matrix. From the figure darker shades have a value closer to zero indicating higher similarity, while lighter shades have a value closer to one hundred denoting greater dissimilarity. By examining the heat map, one can discern clusters of stations with similar temperature profiles and identify outliers or stations with distinctive patterns by examining the dendograms on the left hand side depicting clusters. This dendograms show the complex relationships within the dataset, facilitating the identification of regions or groups exhibiting consistent temperature behavior and enhancing our understanding of the

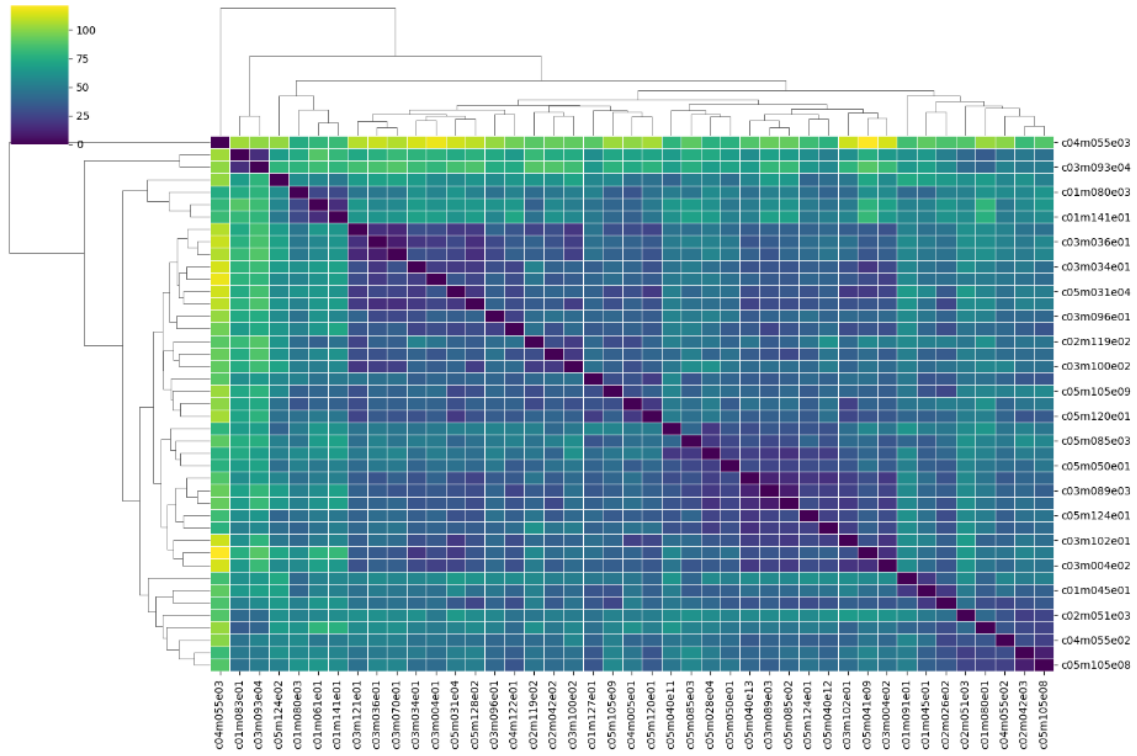


Figure 6.1.: Dynamic Time Warping Similarity Matrix Heat Map .

overall temporal dynamics across the 43 stations.

6.1.2 Spearman's Correlation Method

This study also employed the Spearman's correlation method to analyze the temporal relationships in temperature data from the 43 weather sensors. The primary aim was to assess the degree of monotonic association between the temperature time series of different stations. The outcome of the analysis is encapsulated in a correlation matrix, portraying the strength and direction of the Spearman's rank correlation coefficients for each station pair. This matrix serves as a quantitative measure of the temporal similarity or dissimilarity in temperature trends.

The accompanying heat map (see Figure 6.2)visually represents these correlation coefficients, using a spectrum of colors to convey the strength and direction of the relationships. Positive correlations are typically represented by brighter colors, while negative correlations are depicted by darker tones.As depicted in the legend values approaching 1 have a higher correlation while values approaching 0 have a lower correlation. Interpretation of the Spearman's correlation heat map involves identifying clusters of stations with similar monotonic trends and recognizing areas of divergence. By examining this graphical representation, one can gain insights into the overarching patterns and relationships within the temperature dataset, contributing to a nuanced understanding of the temporal dynamics across the 43 weather sensors.

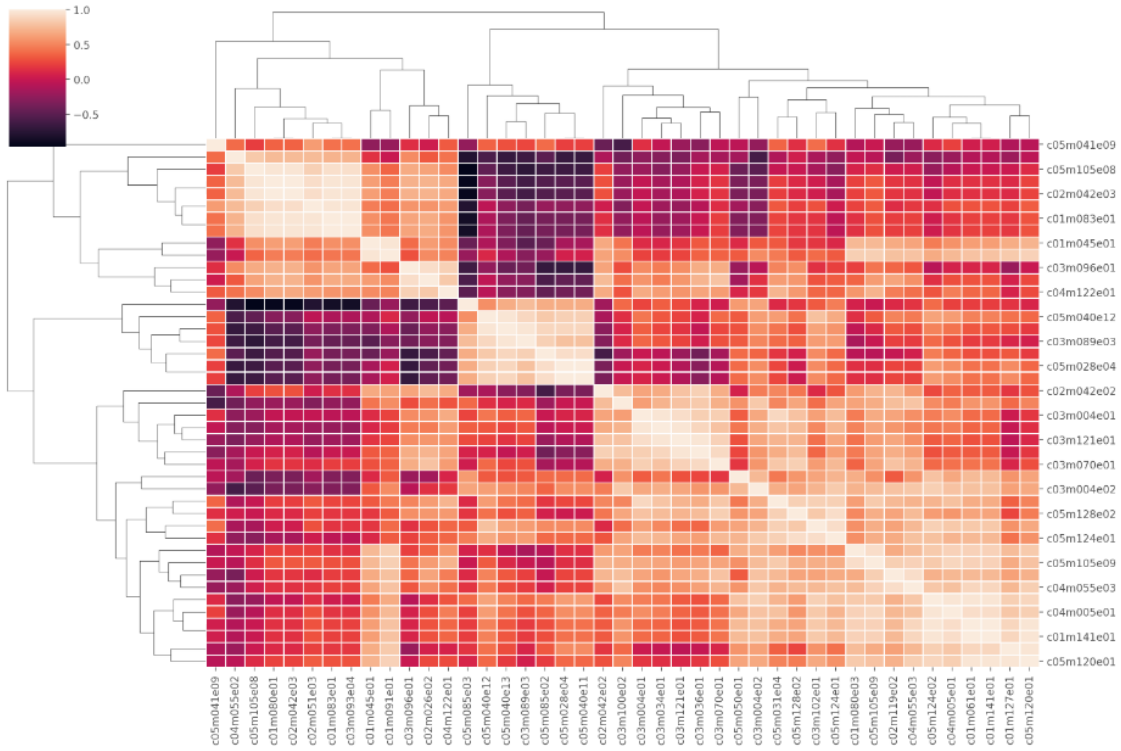


Figure 6.2.: Spearman's Correlation Similarity Matrix Heat Map .

6.1.3 Similarity Analysis Perfomance Evaluation using Silhoutte Score

To evaluate the quality of the similarity analysis results from the two different similarity analysis methods. By treating the identified groups or clusters from the similarity analysis as clusters, this study employed the Orange Data Mining tool to compute the silhouette scores.

This score, ranging from -1 to 1, provided a comprehensive assessment of how well-defined and separated the clusters were. A high silhouette score indicates that the data points were well-matched within their respective clusters and poorly matched to neighboring clusters, suggesting robust and distinct patterns in the data.

On the other hand, a silhouette score close to -1 suggests potential misclassifications, while a score around 0 indicated overlapping clusters. The average silhouette score across all data points serves as a global measure of the overall quality of the similarity analysis results.

The evaluation of similarity analysis results in this thesis revealed that Spearman's method exhibits a higher silhouette score than DTW across all the different number of clusters as seen in Table 6.1

The silhouette score, serving as a metric for assessing the quality of clustering across a range of random number of clusters or similarity analysis, indicates that

Spearman's method leads to more well-defined and internally cohesive clusters compared to DTW. This outcome suggests that, on average, the data points are better matched within their respective clusters and less matched to neighboring clusters when Spearman's method is employed.

Table 6.1.: Cluster Quality Evaluation Silhouette Scores

Cluster Number	**DTW Score**	**Spearmans Score**
2	0.653	0.692
3	0.56	0.632
4	0.543	0.65
5	0.541	0.639
6	0.506	0.659
7	0.48	0.655
8	0.469	0.66

The superior performance of Spearman's correlation method may be attributed to its rank-based correlation approach, capturing monotonic relationships in the data. In contrast, DTW, designed for time-series data with variable speeds, may exhibit lower silhouette scores due to its flexibility. This flexibility is advantageous when dealing with time-series that may exhibit variations in speed, phase shifts, or temporal distortions. However, this flexibility may also lead to challenges in clustering scenarios, as the varying speeds might introduce additional complexity in forming well-defined clusters.

These findings contribute valuable insights into the effectiveness of the similarity analysis methods, aiding in the selection of the most suitable approach based on the specific requirements and characteristics of the dataset under consideration. For this reason this thesis chose to proceed with the similarity results from Spearman's Correlation for the K-means clustering.

6.2 Elbow Method

After establishing that Spearman's method produces clusters of superior quality, the elbow method was employed to determine the most appropriate number of clusters for the K-means clustering. The elbow method involves plotting the within-cluster sum of squares, also known as inertia, against the number of clusters.

Inertia measures how tightly packed the points are within each cluster. The plot visually displays the trade-off between capturing patterns in the data and avoiding unnecessary complexity. The point on the plot resembling an elbow is crucial, as it signifies the optimal number of clusters. At this juncture, adding more clusters would result in diminishing returns in terms of reducing inertia, indicating the most effective balance between explaining variability in the data and avoiding overfitting.

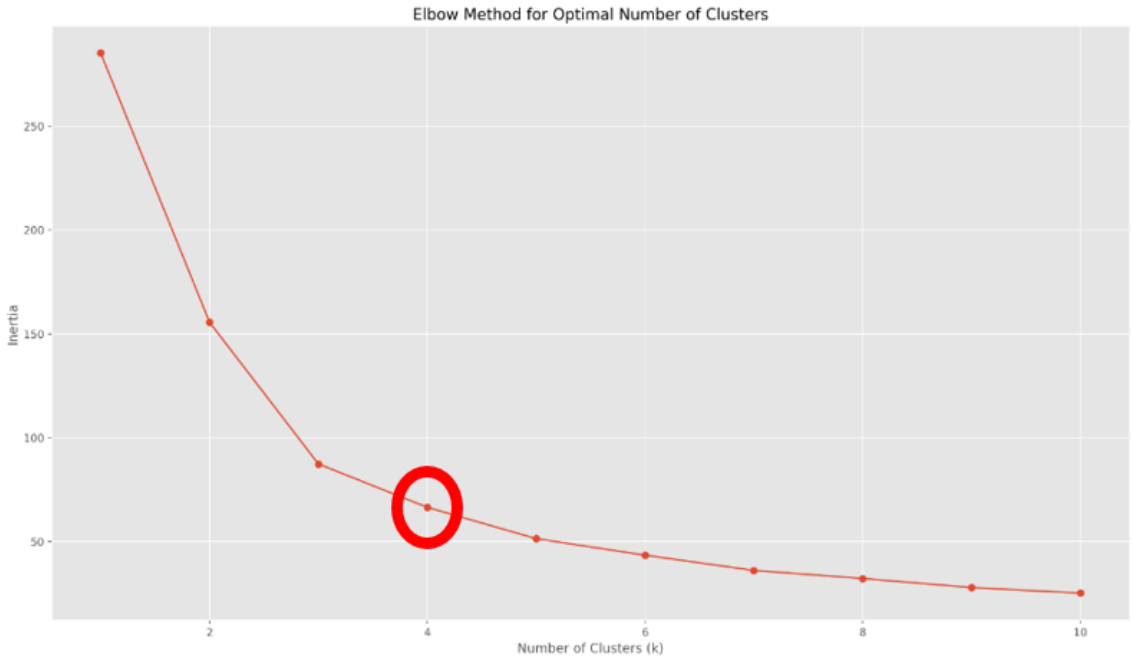


Figure 6.3.: The Elbow Plot for Optimal number of Clusters

Figure 6.3 shows the outcome of visually examining the elbow plot to determine the ideal number of clusters for the subsequent clustering of K-means. Visual analysis indicates that the optimal number of clusters is 4, identified at the juncture where the rate of change in inertia decreases as the number of clusters increases or decreases. This point signifies a balance where further adjustments to the number of clusters yield marginal improvements in reducing inertia, guiding the selection of an effective number of clusters for the subsequent analysis.

6.3 K-means Clustering

The K-means clustering algorithm, a widely used unsupervised learning technique, has been instrumental in uncovering patterns and structuring the dataset in this study.

Prior to performing the clustering of K-means, the similarity data of the weather sensors, derived from the Spearman correlation method, were augmented with geospatial information that included sensor locations and elevations. Integration of these data sets was facilitated using the Orange Data Mining tool, and subsequently, K-means clustering was applied to partition the merged data into distinct clusters, with the predetermined number of clusters (K) identified as 4 using the Elbow method.

The K-means algorithm systematically assigned data points to clusters, iteratively adjusting the centroids until convergence, thus optimizing the sum of squares within the cluster. The outcomes of this clustering process unveil the inherent structure of the dataset, delineating clear groups. The resulting clusters, depicted in the

accompanying Figure 6.3, serve as the basis for further analysis and interpretation. They are recognized as the ***CoI*** for the weather sensors, providing a more nuanced understanding of relationships and patterns within the data. These CoI in this study are invaluable for crafting, validating, testing, and ***sharing the ML models***.

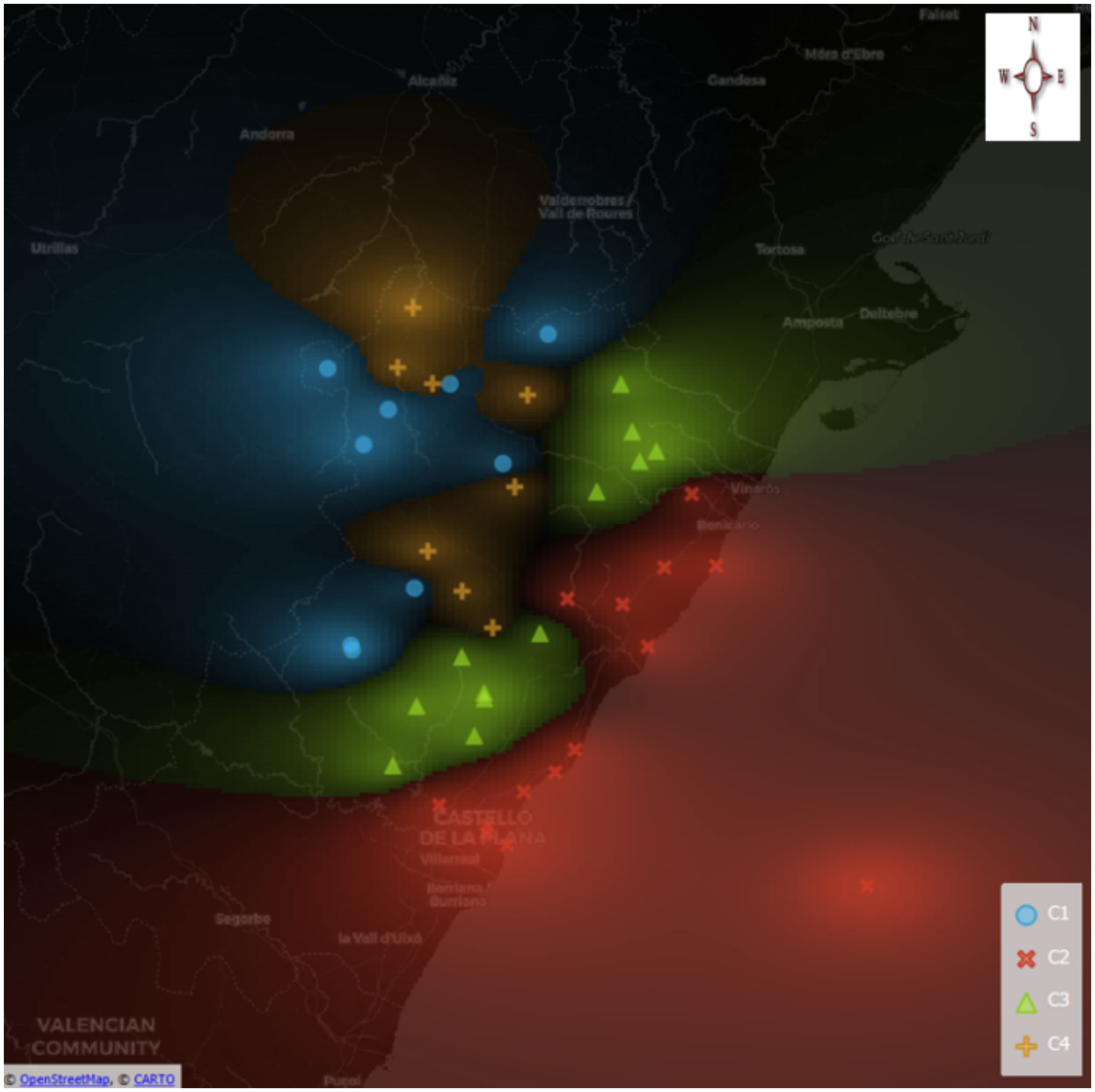


Figure 6.4.: K-Means Clustering results (***Communities of Interest***)

ML Model Results

6.3.1 Training Data

Following the application of K-means clustering, the assessment of the resultant clusters was conducted using the silhouette score. Within each cluster, stations with the highest silhouette scores were identified.

Table 6.2 subsequently shows ,the selected stations employed in the training process of ML models specific to each cluster.

Station Code	Station Name	Cluster	Silhouette	Elevation
c04m055e02	Xodos	C1	0.703216	1074
c05m040e13	Castelló- IES Vicent Sos Baynat	C2	0.715614	22
c03m100e02	Sant Mateu	C3	0.719095	330
c01m061e01	Forcall	C4	0.712083	692

Table 6.2.: Training Data Stations

The selection of stations with the highest silhouette scores for model training was aimed at maximizing the quality of the training data. This ensures that the models learn from stations that exhibit strong cohesion within their clusters and clear separation from other clusters, ultimately leading to more effective and representative ML models.

Model Sharing and Testing

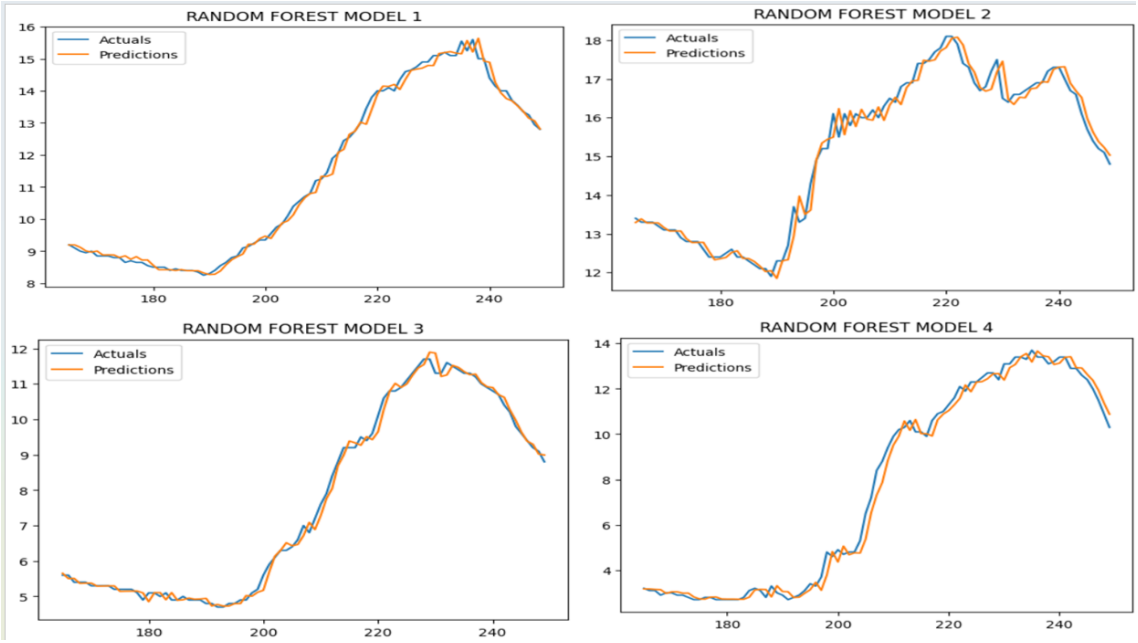


Figure 6.5.: Actuals Vs Predictions - ***Random Forest*** Models

The visual representations provided by the figures for Random Forest (Figure 6.5), LSTM with Nested Cross Validation (Figure 6.6), and LSTM with Three-way Hold Out Cross Validation (Figure 6.7) illustrate the performance of the three machine learning algorithms utilized in this study. Each algorithm underwent training with four distinct models using data from Xodos, Castellon- IES Vicent Sos Baynat, Sant Mateu, and Forcall stations in 2021 see figure 6.2, followed by testing on 2022 data from the same stations. These graphs showcase the comparison between actual values and predictions, providing a visual understanding of how effectively each model performs. These visualizations offer valuable insights into the predictive capabilities of the algorithms, enabling a qualitative evaluation of their performance

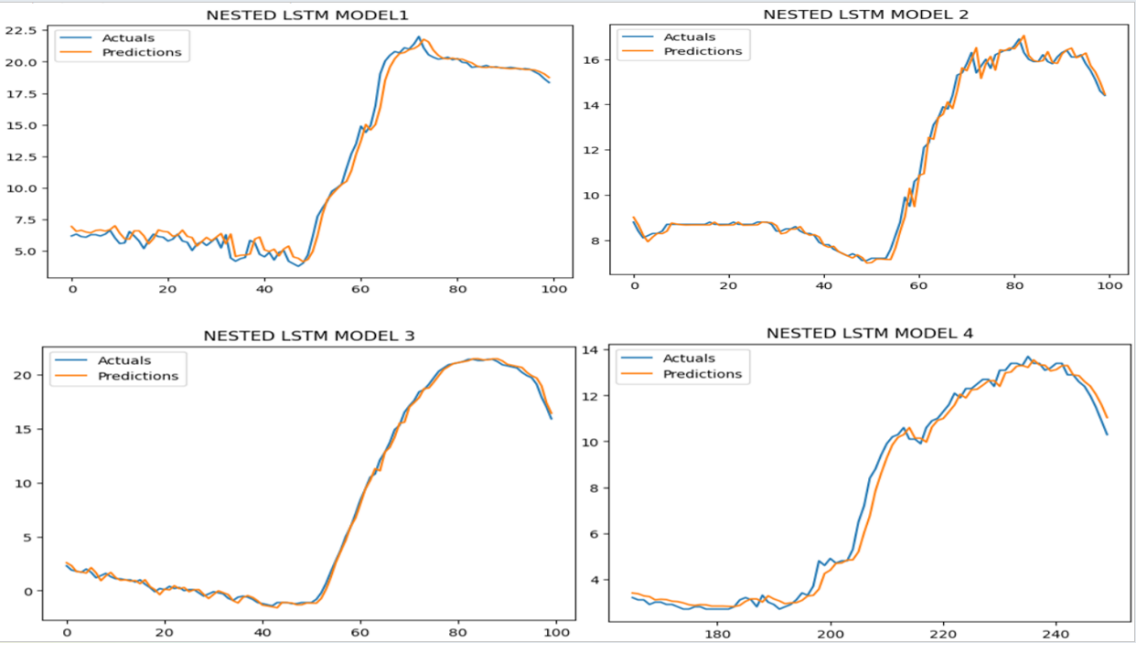


Figure 6.6.: Actuals Vs Predictions - Nested Cross Validation ***LSTM*** Models

Figure 6.7.: Actuals Vs Predictions - Three Way Hold Out ***LSTM*** Models

relative to real-world data.

Following the establishment of the four IoT CoI through K-means clustering utilizing similarity and geospatial components, the subsequent step involved the development of ML models using various algorithms outlined in the methodology section. Subsequently, data from sensors exhibiting the highest Silhouette scores within each CoI or cluster were chosen and employed to train the ML models. These trained models were then assessed using sensor data originating from clusters

distinct from the ones on which they were initially trained. Only the data from the sensors with the highest Silhouette scores were used for this training and testing process.

Table 6.3.: Comparison of ML Model performance using RMSE.

Performance Comparison				
Year	**Community**	**Random Forest**	**LSTM NCV**	**LSTM HV**
Community One: Xodos 2021 (Cluster One)				
Xodos 2022	C1	0.74	0.72	1.08
Portell 2022	C1	1.12	1.08	1.22
Bayanat 2022	C2	1.36	1.28	1.46
Sant Mateu 2022	C3	1.50	1.50	1.43
Forcall 2022	C4	1.94	1.71	2.11
Community Two: Bayanat 2021 (Cluster Two)				
Bayanat 2022	C2	0.37	0.34	0.45
Orpesa 2022	C2	0.33	0.33	0.48
Xodos 2022	C1	0.71	0.46	0.57
Sant Mateu 2022	C3	0.81	0.56	0.68
Forcall 2022	C4	2.46	1.57	2.39
Community Three: Sant Mateu 2021 (Cluster Three)				
Sant Mateu 2022	C3	0.72	0.76	0.50
Pedrera 2022	C3	0.46	0.63	0.53
Xodos 2022	C1	0.60	0.63	0.59
Bayanat 2022	C2	0.38	0.52	0.54
Forcall 2022	C4	3.04	1.88	2.33
Community Four: Forcall 2021 (Cluster Four)				
Forcall 2022	C4	0.34	0.34	1.05
Cati 2022	C4	0.33	0.33	1.05
Xodos 2022	C1	0.43	0.49	1.27
Bayanat 2022	C2	0.34	0.62	1.85
Sant Mateu 2022	C3	0.58	0.84	1.65

Table 6.3 presents the results of four distinct ML tests. Each CoI was used to train three different ML models making a total of 12 models three for each of the four CoIs.The models were subsequently tested on one member from the same community as well as one member from each of the other communities. The three different models employed different algorithms: ***Random Forest with Nested Cross Validation, Long Short Term Memory with Nested Cross Validation, and Long Short Term Memory with Three-way Hold-Out***. This

diversity of approaches serves to validate and cross-verify the hypotheses posited in this study.

In the table (Table 6.3), the orange highlighting indicates instances where the models demonstrated the expected performance. Specifically, this observation holds true for all the Models in Communities One , Two and Four. This implies that these models exhibited lower Root Mean Square Error (RMSE) values, indicative of higher prediction accuracy, when trained and tested on members of the same cluster.

The table clearly illustrates that training a model on data from a member of one cluster and subsequently testing it on data from a member of a different cluster leads to a higher RMSE, thereby indicating lower prediction accuracy. This underscores the importance of training,testing and sharing ML models within the same CoI to achieve optimal performance based on the inherent similarities within those clusters.

However, an anomaly is observed with the red highlighting see figure 6.3 in Community Three, signifying an unexpected outcome. The results of testing the Models in Community Three, which was trained on data from Cluster 3, on data from Cluster 1 and 2, the results deviated from expectations. Surprisingly, the models exhibited a lower Root Mean Square Error (RMSE) and, consequently, higher prediction accuracy when tested on data from sensors in Cluster 1 and 2 compared to its performance on members from Cluster 3, where the model was originally trained.

6.4 Discussion

In this study, three different algorithms, namely Random Forest and LSTM with NCV, as well as Three-Way Holdout Cross-Validation LSTM, were utilized. Each of these models was individually trained using data from one CoI at a time. With the creation of four CoI through K-means clustering, a total of 12 ML models were developed, consisting of three models for each of the four IoT CoIs established.

In the outcomes, it was observed that 10 out of the 12 models exhibited reduced RMSE, indicating higher predictive accuracy when assessed on data originating from the same CoI on which they were initially trained. Conversely, when these models were tested on data from sensors located in different CoIs, their prediction accuracy decreased.

In Communities One, Two and Four, all the models exhibited a consistent pattern highlighted in orange see figure 6.3, whereas only one model in Community Three adhered to the observed pattern seen in the models from other CoIs. Among the 12 models, only two did not show the same pattern as the remaining 10. Specifically, these models were the Random Forest and LSTM with NCV in Community Three highlighted in red see figure 6.3.

The results indicate that 83% of the models revealed the effectiveness of categorizing IoT devices into CoIs based on the similarity of temporal data and geospatial attributes to facilitate the sharing of ML models among devices.

6.5 Limitations

While this study contributes valuable insights into the sharing of ML models among IoT devices through CoI, several limitations warrant consideration.

The generalizability of the findings may be constrained by the specific context and conditions of this research, potentially limiting applicability to diverse deployment scenarios. The data source utilized, consisting of weather data sensors simulating real IoT devices, may not fully capture the complexities of real-world IoT device data.

Scalability remains a concern, as scaling the proposed method to larger IoT networks or accommodating dynamic changes in network composition may pose technical challenges. The absence of mobile sensor data limits the generalizability of the findings to scenarios involving stationary sensors only. The proposed method's effectiveness relies on specific assumptions and parameters, which may not always hold true in practical deployment scenarios.

Another limitation of this study is that it did not progress to the development of tinyML models for testing on real-world IoT devices. Tiny ML models, created using TensorFlow Lite, offer the potential for efficient deployment on resource-constrained devices. This avenue remains unexplored within the scope of this project, leaving room for future investigation into the implementation and performance of such models in practical IoT environments.

Resource constraints inherent to IoT devices, including processing power, memory, and energy, were also not fully explored. Addressing these limitations and further investigating their implications is crucial for the successful implementation and deployment of ML models in real-world IoT environments.

7. Conclusions and Future Work

The primary objective of this thesis was to establish an effective method for sharing ML models across IoT devices. To achieve this goal, the thesis proposed an innovative approach centered on distributing ML models among IoT-CoI based on the similarity of IoT data streams and geospatial components, specifically location and elevation. To validate the feasibility of this approach, the study adopted a cluster-based strategy to form IoT-CoIs. The initial phase involved a comprehensive similarity analysis of IoT weather sensor data streams using both DTW and Spearman's correlation methods.

Evaluation of the similarity results through the Silhouette score revealed that Spearman's correlation outperformed DTW, indicating its superiority in producing higher-quality and more coherent clusters.This superior performance may be attributed to Spearman's correlation robustly capturing monotonic relationships and being less sensitive to temporal misalignments, characteristics that are crucial for the nature of the IoT weather sensor data streams.

Due to this observation, this study proceeded by utilizing the similarity analysis outcomes derived from Spearman's correlation for K-means clustering. The optimal number of clusters, determined as four through the elbow method, guided the subsequent K-means clustering. This clustering process incorporated both the similarity results and an additional geospatial component comprising location and elevation. The resulting clusters formed the basis for IoT-CoIs, instrumental in the development, validation, testing, and sharing of ML models.

Assessment of ML model performance during the ***sharing and testing*** phases revealed a notable trend: the majority (83%) of ML models exhibited superior performance when trained, tested, and shared within the same CoI dataset. This was evidenced by lower RMSE values, indicative of higher prediction accuracy. Specifically, 10 out of 12 models followed this pattern, demonstrating improved performance when operating within the same CoI. Conversely, models trained on a different CoI exhibited poorer performance when tested on members of another CoI, reflected in higher RMSE values and lower prediction accuracy.

The findings of this study provide conclusive answers to the posed research questions. Firstly, the study successfully demonstrates that IoT devices can indeed be effectively grouped into CoI based on the similarity of both temporal data and geospatial attributes. Secondly, the investigation establishes that grouping IoT devices into CoI according to their data similarity facilitates the sharing of Tiny ML models among these devices.

Furthermore, the study has achieved its primary aim, which is to leverage geospatial components for the sharing and re-use of pre-trained ML models among IoT devices. The overarching goal of establishing geospatial zones, guided by the CoI concept, has been realized. By delineating these zones based on the inherent similarity of IoT data streams, the thesis successfully crafts and validates ML models tailored to the unique characteristics of each geospatial zone.

7.1 Future Work

To build on the favorable results supporting the proposed method of sharing ML models within IoT CoI based on the similarity of data streams and geospatial components, the future direction of this research presents several compelling avenues for exploration.

To pave the way for future advancements, this study suggests the exploration of the development of an automated recalibration system for clusters. This innovative system, driven by real-time data, holds the potential to not only refine but significantly enhance the adaptability of ML models. The aim is to enable these models to dynamically respond and evolve inline with the intricate and ever-changing patterns within the expansive landscape of IoT.

Exploring the development and testing of tinyML models on real-world IoT devices using TensorFlow Lite remains a valuable avenue. Despite not being pursued in this study, the implementation of such models holds promise for efficient deployment on resource-constrained devices. Investigating the performance and practical implications of these tiny machine learning models in IoT environments could offer valuable insights for future research and application.

To optimize ML applications,this study suggests the incorporation of a location-centric pre-trained model service. This implementation holds the potential to fine-tune models, customizing them to suit the distinctive characteristics of particular geographic regions.

To spearhead the collaborative sharing of ML models within the IoT community.This study would suggest that it is imperative to not only establish a centralized model repository but also explore the potential of decentralized repositories. This dual repository approach aims to diversify avenues for knowledge exchange and collaboration. A web service could further amplify the impact of these repositories, streamlining the publishing and accessibility of ML models. This envisioned web service would serve as a facilitator, providing a user-friendly platform for seamless sharing, exploration, and utilization of models, fostering an environment of collaborative innovation within the ever changing realm of the IoT.

There is also an opportunity for future research to broaden its scope by delving

into phenomena beyond temperature. For instance, exploring ML applications in healthcare for predicting patient outcomes or diagnosing medical conditions could significantly enhance the versatility of the proposed model-sharing method. Embracing diverse algorithms and extending the approach to classification tasks not only opens doors for advancements in fields such as image recognition or natural language processing but also amplifies the adaptability of the shared models. This expanded exploration could revolutionize various domains by harnessing the capabilities of ML for tasks beyond traditional temperature predictions and anomaly detections, fostering a more comprehensive and impactful utilization of the proposed methodology.

Conducting real-world testing, especially with mobile IoT devices positioned in dynamic environments like moving vehicles, holds the promise of providing valuable insights into the adaptability and efficacy of the proposed ML-sharing approach in practical scenarios. These deliberate initiatives collectively aspire to make significant contributions to the broader integration and scalability of the proposed method. This concerted effort aims to strengthen its applicability and impact across diverse IoT applications, ensuring its effectiveness in addressing real-world challenges and fostering widespread adoption within the IoT ecosystem.

A. Annex

A.1 Repository Title:Master Thesis

The is Master Thesis repository [1] hosts code inspired by DataCamp for multivariate time series data exploration, analysis, and similarity analysis. It includes implementations of Dynamic Time Warping and Spearman Correlation methods for similarity analysis.

The repository contains code utilized in crafting, training, validating, testing, and sharing ML models for time series forecasting simulations in real-world Internet of Things (IoT) device ML scenarios. This encompasses 12 ML models, with four models each for Random Forest and Long Short-Term Memory (LSTM) algorithms, incorporating nested cross-validation and three-way holdout validation techniques.

A.1.1 Contents

- Multivariate time series data exploration and analysis scripts.
- Dynamic Time Warping and Spearman Correlation similarity analysis implementations.
- ML model crafting, training, validation, testing, and sharing scripts for time series forecasting (Random Forest and LSTM) with Nested cross-validation and three-way holdout validation code
- Sample datasets and data preprocessing utilities

A.1.2 Features

- Exploratory data analysis (EDA) for multivariate time series data.
- Dynamic Time Warping and Spearman Correlation similarity analysis methods.
- Crafting, training, and evaluation of ML models for time series forecasting.
- Implementation of Random Forest and LSTM algorithms with nested cross-validation and three-way holdout validation.
- Simulation of real-world IoT device ML scenarios

[1] Github: `https://github.com/MikeSirya/Master-Thesis.git`

A.1.3 Dependencies

- Python (3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]).
- NumPy, pandas, matplotlib, scikit-learn, TensorFlow (for LSTM and Random Forest).
- Seaborn,statsmodels, NumPy, pandas, matplotlib and scikit-learn (for DTW Spearmans Correlation).

A.1.4 Usage

- Clone the repository to your local machine.
- Navigate to the desired script or module.
- Install dependencies using ‘pip install -r requirements.txt‘.
- Run the scripts with appropriate parameters or configurations.

A.1.5 Acknowldgements

The code in this repository draws inspiration from DataCamp tutorials and research in multivariate time series analysis and ML for IoT applications. I would like to express my gratitude to Agència Valenciana de Meteorologia (AVAMET) for providing the weather sensor data used in this project to simulate real-world IoT data stream scenarios. I also would like to acknowledge the use of the Orange data mining tool for conducting K-means clustering and generating CoI map, which greatly enhanced the analysis and visualization capabilities of this project. I am also thankful for the computing resources and collaborative environment provided by Google Colab, which facilitated the development and execution of my experiments.

Bibliography

Amaia Abanda, Usue Mori, and Jose A. Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412, March 2019. ISSN 1573-756X. doi: 10.1007/s10618-018-0596-4. URL https://doi.org/10.1007/s10618-018-0596-4.

Asmaa Achtaich, Nissrine Souissi, Raul Mazo, Camille Salinesi, and Ounsa Roudies. Designing a Framework for Smart IoT Adaptations. In Fatna Belqasmi, Hamid Harroud, Max Agueh, Rachida Dssouli, and Faouzi Kamoun, editors, *Emerging Technologies for Developing Countries*, volume 206, pages 57–66. Springer International Publishing, Cham, 2018. ISBN 978-3-319-67836-8 978-3-319-67837-5. doi: 10.1007/978-3-319-67837-5_6. URL http://link.springer.com/10.1007/978-3-319-67837-5_6. Series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.

Erwin Adi, Adnan Anwar, Zubair Baig, and Sherali Zeadally. Machine learning and data analytics for the IoT. *Neural Computing and Applications*, 32(20):16205–16233, October 2020. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-020-04874-y. URL https://link.springer.com/10.1007/s00521-020-04874-y.

Kiran Adnan, Rehan Akbar, and Siak Khor. International journal of recent technology and engineering (ijrte). 08 2019. doi: 10.35940/ijrte.B1074.0882S819.

Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering – A decade review. *Information Systems*, 53:16–38, October 2015. ISSN 03064379. doi: 10.1016/j.is.2015.04.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0306437915000733.

Juba Agoun, Yanis Bouallouche, and Mohand-Saïd Hacid. OptiClust4Rec: Unsupervised Data-Driven Methodology for Quality of Life Recommendations During a Medical Therapy (Extended Abstract).

Fachrizal Aksan, Michał Jasiński, Tomasz Sikorski, Dominika Kaczorowska, Jacek Rezmer, Vishnu Suresh, Zbigniew Leonowicz, Paweł Kostyła, Jarosław Szymańda, and Przemysław Janik. Clustering Methods for Power Quality Measurements in Virtual Power Plant. *Energies*, 14(18):5902, September 2021. ISSN 1996-1073. doi: 10.3390/en14185902. URL https://www.mdpi.com/1996-1073/14/18/5902.

Monira N. Aldelaimi, M. Anwar Hossain, and Mohammed F. Alhamid. Building Dynamic Communities of Interest for Internet of Things in Smart Cities. *Sensors*,

20(10):2986, May 2020. ISSN 1424-8220. doi: 10.3390/s20102986. URL https://www.mdpi.com/1424-8220/20/10/2986.

Ahmed A. Alwan, Allan J. Brimicombe, Mihaela Anca Ciupala, Seyed Ali Ghorashi, Andres Baravalle, and Paolo Falcarin. Time-series clustering for sensor fault detection in large-scale Cyber–Physical Systems. *Computer Networks*, 218: 109384, December 2022. ISSN 13891286. doi: 10.1016/j.comnet.2022.109384. URL https://linkinghub.elsevier.com/retrieve/pii/S1389128622004182.

Ilham Firman Ashari, Eko Dwi Nugroho, Randi Baraku, Ilham Novri Yanda, and Ridho Liwardana. Analysis of Elbow, Silhouette, Davies-Bouldin, Calinski-Harabasz, and Rand-Index Evaluation on K-Means Algorithm for Classifying Flood-Affected Areas in Jakarta. *JAIC*, 7(1):89–97, July 2023. ISSN 2548-6861. doi: 10.30871/jaic.v7i1.4947. URL https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/4947.

Hany F. Atlam, Robert J. Walters, and Gary B. Wills. Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues. *International Journal of Intelligent Computing Research*, 9(3):928–938, September 2018. ISSN 20424655. doi: 10.20533/ijicr.2042.4655.2018.0112. URL https://infonomics-society.org/wp-content/uploads/ijicr/published-papers/volume-9-2018/Internet-of-Things-State-of-the-art-Challenges-Applications-and-Open-Issues.pdf.

Luigi Atzori, Antonio Iera, and Giacomo Morabito. SIoT: Giving a Social Structure to the Internet of Things. *IEEE Communications Letters*, 15(11):1193–1195, November 2011. ISSN 1089-7798. doi: 10.1109/LCOMM.2011.090911.111340. URL http://ieeexplore.ieee.org/document/6042288/.

Fenye Bao, Ing-Ray Chen, and Jia Guo. Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems. In *2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*, pages 1–7, Mexico City, Mexico, March 2013. IEEE. ISBN 978-1-4673-5070-9 978-1-4673-5069-3. doi: 10.1109/ISADS.2013.6513398. URL http://ieeexplore.ieee.org/document/6513398/.

Romil Barthwal, Sudip Misra, and Mohammad S. Obaidat. Finding overlapping communities in a complex network of social linkages and Internet of things. *The Journal of Supercomputing*, 66(3):1749–1772, December 2013. ISSN 0920-8542, 1573-0484. doi: 10.1007/s11227-013-0973-0. URL http://link.springer.com/10.1007/s11227-013-0973-0.

Stephen Bates, Trevor Hastie, and Robert Tibshirani. Cross-Validation: What Does It Estimate and How Well Does It Do It? *Journal of the American*

Statistical Association, pages 1–12, May 2023. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2023.2197686. URL https://www.tandfonline.com/doi/full/10.1080/01621459.2023.2197686.

Fernando Bação, Victor Lobo, and Marco Painho. Self-organizing Maps as Substitutes for K-Means Clustering. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Vaidy S. Sunderam, Geert Dick Van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Computational Science – ICCS 2005*, volume 3516, pages 476–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-26044-8 978-3-540-32118-7. doi: 10.1007/11428862_65. URL http://link.springer.com/10.1007/11428862_65. Series Title: Lecture Notes in Computer Science.

Christoph Bergmeir and José M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, May 2012. ISSN 00200255. doi: 10.1016/j.ins.2011.12.028. URL https://linkinghub.elsevier.com/retrieve/pii/S0020025511006773.

Daniel Berrar. Cross-Validation. In *Encyclopedia of Bioinformatics and Computational Biology*, pages 542–545. Elsevier, 2019. ISBN 978-0-12-811432-2. doi: 10.1016/B978-0-12-809633-8.20349-X. URL https://linkinghub.elsevier.com/retrieve/pii/B978012809633820349X.

Juan Vicente Bogado Machuca, Diego Herbin Stalder Díaz, and Christian Emilio Schaerer Serra. Cluster-based LSTM models to improve Dengue cases forecast. *CLEI Electronic Journal*, 26(1), May 2023. ISSN 0717-5000. doi: 10.19153/cleiej.26.1.4. URL https://clei.org/cleiej/index.php/cleiej/article/view/580.

Fabrizio Bonacina, Eric Stefan Miele, and Alessandro Corsini. Time Series Clustering: A Complex Network-Based Approach for Feature Selection in Multi-Sensor Data. *Modelling*, 1(1):1–21, May 2020. ISSN 2673-3951. doi: 10.3390/modelling1010001. URL https://www.mdpi.com/2673-3951/1/1/1.

Leon Bornemann, Tobias Bleifuß, Dmitri Kalashnikov, Felix Naumann, and Divesh Srivastava. Data Change Exploration Using Time Series Clustering. *Datenbank-Spektrum*, 18(2):79–87, July 2018. ISSN 1618-2162, 1610-1995. doi: 10.1007/s13222-018-0285-x. URL http://link.springer.com/10.1007/s13222-018-0285-x.

T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute

error (MAE)? preprint, Numerical Methods, February 2014. URL https://gmd.copernicus.org/preprints/7/1525/2014/gmdd-7-1525-2014.pdf.

Juan Manuel Corchado. Aiot for smart territories. In *2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 1–1, Dec 2020. doi: 10.1109/IOTSMS52051.2020.9340211.

Sivo Daskalov and Ventsislav Nikolov. Prediction of univariate time series based on clustering. 2017. URL https://api.semanticscholar.org/CorpusID:208980605.

Nabil Djedjig, Djamel Tandjaoui, Faiza Medjek, and Imed Romdhani. Trust-aware and cooperative routing protocol for IoT security. *Journal of Information Security and Applications*, 52:102467, June 2020. ISSN 22142126. doi: 10.1016/j.jisa.2020.102467. URL https://linkinghub.elsevier.com/retrieve/pii/S2214212619306751.

Esma Ergüner Özkoç. Clustering of Time-Series Data. In Derya Birant, editor, *Data Mining - Methods, Applications and Systems*. IntechOpen, January 2021. ISBN 978-1-83968-318-3 978-1-83968-319-0. doi: 10.5772/intechopen.84490. URL https://www.intechopen.com/books/data-mining-methods-applications-and-systems/clustering-of-time-series-data.

Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys*, 45(1):1–34, November 2012. ISSN 0360-0300, 1557-7341. doi: 10.1145/2379776.2379788. URL https://dl.acm.org/doi/10.1145/2379776.2379788.

Ivan Farris, Roberto Girau, Leonardo Militano, Michele Nitti, Luigi Atzori, Antonio Iera, and Giacomo Morabito. Social Virtual Objects in the Edge Cloud. *IEEE Cloud Computing*, 2(6):20–28, November 2015. ISSN 2325-6095. doi: 10.1109/MCC.2015.116. URL http://ieeexplore.ieee.org/document/7397053/.

Khaled Fawagreh, Mohamed Medhat Gaber, and Eyad Elyan. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering*, 2(1):602–609, December 2014. ISSN 2164-2583. doi: 10.1080/21642583.2014.956265. URL http://www.tandfonline.com/doi/abs/10.1080/21642583.2014.956265.

T Gauthier. Detecting Trends Using Spearman's Rank Correlation Coefficient. *Environmental Forensics*, 2(4):359–362, December 2001. ISSN 15275922. doi: 10.1006/enfo.2001.0061. URL http://linkinghub.elsevier.com/retrieve/pii/S1527592201900618.

Martín González-Soto, Rebeca P. Díaz-Redondo, Manuel Fernández-Veiga, Bruno Fernández-Castro, and Ana Fernández-Vilas. Decentralized and collaborative machine learning framework for IoT. *Computer Networks*, 239:110137, February 2024. ISSN 13891286. doi: 10.1016/j.comnet.2023.110137. URL https://linkinghub.elsevier.com/retrieve/pii/S1389128623005820.

Carlos Granell, Andreas Kamilaris, Alexander Kotsev, Frank O Ostermann, and Sergio Trilles. Internet of things. *Manual of digital earth*, pages 387–423, 2020.

Sahibzada Saadoon Hammad, Ditsuhi Iskandaryan, and Sergio Trilles. An unsupervised tinyml approach applied to the detection of urban noise anomalies under the smart cities environment. *Internet of Things*, 23:100848, 2023.

Agnieszka Jastrzebska, Gonzalo Nápoles, Yamisleydi Salgueiro, and Koen Vanhoof. Evaluating time series similarity using concept-based models. *Knowledge-Based Systems*, 238:107811, February 2022. ISSN 09507051. doi: 10.1016/j.knosys.2021.107811. URL https://linkinghub.elsevier.com/retrieve/pii/S0950705121010108.

Ali Javed, Byung Suk Lee, and Donna M. Rizzo. A benchmark study on time series clustering. *Machine Learning with Applications*, 1:100001, September 2020. ISSN 26668270. doi: 10.1016/j.mlwa.2020.100001. URL https://linkinghub.elsevier.com/retrieve/pii/S2666827020300013.

Ali Javed, Donna M. Rizzo, Byung Suk Lee, and Robert Gramling. SOMTimeS: Self Organizing Maps for Time Series Clustering and its Application to Serious Illness Conversations, August 2021. URL http://arxiv.org/abs/2108.11523. arXiv:2108.11523 [cs].

Nickson M. Karie, Nor Masri Sahri, and Paul Haskell-Dowland. IoT Threat Detection Advances, Challenges and Future Directions. In *2020 Workshop on Emerging Technologies for Security in IoT (ETSecIoT)*, pages 22–29, Sydney, Australia, April 2020. IEEE. ISBN 978-1-72818-019-9. doi: 10.1109/ETSecIoT50046.2020.00009. URL https://ieeexplore.ieee.org/document/9097762/.

Eamonn Keogh and Jessica Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, August 2005. ISSN 0219-1377, 0219-3116. doi: 10.1007/s10115-004-0172-7. URL http://link.springer.com/10.1007/s10115-004-0172-7.

Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, March 2005.

ISSN 0219-1377, 0219-3116. doi: 10.1007/s10115-004-0154-9. URL http://link.springer.com/10.1007/s10115-004-0154-9.

A. Kianimajd, M.G. Ruano, P. Carvalho, J. Henriques, T. Rocha, S. Paredes, and A.E. Ruano. Comparison of different methods of measuring similarity in physiologic time series. *IFAC-PapersOnLine*, 50(1):11005–11010, July 2017. ISSN 24058963. doi: 10.1016/j.ifacol.2017.08.2479. URL https://linkinghub.elsevier.com/retrieve/pii/S2405896317333967.

Hyojeoung Kim, Sujin Park, and Sahm Kim. Time-series clustering and forecasting household electricity demand using smart meter data. *Energy Reports*, 9:4111–4121, December 2023. ISSN 23524847. doi: 10.1016/j.egyr.2023.03.042. URL https://linkinghub.elsevier.com/retrieve/pii/S2352484723002731.

Evangelos A. Kosmatos, Nikolaos D. Tselikas, and Anthony C. Boucouvalas. Integrating RFIDs and Smart Objects into a UnifiedInternet of Things Architecture. *Advances in Internet of Things*, 01(01):5–12, 2011. ISSN 2161-6817, 2161-6825. doi: 10.4236/ait.2011.11002. URL http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ait.2011.11002.

Peter Laurinec and Mária Lucká. Clustering-based forecasting method for individual consumers electricity load using time series representations. *Open Computer Science*, 8(1):38–50, July 2018. ISSN 2299-1093. doi: 10.1515/comp-2018-0006. URL https://www.degruyter.com/document/doi/10.1515/comp-2018-0006/html.

Lianhong Ding, Peng Shi, and Bingwu Liu. The clustering of Internet, Internet of Things and social network. In *2010 Third International Symposium on Knowledge Acquisition and Modeling*, pages 417–420, Wuhan, China, October 2010. IEEE. ISBN 978-1-4244-8004-3. doi: 10.1109/KAM.2010.5646274. URL http://ieeexplore.ieee.org/document/5646274/.

Akanksha Maurya, Alper Sinan Akyurek, Baris Aksanli, and Tajana Simunic Rosing. Time-series clustering for data analysis in Smart Grid. In *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 606–611, Sydney, Australia, November 2016. IEEE. ISBN 978-1-5090-4075-9. doi: 10.1109/SmartGridComm.2016.7778828. URL http://ieeexplore.ieee.org/document/7778828/.

Nicolas Andres Melo Riveros, Bayron Alexis Cardenas Espitia, and Lilia Edith Aparicio Pico. Comparison between K-means and Self-Organizing Maps algorithms used for diagnosis spinal column patients. *Informatics in Medicine Unlocked*, 16:100206, 2019. ISSN 23529148. doi: 10.1016/j.imu.2019.100206. URL https://linkinghub.elsevier.com/retrieve/pii/S235291481930098X.

Jorge Mira, Iván Moreno, Hervé Bardisbanian, and Jesús Gorroñogoitia. Machine Learning (ML) as a Service (MLaas): Enhancing IoT with Intelligence, Adaptive Online Deep and Reinforcement Learning, Model Sharing, and Zero-knowledge Model Verification. In *Shaping the Future of IoT with Edge Intelligence*, pages 63–93. River Publishers, New York, 1 edition, November 2023. ISBN 978-1-03-263240-7. doi: 10.1201/9781032632407-6. URL https://www.taylorfrancis.com/books/9781032632407/chapters/10.1201/9781032632407-6.

Sudip Misra, Romil Barthwal, and Mohammad S. Obaidat. Community detection in an integrated Internet of Things and social network architecture. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 1647–1652, Anaheim, CA, USA, December 2012. IEEE. ISBN 978-1-4673-0921-9 978-1-4673-0920-2 978-1-4673-0919-6. doi: 10.1109/GLOCOM.2012.6503350. URL http://ieeexplore.ieee.org/document/6503350/.

Lam Duc Nguyen, Shashi Raj Pandey, Soret Beatriz, Arne Broering, and Petar Popovski. A Marketplace for Trading AI Models based on Blockchain and Incentives for IoT Data, December 2021. URL http://arxiv.org/abs/2112.02870. arXiv:2112.02870 [cs].

Michele Nitti, Luigi Atzori, and Irena Pletikosa Cvijikj. Friendship Selection in the Social Internet of Things: Challenges and Possible Strategies. *IEEE Internet of Things Journal*, 2(3):240–247, June 2015. ISSN 2327-4662. doi: 10.1109/JIOT.2014.2384734. URL http://ieeexplore.ieee.org/document/6994231/.

Pinyarash Pinyoanuntapong, Wesley Houston Huff, Minwoo Lee, Chen Chen, and Pu Wang. Toward Scalable and Robust AIoT via Decentralized Federated Learning. *IEEE Internet of Things Magazine*, 5(1):30–35, March 2022. ISSN 2576-3180, 2576-3199. doi: 10.1109/IOTM.006.2100216. URL https://ieeexplore.ieee.org/document/9773089/.

Partha Pratim Ray. A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1595–1623, April 2022. ISSN 13191578. doi: 10.1016/j.jksuci.2021.11.019. URL https://linkinghub.elsevier.com/retrieve/pii/S1319157821003335.

Abdul Razaque, Marzhan Abenova, Munif Alotaibi, Bandar Alotaibi, Hamoud Alshammari, Salim Hariri, and Aziz Alotaibi. Anomaly Detection Paradigm for Multivariate Time Series Data Mining for Healthcare. *Applied Sciences*, 12 (17):8902, September 2022. ISSN 2076-3417. doi: 10.3390/app12178902. URL https://www.mdpi.com/2076-3417/12/17/8902.

Sofien Resifi, Hassan Hassan, and Khalil Drira. Adapting Deep Learning models to IoT environments. In *2022 5th Conference on Cloud and Internet of Things*

(CIoT), pages 67–74, Marrakech, Morocco, March 2022. IEEE. ISBN 978-1-66547-964-6. doi: 10.1109/CIoT53061.2022.9766636. URL https://ieeexplore.ieee.org/document/9766636/.

Caleb Robinson, Anthony Ortiz, Juan M. Lavista Ferres, Brandon Anderson, and Daniel E. Ho. Temporal Cluster Matching for Change Detection of Structures from Satellite Imagery, June 2021. URL http://arxiv.org/abs/2103.09787. arXiv:2103.09787 [cs].

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Subhash Sagar, Adnan Mahmood, Quan Z. Sheng, Munazza Zaib, and Wei Emma Zhang. Towards a Machine Learning-driven Trust Evaluation Model for Social Internet of Things: A Time-aware Approach, February 2021. URL http://arxiv.org/abs/2102.10998. arXiv:2102.10998 [cs].

Joan Serrà and Josep Lluis Arcos. An Empirical Evaluation of Similarity Measures for Time Series Classification. *Knowledge-Based Systems*, 67:305–314, September 2014. ISSN 09507051. doi: 10.1016/j.knosys.2014.04.035. URL http://arxiv.org/abs/1401.3973. arXiv:1401.3973 [cs, stat].

Saima Shahab, Parul Agarwal, Tabish Mufti, and Ahmed J. Obaid. SIoT (Social Internet of Things): A Review. In Simon Fong, Nilanjan Dey, and Amit Joshi, editors, *ICT Analysis and Applications*, volume 314, pages 289–297. Springer Nature Singapore, Singapore, 2022. ISBN 9789811656545 9789811656552. doi: 10.1007/978-981-16-5655-2_28. URL https://link.springer.com/10.1007/978-981-16-5655-2_28. Series Title: Lecture Notes in Networks and Systems.

Congming Shi. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. 2021.

Congming Shi, Bingtao Wei, Shoulin Wei, Wen Wang, Hai Liu, and Jialei Liu. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm, 08 2020.

Manie Tadayon and Yumi Iwashita. A clustering approach to time series forecasting using neural networks: A comparative study on distance-based vs. feature-based clustering methods, March 2021. URL http://arxiv.org/abs/2001.09547. arXiv:2001.09547 [cs, stat].

Sergio Trilles, Alejandro Luján, Óscar Belmonte, Raúl Montoliu, Joaquín Torres-Sospedra, and Joaquín Huerta. Senviro: A sensorized platform proposal using open hardware and open standards. *Sensors*, 15(3):5555–5582, 2015.

Sergio Trilles, Andrea Calia, Óscar Belmonte, Joaquín Torres-Sospedra, Raúl Montoliu, and Joaquín Huerta. Deployment of an open sensorized platform in a smart city context. *Future Generation Computer Systems*, 76:221–233, 2017.

Sergio Trilles, Joaquín Torres-Sospedra, Óscar Belmonte, F. Javier Zarazaga-Soria, Alberto González-Pérez, and Joaquín Huerta. Development of an open sensorized platform in a smart agriculture context: A vineyard support system for monitoring mildew disease. *Sustainable Computing: Informatics and Systems*, 28:100309, 2020. ISSN 2210-5379. doi: https://doi.org/10.1016/j.suscom.2019.01.011. URL https://www.sciencedirect.com/science/article/pii/S2210537918302270.

Sergio Trilles, Sahibzada Saadoon Hammad, and Ditsuhi Iskandaryan. Anomaly detection based on artificial intelligence of things: A systematic literature mapping. *Internet of Things*, 25:101063, 2024. ISSN 2542-6605. doi: https://doi.org/10.1016/j.iot.2024.101063. URL https://www.sciencedirect.com/science/article/pii/S2542660524000052.

Hristos Tyralis and Georgia Papacharalampous. Variable Selection in Time Series Forecasting Using Random Forests. *Algorithms*, 10(4):114, October 2017. ISSN 1999-4893. doi: 10.3390/a10040114. URL http://www.mdpi.com/1999-4893/10/4/114.

Alfred Ultsch and Fabian Morchen. ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM.

M. Van Onsem, D. De Paepe, S. Vanden Hautte, P. Bonte, V. Ledoux, A. Lejon, F. Ongenae, D. Dreesen, and S. Van Hoecke. Hierarchical pattern matching for anomaly detection in time series. *Computer Communications*, 193:75–81, September 2022. ISSN 01403664. doi: 10.1016/j.comcom.2022.06.027. URL https://linkinghub.elsevier.com/retrieve/pii/S0140366422002298.

Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-Based Clustering for Time Series Data. *Data Mining and Knowledge Discovery*, 13(3):335–364, September 2006. ISSN 1384-5810, 1573-756X. doi: 10.1007/s10618-005-0039-x. URL http://link.springer.com/10.1007/s10618-005-0039-x.

T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, November 2005. ISSN 00313203. doi: 10.1016/j.

patcog.2005.01.025. URL https://linkinghub.elsevier.com/retrieve/pii/S0031320305001305.

Li C Xia, Joshua A Steele, Jacob A Cram, Zoe G Cardon, Sheri L Simmons, Joseph J Vallino, Jed A Fuhrman, and Fengzhu Sun. Extended local similarity analysis (eLSA) of microbial community and other time series data with replicates. *BMC Systems Biology*, 5(S2):S15, December 2011. ISSN 1752-0509. doi: 10.1186/1752-0509-5-S2-S15. URL https://bmcsystbiol.biomedcentral.com/articles/10.1186/1752-0509-5-S2-S15.

Tianqi Yu and Xianbin Wang. Real-Time Data Analytics in Internet of Things Systems. In Yu-Chu Tian and David Charles Levy, editors, *Handbook of Real-Time Computing*, pages 1–28. Springer Singapore, Singapore, 2020. ISBN 978-981-4585-87-3. doi: 10.1007/978-981-4585-87-3_38-1. URL http://link.springer.com/10.1007/978-981-4585-87-3_38-1.

Hao Yue, Linke Guo, Ruidong Li, Hitoshi Asaeda, and Yuguang Fang. DataClouds: Enabling Community-Based Data-Centric Services Over the Internet of Things. *IEEE Internet of Things Journal*, 1(5):472–482, October 2014. ISSN 2327-4662, 2372-2541. doi: 10.1109/JIOT.2014.2353629. URL https://ieeexplore.ieee.org/document/6888483/.

Xianfei Zhou, Kai Xu, Naiyu Wang, Jianlin Jiao, Ning Dong, Meng Han, and Hao Xu. A Secure and Privacy-Preserving Machine Learning Model Sharing Scheme for Edge-Enabled IoT. *IEEE Access*, 9:17256–17265, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3051945. URL https://ieeexplore.ieee.org/document/9326411/.

Hanlin Zhu, Yongxin Zhu, Di Wu, Hui Wang, Li Tian, Wei Mao, Can Feng, Xiaowen Zha, Guobao Deng, Jiayi Chen, Tao Liu, Xinyu Niu, Kuen Hung Tsoi, and Wayne Luk. Correlation Coefficient Based Cluster Data Preprocessing and LSTM Prediction Model for Time Series Data in Large Aircraft Test Flights. In Meikang Qiu, editor, *Smart Computing and Communication*, volume 11344, pages 376–385. Springer International Publishing, Cham, 2018. ISBN 978-3-030-05754-1 978-3-030-05755-8. doi: 10.1007/978-3-030-05755-8_37. URL http://link.springer.com/10.1007/978-3-030-05755-8_37. Series Title: Lecture Notes in Computer Science.

Seyedjamal Zolhavarieh, Saeed Aghabozorgi, and Ying Wah Teh. A Review of Subsequence Time Series Clustering. *The Scientific World Journal*, 2014:1–19, 2014. ISSN 2356-6140, 1537-744X. doi: 10.1155/2014/312521. URL http://www.hindawi.com/journals/tswj/2014/312521/.

Masters Program in **Geospatial Technologies**

Supported by
Education and Culture
ERASMUS MUNDUS