

# Masters Program in **Geospatial Technologies**

---



**Energy Consumption Modelling in IoT-surveyed Peacekeeping Missions**

---

Rodrigo Malagón Rodríguez

---

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

# Energy Consumption Modelling in IoT-surveyed Peacekeeping Missions

*MSc. Geospatial Technologies*

Rodrigo Malagón Rodríguez

*Supervised by:*

Prof. Dr. Jorge Mateu Mahiques

Department of Mathematics

Universitat Jaume I

Prof. Dr. Edzer Pebesma

Institute for Geoinformatics

Universität Münster

Prof. Dr. Ana Cristina Costa

NOVA Information Management School

Universidade NOVA de Lisboa

Castelló de la Plana

February 15, 2025

# Acknowledgements

I would like to express my sincere gratitude to Prof. Dr. Jorge Mateu Mahiques for his guidance and constant engagement in this process. To Prof. Dr. Edzer Pebesma for conducting the interesting course on Spatial Data Science with R. To Dr. Prof. Ana Cristina Costa for her valuable comments on this work.

To Dr. Laura Diaz Sanchez from the United Nations Global Service Centre for the possibility to work on this project. To the UNGSC matter experts, for the mindful insights on the challenges faced and the data used in this work.

To the MSc. Geospatial Technologies board, for giving me the opportunity to further develop my career and my life. To the university, as a space and a collection of people and their believes.

To my colleagues, with special attention to Jhony, Hager, and Rebeca. This path would mean nothing without their teachings and their friendship. To my family and my friends, for the love and the help that I have always received from them. To Sofia, for all her kind encouragement and her invaluable support.

# Abstract

The Internet of Things (IoT) framework enables the monitoring of power consumption in electrical devices. Different Machine Learning (ML) techniques can be leveraged in this context to perform energy usage prediction. This work presents three main temporal and spatial methods –Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), and Besag-York-Mollié (BYM)– to build forecast models for electricity consumption in IoT-surveyed peacekeeping mission camps.

The adoption of Deep Learning (DL) to perform forecast tasks has recently dominated the literature in the IoT context. We built the proposed models with a baseline LSTM approach. However, further insights were extracted with other classical methods coming from Statistics. In particular, this work uses the Gaussian adaptation of the BYM model to estimate residuals. With this technique, we aim to enhance accuracy and interpretability. Moreover, we studied the role of neighbourhood features among sensors in increasing the models' effectiveness and stability.

One of the main challenges in this setting is dealing with noisy readings coming from local network issues or user manipulation. This project presents different data transformations to improve data quality, clean incoming outliers, and tailor it for prediction.

**Keywords:** Internet of Things, power consumption, peacekeeping missions, neural networks, autoregressive models, Bayesian inference.

# Contents

Acknowledgements . . . . .	i
Abstract . . . . .	ii
List of Tables . . . . .	vi
List of Figures . . . . .	viii
List of Acronyms . . . . .	viii
<b>1 Introduction</b>	<b>1</b>
1.I Motivation . . . . .	1
1.II Related Work . . . . .	2
1.III Aim and Research Questions . . . . .	5
1.IV Outline . . . . .	5
<b>2 Problem and Data Description</b>	<b>6</b>
2.I Data Structure . . . . .	6
2.II Project Scope and Data Selection . . . . .	8
2.III Data Processing . . . . .	9
2.IV Software . . . . .	15
<b>3 Methods</b>	<b>16</b>
3.I Deep Learning Models . . . . .	16
3.I.i Recurrent Neural Networks . . . . .	17
3.I.ii Long Short-Term Memory Networks . . . . .	20
3.I.iii Optimization and Implementation . . . . .	22
3.II Regressive Temporal Models . . . . .	23
3.II.i AR and MA Models . . . . .	23
3.II.ii Causality and Invertibility . . . . .	24
3.II.iii ARMA Models . . . . .	25

3.II.iv	ARIMA models . . . . .	25
3.II.v	Optimization and Implementation . . . . .	26
3.III	Spatial Models . . . . .	27
3.III.i	Besag-York-Mollié Models . . . . .	28
3.III.ii	Inference and Implementation . . . . .	29
<b>4</b>	<b>Model Implementations</b>	<b>32</b>
4.I	Data Preparation . . . . .	32
4.II	Model Design . . . . .	37
<b>5</b>	<b>Results</b>	<b>40</b>
5.I	Model Results . . . . .	40
5.II	Results Discussion . . . . .	47
<b>6</b>	<b>Future Work and Conclusions</b>	<b>51</b>
6.I	Future Work . . . . .	51
6.II	Conclusions . . . . .	52
	References . . . . .	54

# List of Tables

2.1 Raw Data Attributes. . . . .	7
2.2 Outlier threshold parameters. . . . .	11
2.3 Maximum outlier difference values per sensor (largest 10). . . . .	11
2.4 Percentage of data manipulation by sensor and category (10 most processed sensors). . . . .	13
2.5 R packages used for the project. . . . .	15
3.1 Behaviour of the ACF and PACF plots for ARMA models (Shumway & Stoffer, 2017, Table 3.1). . . . .	26
4.1 Features extracted per time series. . . . .	33
4.2 Mean range values from hourly semivariograms in four dates. . . . .	34
4.3 LSTM model design. . . . .	37
4.4 Parameter settings for LSTM models. . . . .	38
4.5 Prior settings for precisions $\tau_u$ , $\tau_v$ , and $\tau_\epsilon$ . . . . .	39
5.1 Performance by type of model. . . . .	42
5.2 RMSE improvement by AutoARIMA for four sensors with $p > 5$ . . . . .	48
5.3 Correlation analysis of test RMSE (models with spatial feature) (* $p < 0.05$ , ** $p < 0.01$ ). . . . .	50
5.4 Correlation analysis of test RMSE (models without spatial feature) (* $p < 0.05$ , ** $p < 0.01$ ). . . . .	50

# List of Figures

1.1	UNISFA mission map on May 2023 (UN Geospatial Information Section, 2023).	3
2.1	Example of an SQL query from the local database. Power consumption data for the whole UNISFA mission was retrieved ensuring sequential time ordering using the timestamp field. Here the query is limited to 100 entries to simplify the example.	7
2.2	Spatial distribution of power consumption sensors at the Abyei camp, with selection of isolated sensors inside a buffer distance of 4 m.	9
2.3	Distribution of power consumption differenced values $y'_t$ of a sensor.	12
2.4	Example of data transformation for an individual sensor from cumulative raw values, labelled as correct and incorrect according to the quality check rules (left), to clean time and consumption values (right).	14
2.5	Data selection and processing workflow.	14
3.1	Unfolding of an RNN hidden layer. The black square refers to a delayed passing of the hidden state $h$ on each time step (Goodfellow, Bengio, & Courville, 2016).	17
3.2	Calculation of the cross-entropy loss (here denoted by $H$ ) in a land cover classification task, using input spectral data $x_t$ of dimension $d = n_b \times p^2$ , that is, $n_b$ spectral bands of raster images with size $p \times p$ (Rußwurm & Körner, 2017).	19
3.3	Inner structure of an LSTM cell. Here $f$ represents the forget gate, $i$ the input gate, $m$ the modulation gate, and $o$ the output gate. Figure adapted from (Körner & Rußwurm, 2021).	21
4.1	Semivariogram estimation for September 1st, 2023 at 18:00.	34

4.2	Overlay of the two voronoi collections (10m and 30m buffer distances) and their corresponding adjacency graph structures. . . . .	36
4.3	Data preparation and modelling workflow. . . . .	39
5.1	Comparison of models fitting the test dataset for an individual sensor. . . . .	41
5.2	RMSE (left) and $R^2$ (right) comparison across models with spatial feature. .	43
5.3	RMSE (left) and $R^2$ (right) comparison across models without spatial feature.	43
5.4	Test RMSE distribution across sensors (models with spatial feature). . . . .	44
5.5	Test RMSE distribution across sensors (models without spatial feature). . .	44
5.6	BYM spatial effects with the spatial feature for models with a buffer distance of 30m (top) and 10m (bottom). . . . .	45
5.7	BYM spatial effects without the spatial feature for models with a buffer distance of 30m (top) and 10m (bottom). . . . .	46
5.8	Distribution of order values of AutoARIMA with (left) and without (right) the spatial feature. . . . .	48
5.9	Spatial distribution of RMSE for the LSTM+AutoARIMA+BYM(10) models with (left) and without (right) the spatial feature. . . . .	49

# List of Acronyms

ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
BYM	Besag York Mollié
DL	Deep Learning
INLA	Integrated Nested Laplace Approximation
IoT	Internet of Things
IoT-ECP	Internet of Things Energy Consumption Prediction
LSTM	Long Short-Term Memory
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
OSM	OpenStreetMap
RNN	Recurrent Neural Network
UN	United Nations
UNGSC	United Nations Global Service Centre

# **Chapter 1**

## **Introduction**

### **1.1 Motivation**

Peacekeeping missions have proven to be central in providing support in humanitarian struggles. According to a research presented by Professor Lise Howard of Georgetown University on peer-reviewed sources on this matter for the UN Story channel, in two out of three cases, peacekeeping missions have successfully implemented their mandates (United Nations, 2022). Apart from these raw figure, there is evidence of the positive effect of UN personnel on the protection of civilians in conflicts.

A study by Hultman, Kathman, and Shannon (2013) reviews the influence of the presence of UN peacekeeping forces –accounting for different types of personnel— on the reduction of civilian deaths in African armed conflicts from 1991 to 2008. As shown by the authors, when assigned suitable tasks and deployed in large scale, peacekeeping personnel significantly undermined civilian killings. Notably, particular tactics by the UN forces, such as divide-and-conquer maneuvers, diminish the risk of one-sided violence among factions, so that truces and stand-down become more likely (Hultman et al., 2013).

On the other hand, the successful coordination of the missions' tasks by logistic non-governmental agencies relies on the careful and optimal use of resources, among which electrical power is of central importance. Within this context, identifying consumption patterns inside peacekeeping mission camps allows for effective allocation of assets and the development of energy conservation strategies.

Moreover, from a green perspective, Sustainable Development Goal 7 (Affordable and Clean Energy) stresses the generalised need to reduce the environmental impact of energy consumption (United Nations, 2023). Thus, to be consistent with this objective, optimal

energy consumption arises as a key factor for UN peacekeeping missions.

To this end, the United Nations Global Service Centre (UNGSC) –in charge of the logistics of several UN peacekeeping missions around the globe— has deployed networks of sensors within UN mission camps to track down energy consumption by its personnel (United Nations Global Service Centre, 2023). The installed sensing devices operate in the Internet of Things (IoT) framework. Moreover, through its Field Remote Infrastructure Monitoring (FRIM) initiative, the agency has integrated features for automation and remote control. An example of a current UN peacekeeping mission with a sensor network of this type is the United Nations Interim Security Force for Abyei (UNISFA) mission (United Nations Global Service Centre, 2024).

In its resolution 1990 (2011), the UN Security Council decided to establish the UNISFA mission, after a demilitarization agreement was reached between the Sudan Government and the Sudan People's Liberation Movement. Its main tasks included thereupon the redeployment of military forces from both factions and the provision of humanitarian aid for civilians in the Abyei region (United Nations, 2011). Though defined for an initial period of 6 months, the UNISFA mandate has been subsequently extended (United Nations, 2024). The distribution of the UNISFA camps and peacekeeping forces as of May 2023 is shown in Figure 1.1.

In this project, we developed a model of energy consumption for the Abyei camp, within the UNISFA mission. The main goal of the model is to forecast power demand at the camp level in a daily basis. To fulfil our objective, a complete workflow from raw data processing to analysis and modelling was carried out with data provided by the UNGSC.

## 1.II Related Work

The recent advances in the IoT paradigm applied to energy consumption prediction (IoT-ECP) have been heavily influenced by the increasing developments in ML algorithms. Cheng, Lim, and Hui (2022) provide an insightful account of the most notable recent research done on IoT-ECP. The authors conclude that many of these improvements have concentrated on enhancing algorithmic accuracy, while an emphasis on understanding and predicting consumption patterns is lacking in the literature (Cheng et al., 2022).

Some methodologies focus on the optimal utilisation of data structures coming from IoT environments. Souza et al. (2016) deploy several statistical and ML methods, including DL and decision-tree-based models, to contrast the use of aggregated and disaggre-

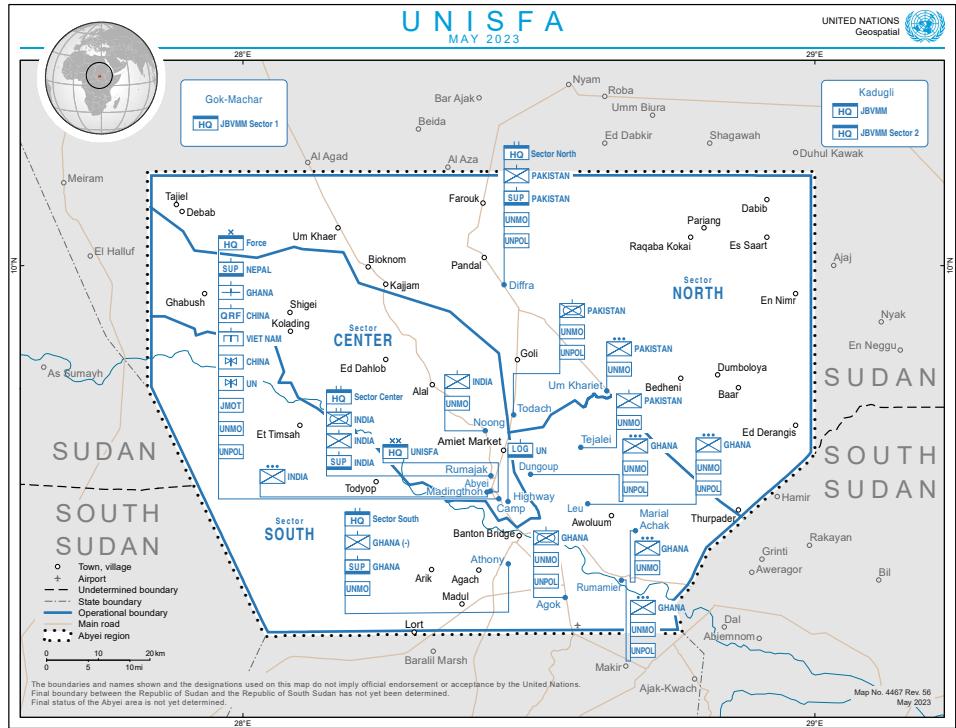


Figure 1.1: UNISFA mission map on May 2023 (UN Geospatial Information Section, 2023).

gated databases. The researchers argue that the inefficient aggregation of granular samples from IoT-surveyed devices with different power demand rates raises the uncertainty of the model and the RMSE (Root Mean Square Error). Accordingly, the authors showed that using the disaggregated approach could reduce the forecast error by up to 80% (Souza et al., 2016).

Another consideration that has to be done is the dataset size. Rabie, Ali, Saleh, and Ali (2020) introduce a load prediction technique that outperforms recent methods for load prediction with large datasets. To achieve this, the authors took advantage of the reduction power of feature selection and outlier rejection methods and then applied a multi-ensemble technique that gives fast and precise load forecasts. As a further development, the authors present the use of fog computing to optimise processing time (Rabie et al., 2020).

Moreover, in the context of real-time data analysis, it is essential to develop methods that easily adapt to new incoming data. The Autoregressive Integrated Moving Average (ARIMA) algorithm is one of the conventional methods to solve the power prediction problem (Cheng et al., 2022). As an example of its application, Alberg and Last (2018) show an

improvement in ARIMA that updates the predictive model using a sliding window technique. This enhancement allows for the disregard of older historical data to prioritise the most recent sensor readings and constitutes a suitable solution for the IoT-ECP in this context.

These authors thus stress the usefulness of adjusting state-of-the-art techniques. In this sense, R. J. Hyndman and Athanasopoulos (2018) showcase the adaptability of ARIMA models through the Hyndman-Khandakar algorithm developed in R. J. Hyndman and Khandakar (2008). This procedure, uses different tests and optimization techniques to automate the selection of ARIMA models for univariate time series modelling. This comes at hand in the context of parallel data sequences arising in the IoT context.

An important insight that ML methods can drive is the identification of anomalous consumption. Fong et al. (2018) provide an example of this type of advance. In their work, the authors present the application of a Very Fast Decision Tree algorithm that uses a pre-processing technique (Misclassified Recall) to filter noisy data and forecast abnormal consumption. We stress the relevance of this type of approach that deals with undesirable data quality and simultaneously attends to the anomaly detection.

Another example of outliers detection through DL techniques is presented by Li, Logenthiran, Phan, and Woo (2019). In this case, the methodology developed by the authors is focused on energy theft in a power grid. Furthermore, the authors give an insightful comparison of different DL methods to achieve their objectives.

As commented by Cheng et al. (2022), there is a general lack of more comprehensive approaches. Following the literature review carried out by the authors, we also identified a shortage of explanations of spatial relationships between the sensors that might account for spatially structured or unstructured effects. In particular, most of the methods currently use ML algorithms with no emphasis on any statistical spatio-temporal analysis. The current project seeks to fill this void, with the core goal of incorporating the spatial relationships of IoT meters to improve comprehension and prediction of an IoT-ECP model.

A novel approach leveraging both ML techniques and Spatial Statistics was applied by Niraula, Mateu, and Chaudhuri (2022) to model Covid in areal count data. Particularly, the predictions of LSTM models per region were used as the expected values of a Poisson regression using Bayesian inference, to further fit the residues and take into account spatial features, such as the regions' neighbourhood structure and mobility patterns among them. Though different in the underlying physical nature of the problems, the lattice structure

of the data in both epidemiology and IoT-ECP settings suggests a possible adoption of the aforementioned techniques to bridge ML and Spatial Statistics.

### **1.III Aim and Research Questions**

Previous outstanding efforts have been made in research to model and predict power consumption in a surveyed environment within the IoT paradigm. However, as discussed in Section 1.II, less attention has been dedicated to the application of statistical modelling of sensors to account for spatial relationships that might influence power consumption.

As an improvement to this identified gap in literature, this project leveraged Spatial Statistics to account for these relationships as a complementary modelling feature. Linked to these objectives, the research was carried out by taking as a guide the following research questions:

- Q1. To what extent can ML and Spatial Statistics enhance model robustness against noisy data?
- Q2. How could the incorporation of spatial relationships among sensors into a model increase its comprehension and accuracy?

To address these questions, we developed a layered model that breaks the forecast task into three main sub-models: LSTM, ARIMA, and BYM. The design and implementation of these model components is described in Chapter 4. The complete data selection and analysis procedures addressed Q1, whereas the inclusion of spatial features in the LSTM and statistical models covered Q2.

### **1.IV Outline**

This work is divided into five chapters. Chapter 2 describes the data at hand, its structure, the process followed to successfully select the scope of the research, and the data cleaning and transformation performed. Then, Chapter 3 offers a synthetic exposure of the theory behind the models used in this project.

Chapter 4 showcases the application of the methodology to the posed problem. We present and discuss the results of the models deployed within this research in Chapter 5. Then, the final remarks and conclusions are addressed in Chapter 6.

# Chapter 2

## Problem and Data Description

In this chapter we describe the data provided by the UNGSC, the challenges encountered in the selection of the project scope and the processing followed in preparation for the main analysis. A summary of the workflow is depicted in Figure 2.5.

### 2.1 Data Structure

Given the large size of the provided data, we used a local database to store it. The data includes sensor readings of water and energy supply, consumption, or generation in the UNISFA mission between two years (from September 23rd 2022 to September 23rd 2022). A query example from the local database is shown in Figure 2.1.

The database contains 29 fields that include numeric readings, sensor information, and metadata (e.g. manual descriptions and custom sensor names). It provides two fundamental data dimensions: time and individual sensor. More precisely, the latter can be identified by a unique sensor ID.

In Table 2.1, we summarize the most relevant dataset attributes for our project, in terms of meaningfulness and completeness. Values are provided for categorical fields with a small amount of classes. The descriptions are based on a dictionary provided by the UNGSC, as well as on inspection and exploration of the database.

```

1 --COPY(
2  SELECT "MISSION_DEVICE_TAG",
3         "TAG_WEIGHT",
4         "TAG_VALUE_TIMESTAMP",
5         "TAG_VALUE_RAW",
6         "TAG_VALUE_DCU_PI_INFO",
7         "TAG_VALUE_DATE",
8         "TAG_VALUE_TIME"
9      FROM raw_data
10     WHERE "MISSION_NAME" = 'UNISFA'
11       AND "OGI_LEVEL" = '2'
12       AND "OGI_MEASURE_TYPE" = 'Con'
13       AND "TAG_DISPLAY_UNIT" = 'kWh'
14   ORDER BY "TAG_VALUE_TIMESTAMP" ASC
15 LIMIT 100
16 --)
17 --TO 'C:/Program Files/PostgreSQL/my_SQL_data/Master thesis SQL WITH CSV HEADER'
18 --WITH CSV HEADER

```

Total rows: 100 of 100 | Query complete 00:00:39.540 | Ln 3, Col 24

Figure 2.1: Example of an SQL query from the local database. Power consumption data for the whole UNISFA mission was retrieved ensuring sequential time ordering using the timestamp field. Here the query is limited to 100 entries to simplify the example.

Table 2.1: Raw Data Attributes.

Field name	Description	Values
DEVICE_LOCATION_NAME	Name of camp inside the mission	*
OGI_LEVEL	Sensor level assigning a hierarchy between sensors	1,2,3
MISSION_DEVICE_TAG	Sensor unique ID	*
OGI_LAT	Latitude of sensor	*
OGI_LONG	Longitude of sensor	*
TAG_VALUE_TIMESTAMP	Timestamp of reading	*
TAG_WEIGHT	Bump-up value to convert and report display units	$10^{-3}$ , 1, $10^3$
TAG_VALUE_RAW	Cumulative numeric reading	*
OGI_MEASURE_TYPE	Measurement type	Production, Consumption, Supply
TAG_DISPLAY_UNIT	Unit of measurement	[kWh], [L], [ $m^3$ ], [mm]

## 2.II Project Scope and Data Selection

As it is shown Table 2.1, the provided dataset includes data on two main resources: energy and water. We filtered down only energy consumption data in kWh to address our energy consumption prediction objectives. After this, the key step towards data selection was to detect a camp with complete, abundant, and well distributed sensor data. We now review in detail these aspects.

Regarding time distribution, the key criteria to filter out sensors was completeness of the corresponding time series. Only sensors reporting at least 80% of the readings were kept. To this end, the number of readings per sensor was used to extract a degree of completeness per sensor. However, special care needed to be taken in this step, as different sensors registered consumption at different pace (every 15 minutes or every 30 minutes, in the case of this project).

As for spatial distribution, we noticed that multiple sensors shared location. We identified and discussed this issue with the matter experts in the UNGSC. Two main reasons for this issue were spotted. First, level-1 sensors (registered in the OGI\_LEVEL attribute) capture data incoming from low-level sensors (levels 2 and 3). Thus, they are geolocated at a general location that represents a whole section of a camp or a camp itself. Second, some sensors appear to be geolocated at the camp level rather than an individual room or service location inside the camp possibly due to misuse by operators. Therefore, with this in mind, we kept only sensors of the intermediate level 2 and filtered out sensors with repeated location.

After performing initial data exploration, we found a total number of 345 level-2 sensors at the headquarters camp located in the city of Abyei. After applying the space and time quality filters commented in the previous paragraphs, 220 remained suitable for our objectives. This sufficiently large number of sensors ensured us an adequate setting for the project.

In particular, we highlight that 62 of the sensors were discarded due to their clustered location. Though not identified as sensors with a common location, these devices were not included due to the possible spatial noise that they could add to the incoming spatial modelling. To fulfil this task, a buffer threshold of 4 m was set as a criterion to classify them as isolated or clustered. The selection of this buffer distance was possible by performing the overlay of sensor locations in this camp-level combination with OpenStreetMap (OSM) data depicting room locations (Figure 2.2) and manually extracting the distance between

rooms for some chosen pairs of buildings.

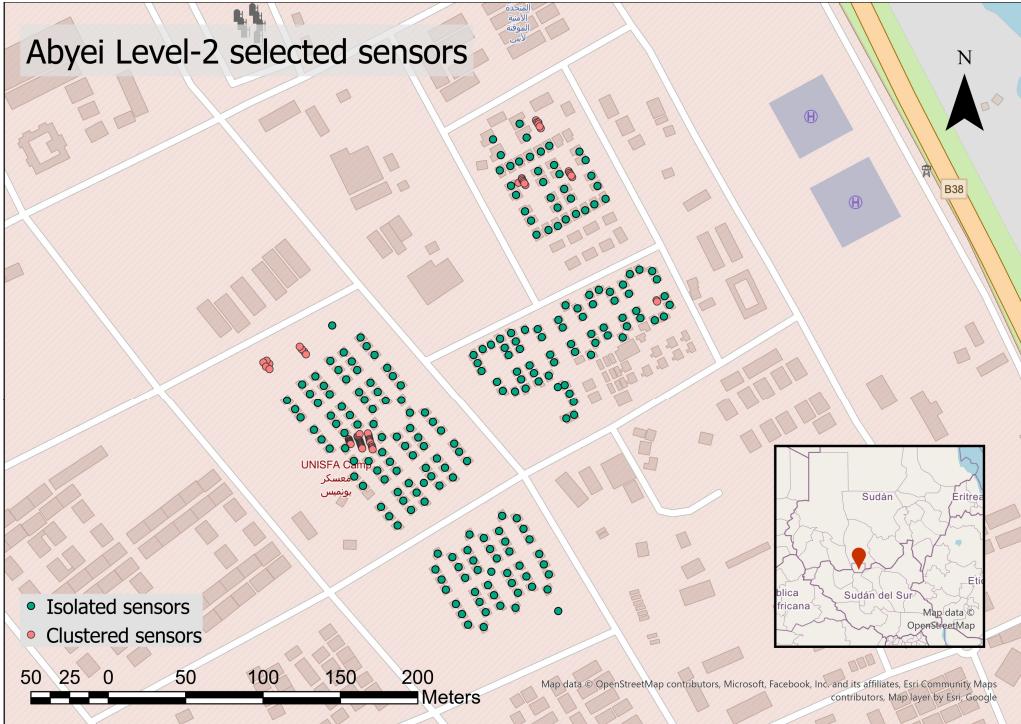


Figure 2.2: Spatial distribution of power consumption sensors at the Abyei camp, with selection of isolated sensors inside a buffer distance of 4 m.

It was central to understand that the reported measurements are cumulative readings, meaning that at the sensor level time series should show a monotonic nature. Also, it is worth mentioning that after the data selection, all the weight values were identically 1. Therefore, the field was further disregarded and no unit conversion was executed. Thus, we left all power consumption units as [kWh] throughout the project.

## 2.III Data Processing

As a fundamental step towards modelling, the setup of ready-for-analysis data involves several stages that need to be fulfilled in a careful and efficient way. To this end, the data processing developed in this project followed the workflow on (Zeng, Ho, & Yu, 2020), given that the authors clearly outline the process they performed and apply it to a similar general objective (prediction of building power consumption through Gaussian Process Regression). Likewise, we also took into consideration the main data transformation strategies discussed in (Browle, 2020, Chapters 5 and 8). The processing was applied systematically at the sensor level, that is, per univariate time series.

The first important transformation performed involved the time re-indexing of each sequence to guarantee complete and regular time-wise distribution of data slots. Due to timestamp attribute noise in the provided dataset, these values were shifted up to 5 minutes to match regular patterns meeting sharp hour timestamps, where needed, to account for consistency across sensors. We stress that this step did not corrupt the performance of our models and their correct interpretation, as they run sequentially regardless of actual time values and were set at a higher time aggregation level (daily).

The next step consisted on dropping incorrect readings per sensor. To achieve this, the matter experts from the UNGSC ensured us the two basic rules the data should have followed:

R1. All energy consumption recordings should be non-negative.

R2. As data is gathered cumulatively, all time series should be non-decreasing as functions over time.

Therefore, negative and decreasing values were dropped and masked as 'NA'. Up to this point, missing values encompass either empty data slots and incoming incorrect values.<sup>1</sup>

Next, as we were interested in the actual consumption value per time slot rather than the cumulative quantities, we performed the differencing of each time series, by applying the transformation

$$y'_t = y_t - y_{t-1}, \quad t = 2, \dots, T. \quad (2.1)$$

for a time series of length  $T$  and cumulative energy consumption values  $y_t$ . It is worth noting that in this setting, condition R2 was fulfilled by checking the non-decreasing behaviour of data at a local level, that is, from immediately previous values rather than from the time series as a whole.

However, our data exploration also yielded incoming anomalously increasing values that implied positive outlier values for the computed  $y'_t$  differences. To clean these extreme values, only quantities below a set threshold were kept, according to the rule:

$$y'_t \leq \begin{cases} \tau, & \tau > 0, \\ u, & \tau = 0, \end{cases} \quad (2.2)$$

$$\tau = \min \{ \text{Med} + k(P_p - \text{Med}), u \},$$

---

<sup>1</sup>According to the matter experts of the UNGSC, both issues could come from either failure of power supply to a sensor, network issues, or individual sensor malfunction.

where  $\text{Med}$  is the median of the corresponding time series  $\{y_t\}$ ,  $u$  is a customizable overall upper bound set by the user, and the factor  $k$  allows for a scaled deviation from the percentile-median distance  $P_p - \text{Med}$ .

Table 2.2: Outlier threshold parameters.

Parameter	Description	Value
$p$	Percentile	99
$k$	Deviation scaling factor	3
$u$	Overall maximum value per time series	10 [kWh]

Definition 2.2 adapts the median deviation described in (Dekking, Kraaikamp, Lopuhaä, & Meester, 2005, §16.2) to a particular percentile  $P_p$ . Table 2.2 presents the actual parameters applied. The choice of the percentile was done after trying out different values.<sup>2</sup> We highlight that the permissiveness of the final selection  $p = 99$  comes from the large amount of values per time series<sup>3</sup> and the need to minimize data rejection. Some examples of rejected large difference values after criterion (2.2) are captured in Table 2.3.

Table 2.3: Maximum outlier difference values per sensor (largest 10).

Sensor ID	Outlier value [kWh]
1-1750-7601	$4.65 \times 10^8$
1-126-345	$7.83 \times 10^3$
1-47-586	$4.35 \times 10^3$
1-184-254	$2.88 \times 10^3$
1-1726-6996	$2.74 \times 10^3$
1-1758-7591	979
1-2118-9452	449
1-1570-6946	368
1-2123-9457	363
1-1573-6949	361

Moreover, the election of the percentile definition in (2.2) over the more standard  $me-$

---

<sup>2</sup> $p = 90, 95, 97, 99$ .

<sup>3</sup>More than 70,000 in the case of sensors capturing data every 15 minutes.

dian absolute deviation (MAD) approach in (Dekking et al., 2005, §16.2) comes from the fact that for many of the studied time series, the dominant amount of zero values yielded null median and MAD values. This renders the latter procedure unfit for the definition of a positive threshold. To illustrate this issue, we present an example of power consumption values across a time series in Figure 2.3. In this sensor, for instance, differences are concentrated around 0 and 1 with zero median.

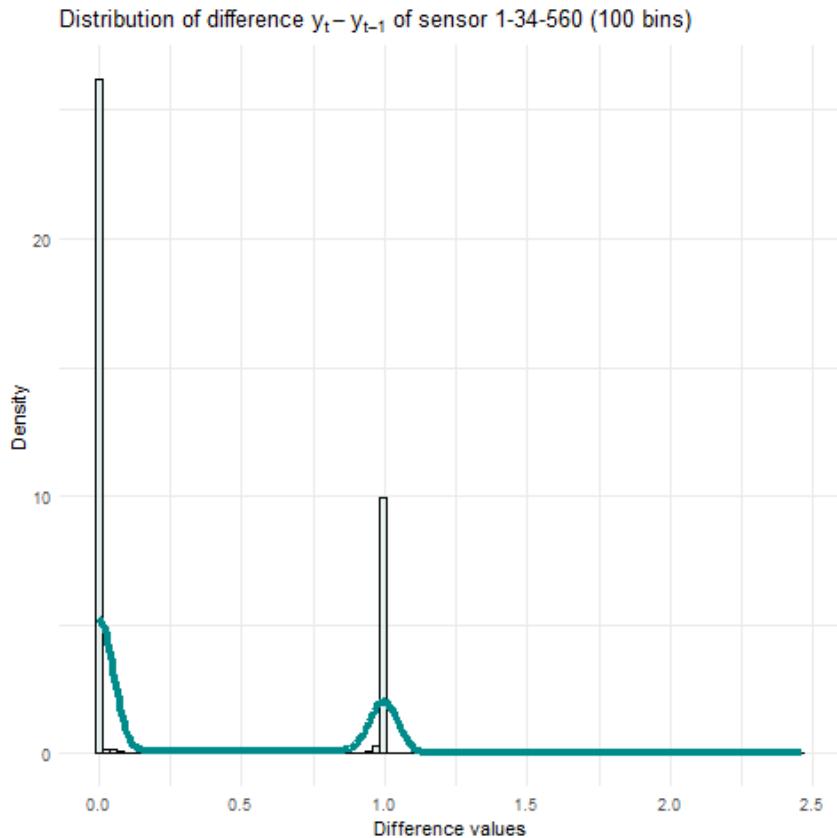


Figure 2.3: Distribution of power consumption differenced values  $y'_t$  of a sensor.

After time irregularity was fixed and the incorrect and anomalous values were discarded, we performed interpolation on individual time series to restore missing values. The *Kalman filter* method was applied to each univariate sequence, as it is a robust possibility to deal with noisy data in the context of power consumption (Khan, Iqbal, Ahmad, & Kim, 2021). The R package `imputeTS` (Moritz & Bartz-Beielstein, 2017) provided us with an automatic and ready-to-use method to apply Kalman smoothing for time series imputation, adapted from the classical methodology developed by Kalman (1960).

The cleaning and transformation procedure exposed in this section entails modifications in the target data for our models. Overall, we claim that the data of the 220 sensors

modelled remains lightly touched apart from a few amount of sensors, in terms of new values imputed.

To understand this, we reviewed the degree of data manipulation per sensor in each of the processing stages. The interpolated new quantities belong to three exclusive categories: missing values in the provided dataset, dropped readings according to rules R1 and R2, and discarded values after the application of the outlier threshold rule (2.2). Table 2.4 shows the sensors with greatest manipulation. In each category, quantities are calculated as a percentage of modified values per device per category. Moreover, the table is ordered by decreasing interpolation degree of manipulation (last column). According to this data, only three time series underwent Kalman interpolation in more than 5% of their data.

Table 2.4: Percentage of data manipulation by sensor and category (10 most processed sensors).

Sensor ID	Missing data	Quality check (R1 and R2) rejection	Outlier rejection	Interpolation
1-163-412	0.01	24.07	0	24.09
1-162-410	0.01	24.07	0	24.09
1-126-345	0.01	8.34	1.71	10.07
1-66-424	0.01	4.53	0	4.55
1-69-430	0.01	4.49	0	4.50
1-68-428	0.01	4.49	0	4.50
1-70-432	0.01	4.49	0	4.50
1-67-426	0.01	4.49	0	4.50
1-92-477	0.01	3.96	0.47	4.45
1-93-479	0.01	3.96	0.32	4.30

To end this section, we show in Figure 2.4 how the time series shape changes after the cleaning and differencing transformations so far described. Likewise, the diagram in Figure 2.5 presents a summary of the described processing workflow of this chapter.

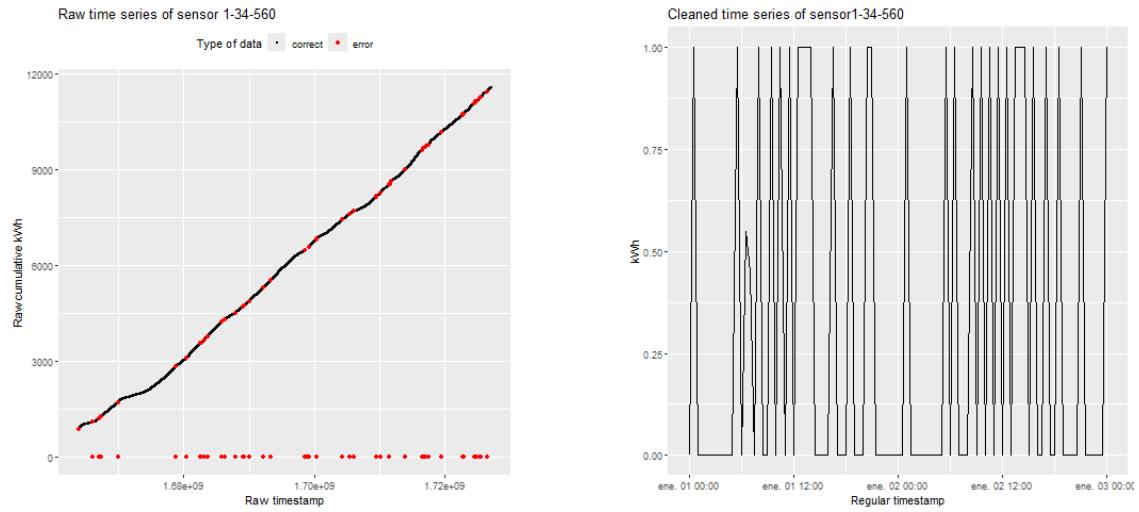


Figure 2.4: Example of data transformation for an individual sensor from cumulative raw values, labelled as correct and incorrect according to the quality check rules (left), to clean time and consumption values (right).

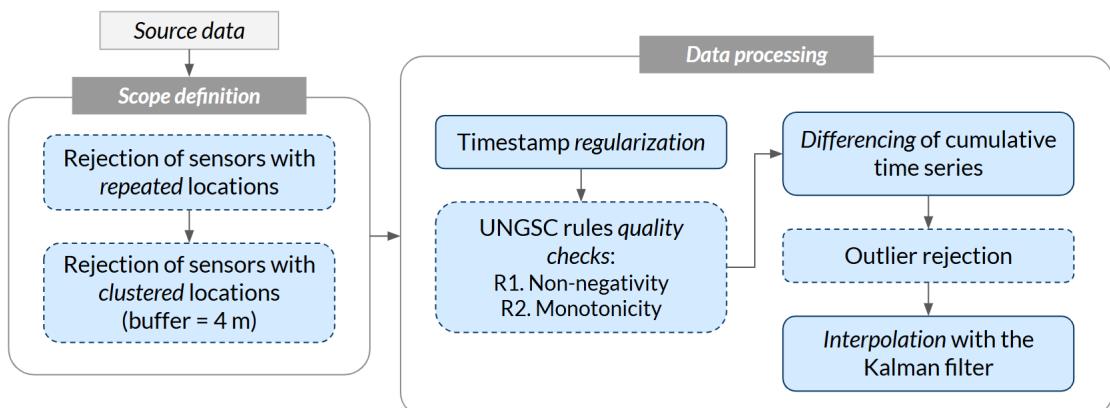


Figure 2.5: Data selection and processing workflow.

## 2.IV Software

All the data exploration, selection, processing, and modelling was performed using R (version 4.3.3). Packages specifications are given in the Table 2.5. Further details on package usages and software settings are explained in the corresponding sections. Figures were also created through R libraries. Customization of maps with contextual OSM data was performed in ArcGIS Pro. The full R source code used for this project is publicly available in the following GitHub repository: <https://github.com/rodrigomalagon/Modelling-of-IoT-surveyed-UN-camps>.

Table 2.5: R packages used for the project.

Package	Version	Main uses	Project stage	Reference
ggplot2	3.5.1	Plotting of graphs and figures	All	(Wickham, 2016)
dplyr	1.1.4	Data manipulation, exploration, and transformation	All	(Wickham et al., 2023)
sf	1.0.19	Spatial data processing and analysis	Exploration, selection, modelling	(E. Pebesma & Bivand, 2023)
imputeTS	3.3	Kalman filter interpolation of missing and dropped data	Processing	(Moritz & Bartz-Beielstein, 2017)
gstat	2.1.2	Varioogram analysis	Modelling	(E. J. Pebesma, 2004)
keras3	1.1.0	LSTM modelling	Modelling	(Kalinowski et al., 2024)
forecast	8.23.0	Automatic ARIMA modelling	Modelling	(R. Hyndman et al., 2024)
spdep	1.3.8	Neighbourhood analysis for BYM model	Modelling	(R. Bivand & Wong, 2018)
INLA	23.9.9	BYM modelling	Modelling	(Rue et al., 2009)

# Chapter 3

## Methods

In this chapter, we describe the main methods used in the modelling stage of the project. Three types of models were implemented: LSTM, ARIMA, and BYM models. These modelling components were assembled in a hierarchical setting, further described in Chapter 4.

### 3.1 Deep Learning Models

Due to its power to model non-linear problems, the DL framework has increasingly been adapted to the study of spatio-temporal phenomena. Recurrent Neural Networks (RNNs) constitute an outstanding type of DL method within this setting, given its flexibility to model diverse sequential patterns of dependence (Wikle, Zammit-Mangion, & Cressie, 2019).

In the classical work by Hochreiter and Schmidhuber (1997), the authors presented the LSTM network, a particular RNN architecture. In the paper, they thoroughly explain its advantages and limitations in comparison with other traditional methods. In particular, with this network design the authors succeeded in improving the bridging of time lags and counteracting the decaying effects of basic RNNs.

The application of these RNN architectures covers a broad spectrum of research fields, including temporal land cover and land use classification (Rußwurm & Körner, 2017), forecast in hydrology (Fang & Shen, 2020; Ziheng Sun & Fang, 2019), or epidemiology (Niraula et al., 2022). Furthermore, in the IoT context, Li et al. (2019) developed an energy theft detection system for Smart Homes based on LSTM modelling.

### 3.I.i Recurrent Neural Networks

An RNN is a particular type of Artificial Neural Network (ANN) that specialises in the processing of sequential data. In this setting, however, in each hidden layer the avoidance of cycles in the network design is relaxed, as depicted in the circuit diagram (left) of Figure 3.1. This feature guarantees that time series can be recursively fed to the model, so that it leverages the temporal context while learning (Körner & Rußwurm, 2021).

Moreover, this recursive paradigm allows the processing of samples of variable length. This flexible feature is not usually adopted by more traditional methods that define and fix *a priori* the possible length of the time series to be read. Already in its foundational paper, Hochreiter and Schmidhuber (1997) express these fundamental differences. Moreover, they stress that LSTM models can handle noisy data in a better way.

However, as illustrated in Figure 3.1, these architectures can be thought of as feed-forward ANNs (through time) after an *unfolding* procedure, where the temporal component of the data is used to represent the network design as a sequence of concatenated cells (Goodfellow et al., 2016).

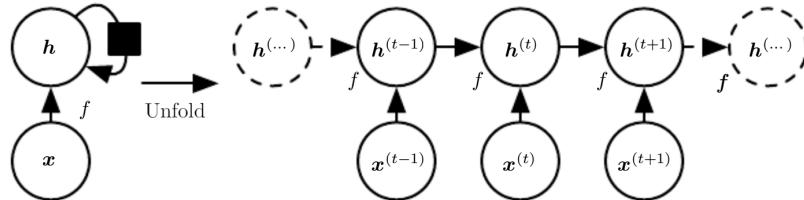


Figure 3.1: Unfolding of an RNN hidden layer. The black square refers to a delayed passing of the hidden state  $h$  on each time step (Goodfellow et al., 2016).

Formally, the recursive paradigm is classically modelled by the update equation:

$$\mathbf{h}_t = \zeta(\mathbf{W}_{in}\mathbf{x}_t + \mathbf{W}_{rec}\mathbf{h}_{t-1} + \mathbf{b}_{rec}), \quad 0 \leq t \leq T, \quad (3.1)$$

where

1.  $\zeta$  is an activation non-linear function that encodes the *inner state*  $s_t = \mathbf{W}_{in}\mathbf{x}_t + \mathbf{W}_{rec}\mathbf{h}_{t-1} + \mathbf{b}_{rec}$ ,
2.  $\mathbf{W}_{in} \in \mathbb{R}^{n \times d}$  is a matrix of weights that includes the  $t$ -th input vector  $\mathbf{x}_t$  of size  $d$  from the time series,

3.  $\mathbf{W}_{rec} \in \mathbb{R}^{n \times n}$  is a matrix of weights that passes onward in time the previous hidden state  $\mathbf{h}_{t-1}$ ,
4.  $\mathbf{b}_{rec} \in \mathbb{R}^n$  is a bias vector that corrects the bias of the inner state.

With this scaffolding, RNNs can model temporal phenomena by means of predictive outputs  $\mathbf{o}_t$  inferred from the hidden states  $\mathbf{h}_t$ . Following Goodfellow et al. (2016) and adopting notation by Körner and Rußwurm (2021), this can be done, on the last hidden layer, by a linear data model of the form

$$\mathbf{o}_t = \mathbf{W}_{out} \mathbf{h}_t + \mathbf{b}_{out}, \quad 0 \leq t \leq T, \quad (3.2)$$

where  $\mathbf{W}_{out} \in \mathbb{R}^{k \times n}$  is a matrix mapping  $n$ -dimensional hidden states into the  $k$ -dimensional space of desired outputs<sup>1</sup> and  $\mathbf{b}_{out} \in \mathbb{R}^k$  the corresponding bias vector.

As Goodfellow et al. (2016) comment, compared to other DL architectures, the two major benefits from this recurrent setting are the invariance of the size of the model – independent of the size of the time series – and the reduction of its complexity to a single transition procedure from state to state, as in equation (3.1).

As RNNs are computationally equivalent –inside a layer– to a feed-forward-through-time network, common learning mechanisms from the general context of ANNs are applicable within this framework. Therefore, learning can be performed by minimising a loss function by *back-propagation through time* (Körner & Rußwurm, 2021).

To this end, loss is defined as a combination of error across the unfolded layer, that is:

$$\mathcal{L} = \sum_{t=0}^T \mathcal{L}_t, \quad (3.3)$$

where the individual loss at a given time step  $t$  captures the level of predictive error at the corresponding cell by comparing the output  $\mathbf{o}_t$  with the target  $\mathbf{y}_t$ .

As an example, in a classification task on  $k$  classes, this can be done by means of applying a function that normalises the  $k$  vector coordinates of  $\mathbf{o}_t$  and derives by-class probabilities:

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{o}_t), \quad (3.4)$$

with  $i$ -th entries

$$(\text{softmax}(\mathbf{o}_t))_i = \frac{e^{(\mathbf{o}_t)_i}}{\sum_{j=1}^k e^{(\mathbf{o}_t)_j}}, \quad 1 \leq i \leq k. \quad (3.5)$$

---

<sup>1</sup>For instance,  $k$  different classes in the case of a classification tasks.

The cross entropy loss would then evaluate the negative log-likelihood of  $y_t$  given the inputs  $\{x_0, \dots, x_t\}$ , that is,

$$\begin{aligned}\mathcal{L}_t(\mathbf{y}_t | \mathbf{x}_0, \dots, \mathbf{x}_t) &= -\log(\hat{\mathbf{y}}_t)_{i_t} \\ &= -\log(\text{softmax}(\mathbf{W}_{out} \mathbf{h}_t + \mathbf{b}_{out}))_{i_t}, \quad 0 \leq t \leq T,\end{aligned}\quad (3.6)$$

where  $1 \leq i_t \leq k$  is the (unique) correct class identifier at time  $t$  and  $(\hat{\mathbf{y}}_t)_{i_t}$  the corresponding vector entry of  $\hat{\mathbf{y}}_t$ . Figure 3.2 illustrates this process in an image processing setting.

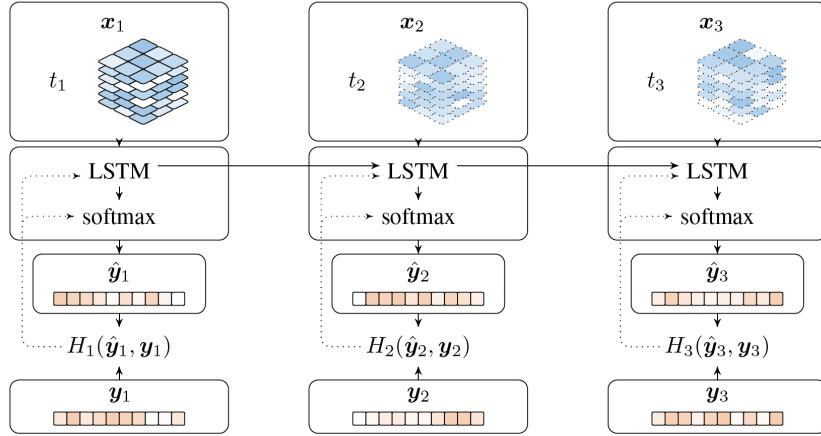


Figure 3.2: Calculation of the cross-entropy loss (here denoted by  $H$ ) in a land cover classification task, using input spectral data  $x_t$  of dimension  $d = n_b \times p^2$ , that is,  $n_b$  spectral bands of raster images with size  $p \times p$  (Rußwurm & Körner, 2017).

As in the feed-forward setting of ANNs, from equation (3.3) and in light of the recursive application of the composition of functions in (3.1), we can deduce a handful way to obtain the gradient of the loss function  $\mathcal{L}$ . It can be conveniently factorised across time to apply back-propagation and compute it. For instance, to update the recurrent weights of  $\mathbf{W}_{rec}$ , we verify by linearity and the iterative application of the chain rule across time:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{rec}} &= \sum_{t=0}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_{rec}} = \sum_{t=0}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \cdot \frac{\partial \mathbf{h}_t}{\partial \mathbf{W}_{rec}} \\ &= \sum_{t=0}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \cdot \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \cdots \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} \cdot \frac{\partial \mathbf{h}_1}{\partial \mathbf{W}_{rec}}\end{aligned}\quad (3.7)$$

which entails an *exploding/vanishing gradient* problem if we take a closer look at the intermediate products in (3.7). Explicitly, adapting the matrix arguments by Hochreiter and Schmidhuber (1997, §3.1.5) we deduce from from equation (3.1) and the chain rule that

$$\frac{\partial \mathbf{h}_{t-\tau}}{\partial \mathbf{h}_{t-\tau-1}} = \zeta'(\mathbf{s}_{t-\tau}) \cdot \mathbf{W}_{rec} = \zeta'(\mathbf{s}_{t-\tau}) \cdot Q \Lambda Q^{-1}, \quad 0 \leq \tau \leq t-2, \quad (3.8)$$

where  $\mathbf{W}_{rec} = Q\Lambda Q^{-1}$  is the eigenvector decomposition of  $\mathbf{W}_{rec}$ ,  $Q \in \mathbb{R}^{n \times n}$  is its unitary matrix of eigenvectors, and  $\Lambda \in \mathbb{R}^{n \times n}$  the diagonal matrix of its eigenvalues. Indeed, for instance when  $\zeta$  is the multidimensional version of the ReLU function defined element-wise by

$$\zeta_i(\mathbf{s}) = \begin{cases} s_i, & s_i \geq 0, \\ 0, & s_i < 0, \end{cases} \quad (3.9)$$

then  $\zeta'(\mathbf{s}_t)$  consists of a diagonal matrix with either 1's or 0's according to the sign of the corresponding elements of the vector  $\mathbf{s}_t$ . Thus, in this case the general behaviour of (3.8) depends on the  $(t - 1)$ -th powers of the eigenvalues  $\lambda_i$  in  $\Lambda$ , which are large if  $|\lambda_i| > 1$  or heavily damped if  $|\lambda_i| < 1$ , as  $t$  increases.

Among the possible existing solutions to this issue, LSTM poses a countermeasure. Its gated structure, explained in the next subsection, guarantees that the flow of information through time is limited by the implementation of regulating gates opening or closing this uncontrolled propagation (Körner & Rußwurm, 2021).

### 3.I.ii Long Short-Term Memory Networks

LSTM architectures are a particular type of RNNs in which the received information by a cell is fragmented into *memory units* or *gates* to gain control of the flow of information over time (Körner & Rußwurm, 2021). The immediate outputs vectors of these components interact internally to regulate the final set of states of the cell. In this case, rather than a single hidden state, the  $t$ -th cell is designed to pass forward two states: the cell state  $c_t$  and the hidden state  $h_t$ .

Formally, in its simplest definition, this is achieved by the following set of combinations of linear and non-linear operations:

$$\begin{cases} c_t = f_t \odot c_{t-1} + i_t \odot m_t, \\ h_t = o_t \odot \tanh(c_t), \end{cases} \quad (3.10)$$

where  $\odot$  stands for the Hadamard (element-wise) product of vectors,  $f_t$  is the *forget gate* vector,  $i_t$  is the *input gate* vector,  $m_t$  the *modulation gate* vector, and  $o_t$  is the *output gate* vector, all of a common dimension  $n > 0$  (Körner & Rußwurm, 2021).

From this set of equations, we interpret two key facts, before further inspection of the definition of the gates. First, the cell state  $c$  depends recursively on its previous steps, regulated by the forget gate vectors  $f$ , in combination with the inclusion of a current input

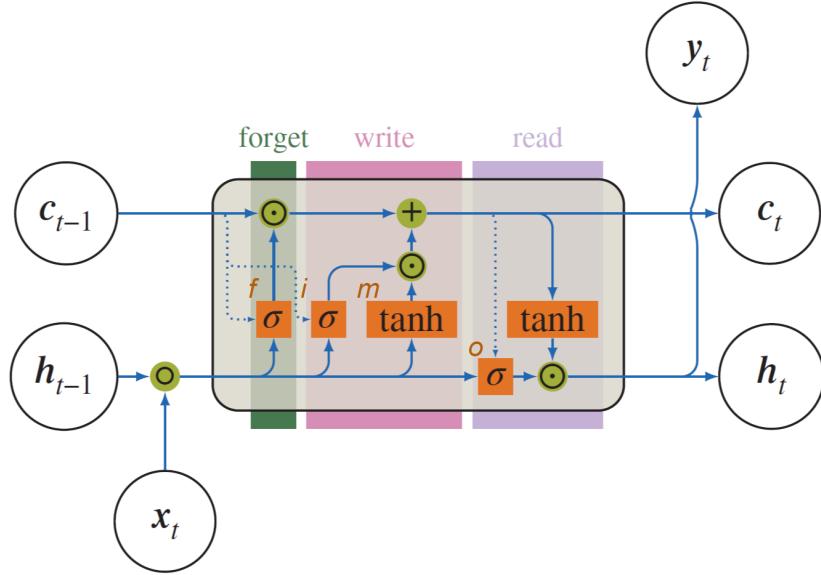


Figure 3.3: Inner structure of an LSTM cell. Here  $f$  represents the forget gate,  $i$  the input gate,  $m$  the modulation gate, and  $o$  the output gate. Figure adapted from (Körner & Rußwurm, 2021).

$i$  weighted by the modulation gate vector  $m$ . This helps to control the forward flow of state information as smaller values in the forget gate vector  $f_t$  would directly hinder the influence of the previous state  $c_{t-1}$  on the current state  $c_t$ .

Second, the hidden state depends on a derived output vector  $o$  weighted by a non-linear activation of the current state  $c$ . These inner interactions can be appreciated in Figure 3.3, where the input and modulation gates make up the *write* substructure.

We now present the definition of each gate in (3.10). Following the update paradigm of equation (3.1), the forget gate consists of the weighted combination of the previous hidden state  $h_{t-1}$  and the current input  $x_t$ —along with a corresponding bias  $b_f$ —followed by the application of an activation function:

$$f_t = \sigma \left( \mathbf{W}_f \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + \mathbf{b}_f \right), \quad (3.11)$$

where  $\mathbf{W}_f \in \mathbb{R}^{n \times (n+d)}$  is the matrix of weights,  $\mathbf{b}_f \in \mathbb{R}^n$ , and  $\sigma$  is the element-wise application of the *logistic sigmoid function* defined by

$$\sigma(x) = (1 + \exp(-x))^{-1}, \quad (3.12)$$

in the one-dimensional case—with range equal to the open unit interval  $(0, 1)$ . Similar

definitions follow the rest of the gates with corresponding weight matrices:

$$\begin{aligned} \mathbf{i}_t &= \sigma \left( \mathbf{W}_i \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_i \right), \\ \mathbf{m}_t &= \tanh \left( \mathbf{W}_m \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_m \right), \\ \mathbf{o}_t &= \sigma \left( \mathbf{W}_o \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_o \right), \end{aligned} \quad (3.13)$$

where the modulation gate  $\mathbf{m}_t$  is activated by the hyperbolic tangent, with range equal to the interval  $(-1, 1)$  (Körner & Rußwurm, 2021).

### 3.I.iii Optimization and Implementation

As stated in Section 3.I.i, standard parameter optimization techniques from ANNs apply to RNNs. The core procedure thus consists of an iterative update of the values of weight matrices  $\mathbf{W}$ . To achieve this, a slight displacement of these quantities is performed across the parameter space in the opposite direction of the gradient of the loss function  $\mathcal{L}$ . Although there are different methodologies to apply this process, they rely on updates of the form:

$$\mathbf{W}' = \mathbf{W} - \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{W}}, \quad (3.14)$$

where  $\alpha$  is a step parameter that controls the *learning rate*. This method is called the *gradient descent* optimization.

More complex versions of gradient descent have been implemented. The AdaGrad algorithm by Duchi, Hazan, and Singer (2011) updates the learning rate based on the local geometry of the parameter space to enhance learning in gently sloped regions. In addition, the RMSProp algorithm proposed in (Hinton, 2012) upgrades AdaGrad to overcome convex regions that trap gradient descent when applying AdaGrad. Furthermore, The Adam optimizer designed by Kingma and Ba (2014) includes a momentum component to the velocity update and corrects biases from RMSProp. This method stands out by its robustness against adjustment of hyperparameters. (Goodfellow et al., 2016, §8.5).

As these techniques involve the computation of the gradient of  $\mathcal{L}$  for incoming data values, efficient procedures to derive it have been developed. The central approach is the *back propagation* method, that leverages the network's feed-forward structure and follows chain-rule simplifications as in (3.7) (Chollet, Kalinowski, & Allaire, 2022, §2.4.4).

In the DL framework, training can be performed in batches of samples, rather than on individual data reads, to speed up and stabilize the weight update procedure. For this purpose, the critical parameter to be adjusted is the batch size. Moreover, iterative batch runs of the complete training dataset, called *epochs*, are set to avoid underfitting or overfitting. Setting different combinations of the learning rate, the batch size, and the number of epochs renders different performances and computing times (Goodfellow et al., 2016, §8.1.3).

Keras is an API built on top of the TensorFlow ML platform that provides implementation facilities of DL tools. Though it is written in Python, an R interface to Keras is available under the `keras3` package (Kalinowski et al., 2024). It provides the capabilities to deploy convolutional-based and recurrent ANN models.

## 3.II Regressive Temporal Models

One of the most extended methods used in forecast tasks is the ARIMA algorithm, which specializes in modelling the time-wise data autocorrelation (R. J. Hyndman & Athanasopoulos, 2018). These models incorporate three main subtypes of model that make up the ARIMA nomenclature: the autoregressive, the moving average and the integration models. We review them separately.

### 3.II.i AR and MA Models

An *autoregressive model* of order  $p > 0$ , denoted by  $\text{AR}(p)$ , is a linear regression model whose predictor variables are the *lagged* values of the independent variable  $y$  and is defined by:

$$\begin{cases} y_t = \alpha + \left( \sum_{j=1}^p \phi_j y_{t-j} \right) + \epsilon_t, \\ \alpha = \mu \left( 1 - \sum_{j=1}^p \phi_j \right), \end{cases} \quad (3.15)$$

where the inferred parameters  $\phi_1, \dots, \phi_p$  ( $\phi_p \neq 0$ ) measure the influence of recent variable measurements into current readings, the error  $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$  is assumed as a white Gaussian noise and  $y_t$  as stationary<sup>2</sup> with mean  $\mu$  (Shumway & Stoffer, 2017, Definition 3.1).

---

<sup>2</sup>Strict stationarity states that the probabilistic behaviour of every collection of values  $\{y_{t_1}, \dots, y_{t_k}\}$  is equal to that of the set after some time lag, say  $\{y_{t_1+h}, \dots, y_{t_k+h}\}$ . We highlight that this assumption implies that the mean and variance of  $y_t$  remains constant over time (*weak stationarity*) (Shumway & Stoffer, 2017, §1.4).

On the other side, a *moving average* model of order  $q > 0$ , denoted as  $\text{MA}(q)$ , fits current values to previous forecast errors (R. J. Hyndman & Athanasopoulos, 2018, §8.4), according to:

$$y_t = \mu + \epsilon_t + \sum_{j=1}^q \theta_j \epsilon_{t-j}, \quad (3.16)$$

where  $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$  and  $\theta_1, \dots, \theta_q$  ( $\theta_q \neq 0$ ) are the moving average parameters (Shumway & Stoffer, 2017, Definition 3.3).

### 3.II.ii Causality and Invertibility

Desirable properties, such as the simplification of the model and the guarantee of computational stability, arise when suitable constraints are imposed on the parameters  $\phi$  and  $\theta$ . For instance, when assuming  $|\phi_1| < 1$ , an AR(1) model of a stationary time series  $\{y_t\}$  with mean  $\mu = 0$  takes, after iterated substitutions, the form

$$y_t = \phi_1 y_{t-1} + \epsilon_t = 1 + \phi_1^2 y_{t-2} + \phi_1 \epsilon_{t-1} + \epsilon_t = \dots = \phi_1^k y_{t-k} + \sum_{j=0}^{k-1} \phi_1^j \epsilon_{t-j}, \quad (3.17)$$

so that the mean square convergence of the error series to the current variable holds (Shumway & Stoffer, 2017, Equation 3.6):

$$\lim_{k \rightarrow \infty} \mathbb{E} \left( y_t - \sum_{j=0}^{k-1} \phi_1^j \epsilon_{t-j} \right)^2 = \lim_{k \rightarrow \infty} \phi_1^{2k} \mathbb{E} (y_{t-k}^2) = 0. \quad (3.18)$$

That is, we can express the model as the simpler linear process

$$y_t = \sum_{j=0}^{\infty} \phi_1^j \epsilon_{t-j}. \quad (3.19)$$

We highlight that this corresponds to a  $\text{MA}(\infty)$  representation of the model and implies that it can be expressed just in terms of the past (Shumway & Stoffer, 2017). This important category of models are known as *causal* processes.

Likewise, condition  $|\theta_1| < 1$  on a  $\text{MA}(1)$  model implies an analogue reversed AR( $\infty$ ) model representation for the error terms:

$$\epsilon_t = \sum_{j=0}^{\infty} \theta_1^j y_{t-j}. \quad (3.20)$$

Moving average models with a representation as in (3.20) are called *invertible* processes.

### 3.II.iii ARMA Models

The conjunction of autoregressive and moving average models make up a more complex type of models. An *autoregressive moving average* model of autoregressive order  $p > 0$  and moving average order  $q > 0$ , denoted by  $\text{ARMA}(p, q)$ , is defined by

$$\begin{cases} y_t = \alpha + \left( \sum_{j=1}^p \phi_j y_{t-j} \right) + \left( \sum_{j=1}^q \theta_j \epsilon_{t-j} \right), \\ \alpha = \mu \left( 1 - \sum_{j=1}^p \phi_j \right), \end{cases} \quad (3.21)$$

where  $y_t$  is assumed stationary with mean  $\mu$  and  $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$  (Shumway & Stoffer, 2017, Definition 3.5). These more complete models include both an influence of past events and noise. In this general setting, causality and invertibility restrictions are defined, respectively, by

$$\begin{cases} \phi(z) = 1 - \sum_{j=1}^p \phi_j z^j \neq 0, & z \in \mathbb{C}, \quad |z| \leq 1, \\ \theta(z) = 1 - \sum_{j=1}^q \theta_j z^j \neq 0, & z \in \mathbb{C}, \quad |z| \leq 1. \end{cases} \quad (3.22)$$

These constraints allow for the selection of unique models among different solutions of the same process (Shumway & Stoffer, 2017). Moreover, these conditions avoid instability of parameter choice between flips of  $\theta_j$  and  $\theta_j^{-1}$  values, as already noted by Hamilton (1994, Equation 3.7.17). That is, interpretable, stable, and unique stationary ARMA models can be found when the polynomials (3.22) have no roots in the unit complex disc (Brockwell & Davis, 2016, §3.1.1).

### 3.II.iv ARIMA models

The last component of ARIMA models emerges as a tool to strive for the stationarity condition of ARMA models in (3.21). The backward shift operator  $B$  applied to a given time series  $\{y_t\}$  is defined as the one-step-before lag of the series, that is,  $By_t = y_{t-1}$ . Thereupon, the *differencing* of order  $d > 0$  of a time series is defined as the iterative application of the subtraction of each time series current value by the immediately previous value, a total of  $d$  times. Formally, this is expressed by the operator composition  $(1 - B)^d$ .

An *autoregressive integrated moving average* model of orders  $(p, d, q)$  of a time series  $\{y_t\}$  is an  $\text{ARMA}(p, q)$  model of the  $d$ -times differenced time series  $\{(1 - B)^d y_t\}$ , that is,

$$\begin{cases} y'_t = \alpha + \left( \sum_{j=1}^p \phi_j y'_{t-j} \right) + \left( \sum_{j=1}^q \theta_j \epsilon_{t-j} \right), \\ \alpha = \mu \left( 1 - \sum_{j=1}^p \phi_j \right), \end{cases} \quad (3.23)$$

where  $y'_t = (1 - B)^d y_t$  is stationary,  $\mu$  is the mean of  $y'_t$ , and  $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$  (R. J. Hyndman & Athanasopoulos, 2018). Moreover, it is traditional to include the border cases  $q = 0$ ,  $p = 0$  or  $d = 0$  (and its combinations) into the ARIMA class to refer to simpler models where either the autoregressive, differencing, or moving average components are absent in equation (3.23)

### 3.II.v Optimization and Implementation

A common way to detect suitable values for  $p$ ,  $d$ , and  $q$  consists of the inspection of autocorrelation plots, *i.e.* the plots of the *autocorrelation function* (ACF) and the *partial autocorrelation function* (PACF). These two functions describe the degree of dependence of a variable  $y_t$  on its previous values values  $y_{t-1}, \dots, y_{t-h}$  (Shumway & Stoffer, 2017, §3.3) and are defined by

$$\text{ACF : } \rho(h) = \text{corr}(y_t, y_{t-h}) = \frac{\text{cov}(y_t, y_{t-h})}{\sqrt{\text{var}(y_t)\text{var}(y_{t-h})}}, \quad (3.24)$$

$$\text{PACF : } \phi_{hh} = \begin{cases} \rho(1), & h = 1, \\ \text{corr}(y_t - \hat{y}_t, y_{t-h} - \hat{y}_{t-h}), & h \geq 2, \end{cases} \quad (3.25)$$

for lags  $h = 1, 2, \dots$ , where  $\hat{y}_t$  and  $\hat{y}_{t-h}$  represent, respectively, the regressions of  $y_t$  and  $y_{t-h}$  on the intermediate variables  $\{y_{t-h+1}, \dots, y_{t-1}\}$  (Shumway & Stoffer, 2017, §3.3).

This inspection method remains as an auxiliary approach that works following the thumb-up rules in Table 3.1. Though good results may arise by this graphical method (R. J. Hyndman & Athanasopoulos, 2018, §8.5), more systematic methods have been designed. R. J. Hyndman and Khandakar (2008) developed an automatic algorithm that selects  $p$ ,  $d$ , and  $q$  out of the parameter space. To achieve this, it applies stationarity tests to deduce a suitable differencing order  $d$  and a stepwise search to find  $p$  and  $q$  that minimize an information criterion.

Table 3.1: Behaviour of the ACF and PACF plots for ARMA models (Shumway & Stoffer, 2017, Table 3.1).

	AR( $p$ )	MA( $p$ )	ARMA( $p, q$ )
ACF	Tails off	Cuts off after lag $q$	Tails off
PACF	Cuts off after lag $p$	Tails off	Tails off

According to the authors, in forecasting tasks it is preferable to reduce the degree of

differencing  $d$  to sharpen predictions. To avoid that the model selection is biased towards over-differencing, R. J. Hyndman and Khandakar (2008) iteratively apply the KPSS unit-root test from Kwiatkowski, Phillips, Schmidt, and Shin (1992) to decide on the necessity of further differencing. Explicitly, this test sets a null hypothesis of time series stationarity, which is equivalent to *a priori* assuming no differencing.

Next, when  $d$  is selected, the stepwise algorithm minimizes the Akaike's Information Criterion (AIC). This quantity is defined in the ARIMA context by:

$$\text{AIC} = -2 \log(\mathcal{L}) + 2(p + q + k), \quad (3.26)$$

where  $k = 1$  if  $\mu \neq 0$  in (3.23) and  $\mathcal{L}$  is the maximized likelihood of the corresponding ARMA( $p, q$ ) model (3.23). That is, the procedure tries to find more accurate and less complex ARMA( $p, q$ ) models.

This automatic ARIMA model selection is readily implemented in the `forecast` package (R. Hyndman et al., 2024). Furthermore, default approximation procedures are set in its configuration to speed up the computations (R. J. Hyndman & Athanasopoulos, 2018, §8.7).

### 3.III Spatial Models

Geostatistical methods to study the spatial relationship of variables have been widely developed in the past and even adapted to the more complex spatio-temporal setting. Forecast based on kriging techniques is used to model latent processes assumed as continuous variables over space or space-time (Wikle et al., 2019).

A starting point and a key ingredient of kriging involves the study of the spatial semi-variogram of a random field  $Z$  over a region  $G$ , given by

$$\gamma(s, s') = \frac{1}{2} \text{var}(Z(s) - Z(s')), \quad (3.27)$$

for any pair of punctual locations  $s, s' \in G$ . This function captures the variability of the dissimilarity of  $Z$  across space. A classical assumption for (3.27) is the *intrinsic stationarity* of  $Z$ , under which  $Z$  possesses constant mean  $\mu = E(Z(s))$  and the semivariogram is assumed to only depend on the displacements  $h = s - s'$ , rather than on individual locations (R. S. Bivand, Pebesma, & Gómez-Rubio, 2013). Furthermore, the *isotropy* condition strengthens this quality by assuming that the semivariogram depends only in the distance regardless of direction, that is,

$$\gamma_Z(||h||) = \gamma(s, s'), \quad h = s - s'. \quad (3.28)$$

In this setting, if the random field  $Z$  is sampled at locations  $s_1, \dots, s_m \in G$ , the semivariogram may be estimated by

$$\hat{\gamma}(h) = \frac{1}{2N_{\tilde{h}}} \sum_{\substack{i < j \\ \|s_i - s_j\| \in \tilde{h}}}^{N_{\tilde{h}}} (Z(s_i) - Z(s_j))^2, \quad h > 0 \quad (3.29)$$

where  $\tilde{h} = [h - \delta, h + \delta]$  is a tolerance interval around  $h > 0$  and  $N_{\tilde{h}}$  represents the number of distinct pairs of sample locations with a distance inside  $\tilde{h}$  (R. S. Bivand et al., 2013; Wikle et al., 2019). Thus, fitting this *empirical variogram* could help us understand the spatial structure of the data, in terms of distance-based correlation.

In addition to these approaches, when working with areal or lattice data<sup>3</sup>, topological relationships of spatially-fixed readings give rise to the need of considering more suitable spatial relationships in the modelling process. This include neighbourhood relationships, which entails the study of data on graph spaces (R. S. Bivand et al., 2013, §9.2.1).

### 3.III.i Besag-York-Mollie Models

Among the spatial methods that take into account neighbourhood structure of lattice data, BYM models stand out for its effectiveness in spatial epidemiology (Morris et al., 2019). This method was developed by Besag, York, and Mollié (1991) in the context of image processing, though the authors provide its applicability to diseases mapping, using lattice data of the French mainland departments. Particularly, this method is framed in the Bayesian inference paradigm, described in the next subsection.

The main spatial component that renders BYM effective in this discrete context is the use of an *intrinsic conditional autoregressive* (ICAR) latent model. This kind of processes smooth noisy data of sparse events in lattices by pooling average information from neighbouring locations (Morris et al., 2019). To achieve this, we consider a lattice structure of  $N$  locations, say  $G = \{s_1, \dots, s_N\}$ . The neighbourhood relationships from the lattice is inherited among the location indices  $1, \dots, N$ . That is, we denote by  $i \sim j$  whenever locations  $s_i$  and  $s_j$  are lattice neighbours. The ICAR model for a latent process  $u$  defined on  $G$  is conditionally defined by

$$u_i | u_{-i}, \tau_u \sim \mathcal{N} \left( \frac{1}{n_i} \sum_{j \sim i} u_j, \frac{1}{n_i \tau_u} \right), \quad (3.30)$$

---

<sup>3</sup>That is, spatial data observed at regular, fixed locations (Wikle & Zammit-Mangion, 2022) with a defined graph structure that captures adjacency.

where  $u_{-i} = \{u_j, j \neq i\}$  denotes the variables  $u_j$  at other locations different from  $s_i$ ,  $n_i$  is the amount of neighbours of location  $s_i$ , and  $\tau_u$  is the precision of  $u$ . We highlight that this method strives for a high local dependence of  $u$ , as a higher amount of neighbours implies a higher precision in (3.30).

Adapting the general definition by Besag et al. (1991), to fulfil the Markov condition (3.30), the joint likelihood of the ICAR vector  $u$  is set as

$$p(u|\tau_u) \propto \exp \left( -\frac{\tau_u}{2} \sum_{j \sim i} (u_i - u_j)^2 \right), \quad (3.31)$$

which could be interpreted as an stochastic formulation of linear interpolation (Besag et al., 1991). Moreover, a centrality condition  $\sum_{i=1}^N u_i = 0$  is usually set to guarantee identifiability of  $u$ , in light of the invariance of (3.31) under scalar displacements of  $u$  (Morris et al., 2019).

Although BYM is originally defined as a log-normal Poisson model for disease mapping, we provide here its Gaussian adaptation for a latent process  $y$  over  $N$  fixed locations. It can be hierarchically stated as

$$\begin{cases} y_i | \mu_i, \sigma^2 \sim \mathcal{N}(\mu_i, \sigma^2), & i = 1, \dots, N, \\ \log(\mu_i) = \beta_0 + \beta^T X_i + u_i + v_i, \end{cases} \quad (3.32)$$

where  $X_i$  are covariates and the terms  $u$  and  $v$  are referred to as the *structured* and *unstructured* spatial effects of the equation, respectively. The term  $u$  is assumed as an ICAR vector following (3.30-3.31) and  $v | \tau_v \sim \mathcal{N}(0, \frac{1}{\tau_v} I)$  is random spatial noise with precision  $\tau_v$ .

### 3.III.ii Inference and Implementation

In the Bayesian inference framework, the model's unspecified parameters  $\theta = (\theta_1, \dots, \theta_k) \in \Theta$  are considered random variables. This, the goal is to estimate their joint distribution conditioned on the observed data  $y$ , denoted by  $p(\theta|y)$ , and maximize it through standard optimization techniques like gradient-based methods (see Section 3.I.iii). This quantity is called the *posterior* distribution and measures the likelihood that the parameters  $\theta$  generated the data  $y$  collected in the studied phenomenon. This relationship is inferred from the Bayes' rule by reversing the parameters-data dependence:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}. \quad (3.33)$$

On the right-hand side of (3.33), the probability  $p(\mathbf{y}|\boldsymbol{\theta})$  is referred to as the *likelihood* distribution. In general, the type of mathematical relationship between the parameters and the retrieved data is stated in this distribution and it captures the likelihood that the outcomes  $\mathbf{y}$  occurred, given the parameters  $\boldsymbol{\theta}$ . The joint parameter distribution  $p(\boldsymbol{\theta})$  reflects previous knowledge of the parameters' values, with some degree of uncertainty. It is thus called the *prior* distribution (Gómez-Rubio, 2020). The denominator in (3.33) is called the *marginal likelihood* and satisfies

$$p(\mathbf{y}) = \int_{\Theta} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad (3.34)$$

Due to the general difficulty in computing this value, Bayes' rule in (3.33) is stated, up to re-scaling, as

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \quad (3.35)$$

In the case of the BYM model, gamma distributions are set for the priors of the precisions  $\tau_u$  and  $\tau_v$  (Besag et al., 1991). The general definition of this class of distributions is parametrically given by two values  $\alpha, \lambda > 0$ , called the *shape* and the *rate* of the distribution, respectively. Explicitly, the underlying probability density function of a  $\text{Gamma}(\alpha, \lambda)$  distribution is

$$f(x; \alpha, \lambda) = \begin{cases} \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\lambda x) & , x > 0, \\ 0 & , x \leq 0. \end{cases} \quad (3.36)$$

Different methodologies exist to optimize the posterior distribution  $p(\boldsymbol{\theta}|\mathbf{y})$  in Bayesian inference. The Integrated Nested Laplace Approximation (INLA) was developed by Rue et al. (2009) to estimate posteriors for latent Gaussian Markov random fields (GMRFs), which include the ICAR and random noise components of the BYM formulation in (3.32).

Though there are alternative and more established approximation methods in Bayesian inference, like the Markov chain Monte Carlo (MCMC) sampling, INLA has shown a good applicability due to its speed<sup>4</sup> and the reduction of errors in comparison with MCMC for this type of models (Rue et al., 2009). INLA succeeds in this inference optimization because it approximates marginal univariate posteriors  $p(\theta_i|\mathbf{y})$  of (3.35), instead of their joint distribution  $p(\boldsymbol{\theta}|\mathbf{y})$  (Gómez-Rubio, 2020). This is achieved by numeric integrations of the type:

$$p(\theta_i|\mathbf{y}) = \int p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}_{-i}, \quad i = 1, \dots, k, \quad (3.37)$$

---

<sup>4</sup>INLA reduces computing times of hours or days in MCMC to seconds or minutes (Martino & Rue, 2009).

carried over sections of the parameter space  $\Theta$  (in general of low dimension<sup>5</sup>).

In Bayesian inference, the selection of a combination of likelihood and prior distributions in (3.35) leads to different families of posterior distributions. As a simple example, the use of Gaussian distributions for the likelihood and the prior distributions ensures a Gaussian posterior distribution (Gómez-Rubio, 2020).

The R package `r-inla`<sup>6</sup> (Rue et al., 2009) implements INLA for multiple latent GMRFs. It includes flexibility to define spatio-temporal dependences on additive processes—including the additive components of BYM models (3.32)—and customizability of prior distributions of the parameters.

---

<sup>5</sup>Martino and Riebler (2019) recommend less than 15 parameters to achieve an efficient performance.

<sup>6</sup>To the date of the writing of this thesis, this package is not available in CRAN. However, it is fully accessible through the R-INLA project's website: [r-inla.org](http://r-inla.org). Accessed on 17 January 2025.

# Chapter 4

## Model Implementations

In this chapter we expose the details about the application of the methods presented in Chapter 3. The first section deals with the feature extraction procedures carried out. Then, the second part explains the model design. A summary of the analysis workflow is shown in Figure 4.3.

### 4.1 Data Preparation

After the data processing stage exposed in Chapter 2, we extracted multiple temporal features and a spatial feature for each cleaned time series. A summary of these features is presented in Table 4.1. These features were incorporated throughout the modelling workflow, though they were mainly set for the core trend modelling task assigned to the LSTM model (see Section 4.II). Moreover, to reduce the variability of the target variable and the complexity of the model, we aggregated all incoming features at the day level. Furthermore, after these transformations, each feature was normalised by sensor (i.e. by time series) to ensure model stability and comparable feature contributions.

The temporal features were readily extracted and processed from the timestamp information. We highlight that two periodic temporal variables with more than 10 classes (*day of the month* and *month of the year*) were treated numerically rather than categorically to simplify the model. Moreover, cosine and sine components of these features were extracted. This aims to avoid model bias towards higher values in late periods of the month or the year and to consider the cyclic nature of these covariates.

As for the spatial feature extraction, we calculated the mean value of power consump-

Table 4.1: Features extracted per time series.

Type of feature extracted	Description
Day of the week	7 classes converted to individual dummy variables
Day of the month	2 cyclic numeric variables
Month of the year	2 cyclic numeric variables
Past power consumption	one-day-before consumption values
Neighbourhood mean consumption	one-day-before mean consumption by neighbouring sensors

tion in a punctured neighbourhood<sup>1</sup> around each sensor across time. The objective of this feature is to account for possible local dependence of energy usage patterns.

The critical step in this procedure was the definition of the size of the neighbourhood of influence, that is, an optimal buffer distance without introducing excessive noise to the model. To this end, we performed a semivariogram analysis, following (3.29) and (R. S. Bivand et al., 2013, §8.4). We stress that, for simplicity, the isotropic and stationarity conditions mentioned in (3.28) were assumed.

The analysis was conducted at the hourly level of temporal aggregation, to account for the varying temporal recurrence of values across sensors (every 15 minutes or every 30 minutes). Four different dates were selected from the time span of the dataset (March 1st 2023, September 1st 2023, March 1st 2024, September 1st 2024) and twelve hourly readings (every two hours) were selected per date.

To perform the semivariogram analysis, we used the R package `gstat` (E. J. Pebesma, 2004). The exponential fitting method was selected, with a corresponding theoretical semivariogram given by

$$\gamma(h) = c \left( 1 - \exp \left( \frac{3h}{a} \right) \right), \quad h \geq 0, \quad (4.1)$$

where  $c$  is the *partial sill* of the semivariogram —its asymptotic value— and  $a$  its *range* —the distance at which 95% of  $c$  is reached—. After initial calculations of the empirical variogram, we selected a cutoff distance of 20 m and a width of 0.67 m, so that a total of 30 intervals were used to approximate the semivariograms (R. S. Bivand et al., 2013). An example of a fitted semivariogram with this approach is shown in Figure 4.1.

The estimated value for the range  $\hat{a}$  was selected as the reference quantity to deduce

---

<sup>1</sup>That is, considering sensors around the reference location without data from that location.

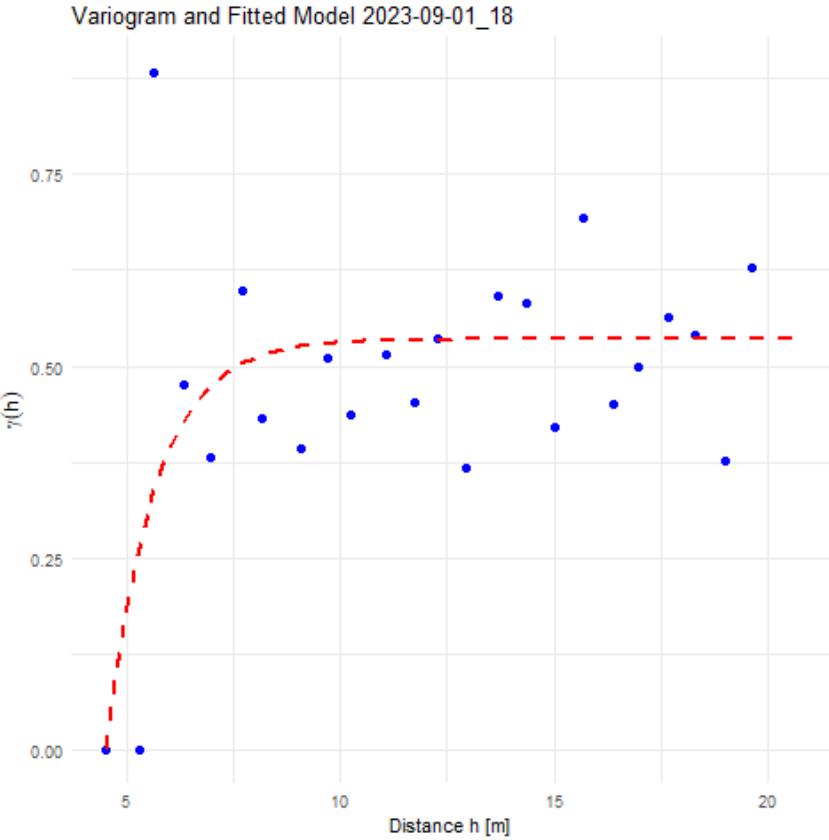


Figure 4.1: Semivariogram estimation for September 1st, 2023 at 18:00.

the optimal buffer distance. The mean of the estimated range across the 48 semivariograms was 5.79 m. In Table 4.2, we present a comparison of mean estimations of the range among the four studied dates. However, as some time slices showed higher values<sup>2</sup>, a more flexible final buffer of 10 m was set for the project.

Table 4.2: Mean range values from hourly semivariograms in four dates.

Date	Mean of estimated range $\hat{a}$ [m]
2023-03-01	5.14
2023-09-01	6.07
2024-03-01	6.38
2024-09-01	5.61

In order to develop a forecast model using the baseline LSTM setting, a suitable tensor structure was to be provided as an input. To this end, a prediction window of size  $w > 0$

<sup>2</sup>The highest estimated range value obtained was 14 m.

needs to be defined, so that for each set of  $h$ -lagged values of predictors  $\{\mathbf{x}_{t-h+1}, \dots, \mathbf{x}_t\}$ , corresponding outcomes  $\mathbf{y}_t = \{y_{t+1}, \dots, y_{t+w}\}$  are deduced (see Table 4.4 for setting details).

For simplicity of the LSTM model, we kept  $w = 1$ , that is, the model predicts a one-day-ahead consumption per sensor. Moreover, this setting facilitates the incorporation of the baseline LSTM trend model to the Bayesian setting later discussed.

To perform Bayesian inference using the BYM model described in Section 3.III.i, we obtained a graph object representing the adjacency relationships of the sensors. To achieve this, we calculated voronoi regions of influence per sensor and deduced their corresponding adjacency structure. We highlight that two different buffer distances were selected to limit the size of the voronoi polygons: 10 m distance –following the insights from the semi-variogram analysis previously discussed– and 30 m distance. The intention of this setting was to run different BYM models and study how could contrasting graph structures affect the outcomes.

The two adjacency situations are depicted in Figure 4.2. We highlight that the main difference among these two structures is the number of connected components. As shown in the figure, the voronoi collection with the 10m buffer distance induces a graph with 4 connected components (green edges), whereas the 30m buffer produces a connected graph (red and green edges). The neighbourhood relationships were obtained using the `spdep` package (R. Bivand & Wong, 2018).

In preparation for modelling, the data was split in train and test data in a 80-20% ratio. The cut date with this setting is the end of May 2nd 2024.

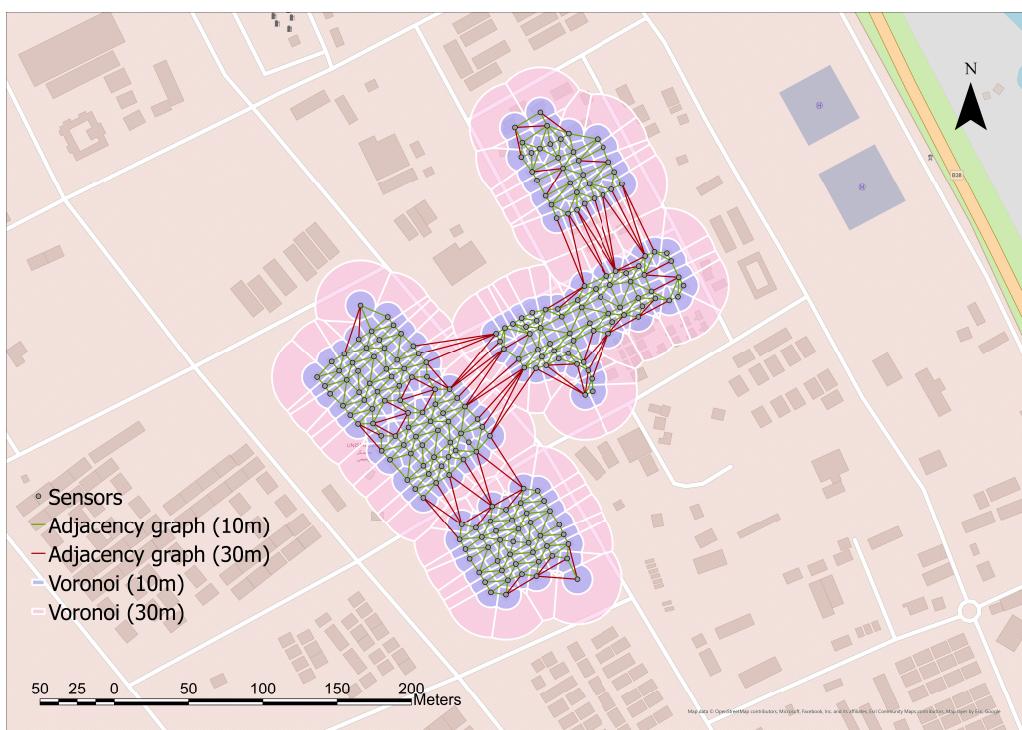


Figure 4.2: Overlay of the two voronoi collections (10m and 30m buffer distances) and their corresponding adjacency graph structures.

## 4.II Model Design

To start the modelling stage, a baseline LSTM model was used as the core forecast model. Its main purpose is to predict the trend structure of the consumption patterns. Two main reasons made LSTM suitable for this task.

First, the distribution of the consumption data per time series (see Figure 2.3) made it difficult to apply a statistical method that assumed known types of distributions, e.g. normality. Though some transformations were tried out following (Browle, 2020, Chapter 8) (e.g. log, Box-Cox), none resulted consistently optimal to solve the high skewness. Second, as discussed in Section 3.I.ii, LSTM models have proven to be a good tool to overcome vanishing problems. This could be useful to detrend the consumption patterns and model seasonality. This intention also justifies the addition of multiple temporal variables to the model, as described in the previous section (see Table 4.1).

Regarding the architecture of the LSTM network, we followed a base-2 exponential design of two hidden layers layers. This setting generated a total of 35,281 to 35,537 parameters, depending on the inclusion of the spatial neighbourhood mean feature. Table 4.3 presents the chosen LSTM structure. The output size for the hidden layers represents the number of lagged temporal values times the number of cells per layer. We chose the ReLU function as the activation method between hidden layers, to reduce the complexity of the gradient flow calculations across layers. As for the activation between hidden units (LSTM cells), we set the sigmoid function  $\sigma$ , following the discussion in Section 3.I.ii. After adjusting the parameters, we set their final values for all sensors as shown in Table 4.4.

Table 4.3: LSTM model design.

Layer	Output size
Input layer	$14 \times 64 \times \text{number of variables}$
Hidden layer 1	$14 \times 64$
Hidden layer 2	$14 \times 32$
Flattening layer	16
Dense layer	1

Table 4.4: Parameter settings for LSTM models.

Parameter	Value
Optimizer	Adam
Learning rate	1e-3
Loss function	Mean squared error
Window $w$ size	1 (one-day-ahead prediction)
Lag $h$ extent	14 (2 weeks)
Batch size	56 ( $\sim 2$ months of data)
Number of epochs	20
Inter-layer activation	ReLU
Inter-cell activation	sigmoid $\sigma$

As these LSTM settings were applied for all sensors, corrective individual ARIMA models were implemented to delete any possible remaining seasonal or autoregressive effect. To this end, we leveraged the automatic ARIMA models explained in Section 3.II.v. We also selected this approach due to its reduced time complexity.

Finally, the LSTM+ARIMA models results provided better evidence of normality of their residuals. Thereupon, the predicted values from these combined models were taken as mean (trend) values  $\mu$  for a spatio-temporal BYM model on the residuals, given by

$$\begin{cases} y_{it} | \mu_{it}, \sigma_y^2 \sim \mathcal{N}(\mu_{it}, \sigma^2) \\ \log(y_{it} - \mu_{it} + \delta) = \beta_0 + u_i + v_i + \epsilon_t, \end{cases} \quad (4.2)$$

for sensor locations  $i = 1, \dots, N$  and an offset  $\delta > 0$ , where

1.  $u$  is an ICAR vector, as in (3.30-3.31),
2.  $v | \tau_v \sim \mathcal{N}(0, \frac{1}{\tau_v} I)$  is the spatial unstructured effect,
3.  $\epsilon | \tau_\epsilon \sim \mathcal{N}(0, \frac{1}{\tau_\epsilon} I)$  temporal white noise.

The main technicality to consider in the Bayesian inference setting of this model is the configuration of the priors with `r-inla` (R-INLA Project, 2025). Regarding this, we set vague priors, so that the optimization was lead by the data (Gómez-Rubio, 2020). These are presented in Table 4.5.

Table 4.5: Prior settings for precisions  $\tau_u$ ,  $\tau_v$ , and  $\tau_e$ .

Quantity	Value
$\alpha$	1
$\lambda$	1e-2
$E(\text{Gamma}(\alpha, \lambda))$	$\alpha\lambda^{-1} = 100$
$\text{Var}(\text{Gamma}(\alpha, \lambda))$	$\alpha\lambda^{-2} = 10,000$

To finalize the chapter, we condense in Figure 4.3 the complete data preparation and modelling from this analysis stage of the project. Green boxes represent spatial analysis whereas blue ones indicate data transformations on the temporal dataframes.

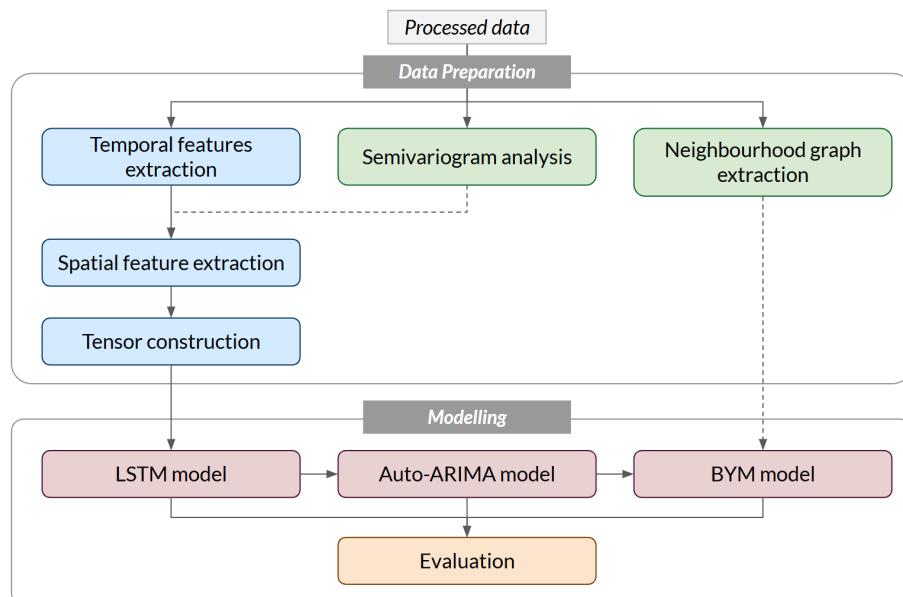


Figure 4.3: Data preparation and modelling workflow.

# Chapter 5

## Results

In this chapter, we present the outcomes of our projects. We break the exposure in two sections. In the first one, we showcase the performance of the different models described in Chapter 4 and in the second one we discuss in more detail the findings of our work.

### 5.1 Model Results

To begin with, we comment that the models were trained on the first 80% of the data (train data) and the progressive application of them was evaluated on the remaining 20% (test data).

An example of the three main models fitting the one-day-ahead values for the test set of a single sensor is shown in Figure 5.1. This figure exhibits the intended error correction done by the Auto-ARIMA approach over the LSTM results. However, in this example there is no visible improvement coming from the BYM model. In Section 5.II, we will discuss this in more detail and argument that this is the general case.

Regarding the evaluation metrics, we used two main standard metrics for time series analysis (Cheng et al., 2022): the Root Mean Square Error (RMSE) to address the size of the forecast error, and the coefficient of determination ( $R^2$ ), to account for the variability of the time series values.

In the modelling stage, we ran four types of model combinations: baseline LSTM models, LSTM models corrected with AutoARIMA, and BYM models applied to the residuals of the LSTM+AutoARIMA models with either 10 or 30 meters of buffer distance considered for the definition of the neighbourhood relationship. Furthermore, we ran the baseline LSTM model with and without the spatial feature (neighbourhood mean) discussed in Sec-

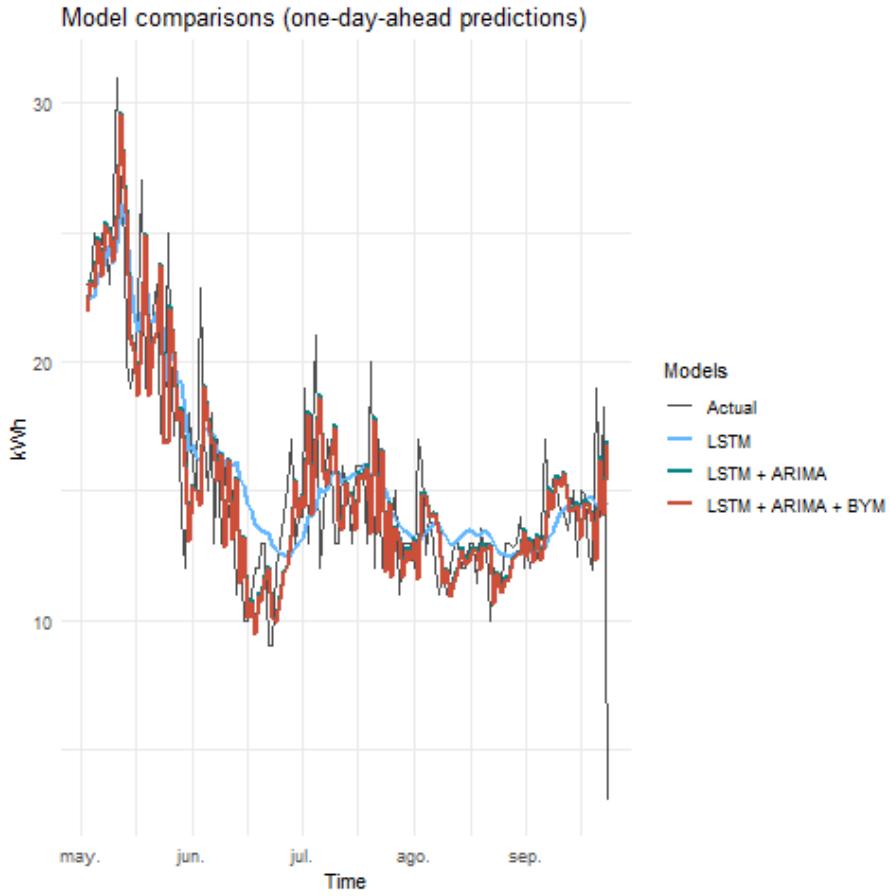


Figure 5.1: Comparison of models fitting the test dataset for an individual sensor.

tion 4.I. Therefore, a total of 8 model combinations were evaluated. We provide the overall performance of the model combinations in Table 5.1. These figures were calculated across all the modelled time series. Moreover, we depict these quantities in Figures 5.2-5.3.

All in all, the best models were the most complex combinations (LSTM+AutoARIMA+BYM models). In terms of the test RMSE, the best performing model is the LSTM+AutoARIMA+BYM model with the spatial feature. The reduction of test RMSE is of 6.45% for the models with the spatial feature and of 4.83% for those without it. Also, there is a generally small difference of the test RMSE between models with and without the spatial feature. Likewise, we highlight that almost no difference was noticeable in terms of accuracy between different buffer distance definitions for the neighbourhood relationship.

We also calculated these metrics per sensor, to further study the variability of prediction performance (see Section 5.II). Here, we present the distribution of the test RMSE across all sensors in Figures 5.4-5.5. In this case, we also note the reduction of the mean of the test RMSE from the baseline LSTM after the application of the three other model im-

provements. Also, in this analysis, another difference between the inclusion and exclusion of the spatial feature is the reduced dispersion of the test RMSE values from the baseline LSTM model.

Table 5.1: Performance by type of model.

<i>Model</i>	<i>train RMSE</i>	<i>train R</i> <sup>2</sup>	<i>test RMSE</i>	<i>test R</i> <sup>2</sup>	Spatial feature
LSTM	2.80	80.53	3.96	65.28	Yes
LSTM+AutoARIMA	2.71	81.79	3.73	69.18	Yes
LSTM+AutoARIMA+	2.51	84.31	3.72	69.46	Yes
BYM(10m)					
LSTM+AutoARIMA+	2.51	84.31	3.72	69.46	Yes
BYM(10m)					
LSTM	2.82	80.11	3.91	66.68	No
LSTM+AutoARIMA	2.76	80.95	3.75	69.28	No
LSTM+AutoARIMA+	2.57	83.54	3.73	69.62	No
BYM(10m)					
LSTM+AutoARIMA+	2.57	83.55	3.73	69.62	No
BYM(30m)					

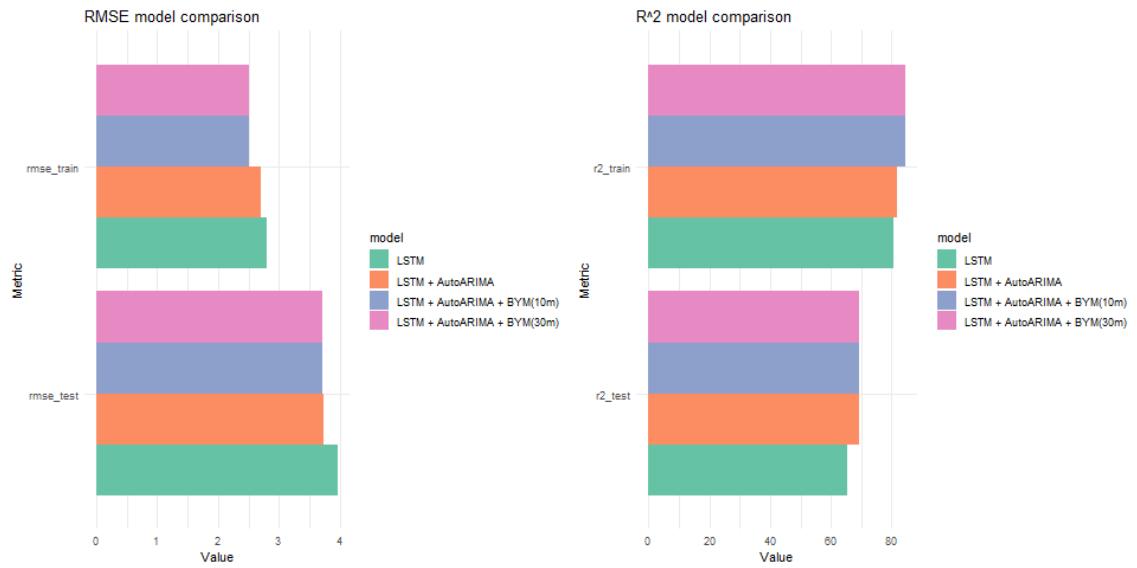


Figure 5.2: RMSE (left) and  $R^2$  (right) comparison across models with spatial feature.

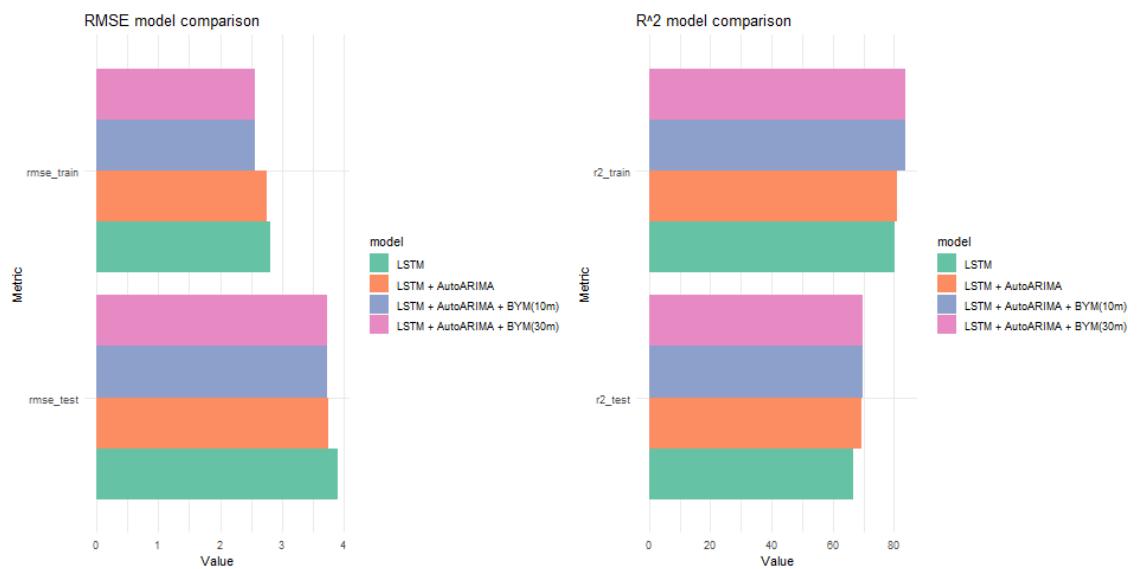


Figure 5.3: RMSE (left) and  $R^2$  (right) comparison across models without spatial feature.

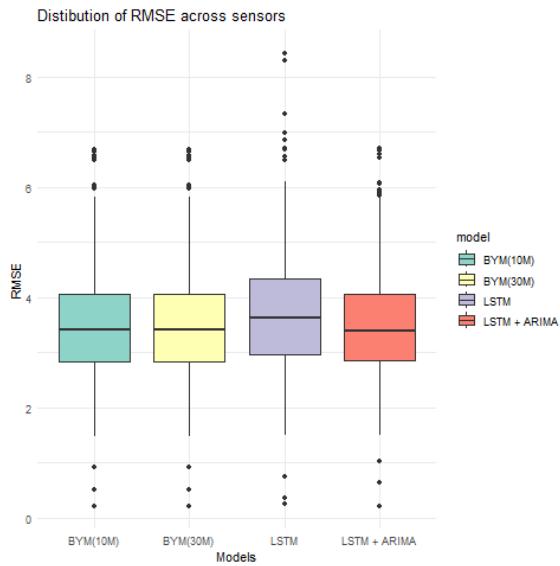


Figure 5.4: Test RMSE distribution across sensors (models with spatial feature).

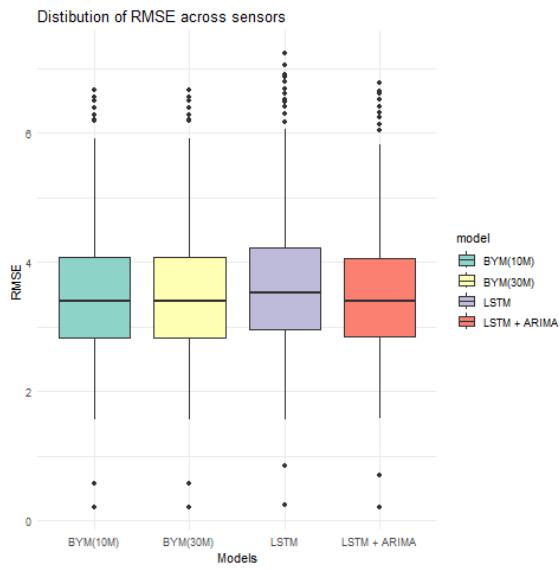


Figure 5.5: Test RMSE distribution across sensors (models without spatial feature).

An important outcome of the BYM models is the contribution of the spatial structured and unstructured effects  $u$  and  $v$  in (4.2), respectively. The estimated posterior mean of these variables is represented in Figures 5.6-5.7. Though these values come at the log scale, we verify that the structured effect was highly dominated by the random spatial component of the BYM model across the four LSTM+AutoARIMA+BYM models. Specifically, the contribution of the latter overcomes the former by the order of the thousands. We also notice that for the 30m-buffer case of the the model without the spatial feature, higher values of  $v_i$  appear to cluster in the centre of the camp (Figure 5.7).



Figure 5.6: BYM spatial effects with the spatial feature for models with a buffer distance of 30m (top) and 10m (bottom).



Figure 5.7: BYM spatial effects without the spatial feature for models with a buffer distance of 30m (top) and 10m (bottom).

## 5.II Results Discussion

In light of all the findings of this work, to answer Q1 we show how the multiple processing and modelling approaches taken ensured acceptable forecast performances across the model combinations. Moreover, as we will further explain, the lingering forecast error highly depends on the variability of the data.

Likewise, in reference to Q2, we will show that the spatial feature within the baseline LSTM model provided some degree of stability when adding AutoARIMA corrections and less degree of relationship between prediction errors and clustering of the sensors. Also, the use of Spatial Statistics helped us to slightly increase model fits. However, minimal insight from the posterior inference of structured effects could be found. We attribute this to the great dependence of the problem to complex human factors, random noise coming from local malfunction, or temporal variability of the readings.

Power consumption modelling presents several challenges. One of the most important is the quality of the data, as discussed with the matter experts from the UNGSC. Due to technical issues coming from the local IoT networks, a heavy amount of noisy readings tend to populate time series of power consumption, as already commented in Chapter 2. However, given this as a premise, the models achieved acceptable results in terms of accuracy. Likewise, the models showed similar  $R^2$  performance across the possible combinations

Furthermore, the adaptive nature of the AutoARIMA method proved to have a main role in RMSE reduction. This was possible to achieve due to the individual optimal calculation of autoregressive and moving-average orders. Within the AutoARIMA correction, we retrieved these orders. The proportion of their values across sensors are shown in Figure 5.8.

In general, low degrees of differencing (order  $d$ ) were chosen by the algorithm designed by R. J. Hyndman and Khandakar (2008). Also, autocorrelation and moving average orders tend to be below 2. This suggests that the detrending and seasonality modelling was efficiently carried out, all in all, by the LSTM architecture.

We highlight that, though minimal in quantity, higher autoregressive  $p$  orders were applied after baseline LSTM models without the spatial feature. The reason behind this could be that the neighbourhood feature efficiently provided valuable information for local long-term inference of power consumption. Moreover, looking into detail to those individual cases (4 sensors with  $p > 5$ ), the RMSE improvement done by the AutoARIMA procedure was either negative or not important in comparison to the rest of the sensors, as shown

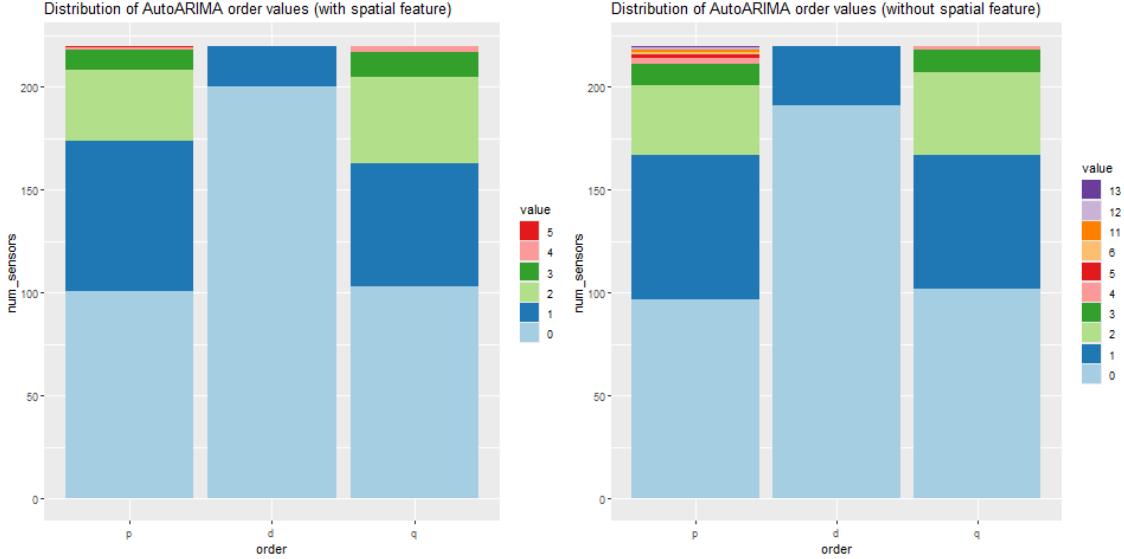


Figure 5.8: Distribution of order values of AutoARIMA with (left) and without (right) the spatial feature.

in Table 5.2. This could mean that the addition of the spatial feature in the baseline LSTM provides more stability, in the sense that it avoids atypically large autoregressive corrective models with worse prediction.

Table 5.2: RMSE improvement by AutoARIMA for four sensors with  $p > 5$ .

Sensor id	AutoRIMA RMSE improvement	Z-score
1-128-349	0.13	-0.04
1-24-540	-0.13	-0.95
1-47-586	-0.18	-1.15
1-852-710	0.14	0.01

Regarding the BYM effects, we should note that the dominant role of the unstructured effect strongly suggests that the errors from the LSTM+AutoARIMA model show low spatial interdependence. According to Besag et al. (1991), when this occurs, the model tends to shrink the estimated posterior towards its mean. That is, this prevents to gain any spatial insight from  $y_{it} - \mu_{it}$  in (4.2). In particular, this occurs on the BYM models over the LSTM+AutoARIMA without the spatial feature, that is, in models without any kind of spatial insight in the baseline temporal model. This is a strong evidence of lack of structure

between the network of sensors.

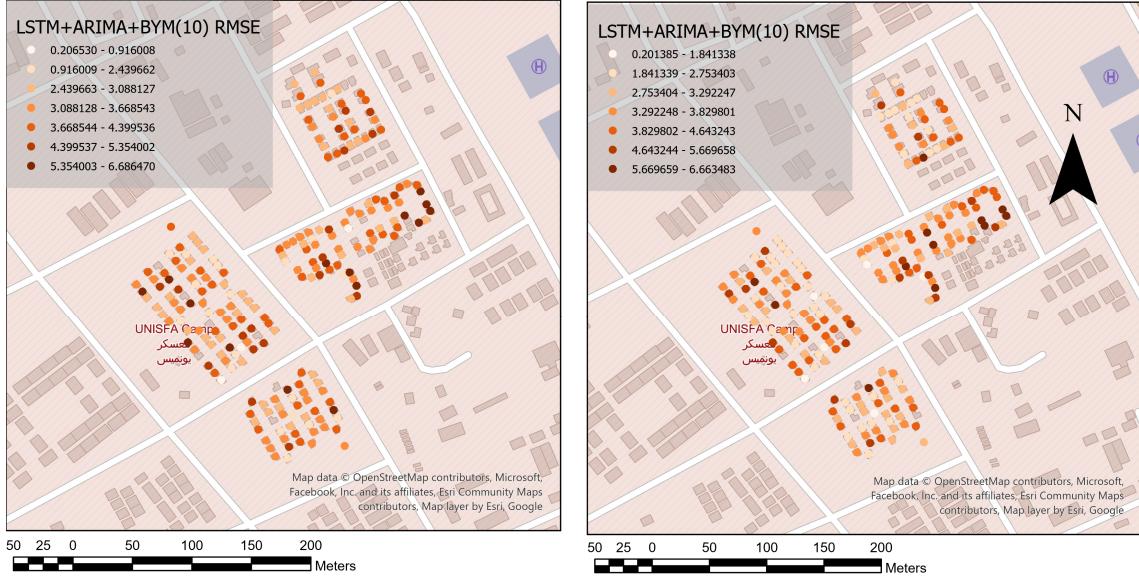


Figure 5.9: Spatial distribution of RMSE for the LSTM+AutoARIMA+BYM(10) models with (left) and without (right) the spatial feature.

In Figure 5.9, we show the spatial distribution of RMSE with and without spatial features for the best-ranking models within these two categories (one of the LSTM+AutoARIMA+BYM combinations). Though no much spatial structure could be obtained by means of the ICAR component of BYM, we notice that higher errors in this model combinations tend to accumulate in border regions. We argument now this hypothesis.

To better explain error distribution, we carried out a correlation study on the eight test RMSE metrics across sensors compared to some temporal and spatial features. The results are shown in Tables 5.3-5.4. From a temporal perspective, we extracted per sensor the variance and standard deviation of the moving variance<sup>1</sup> of the power consumption readings. From a spatial perspective, we evaluated four key features: the amount of neighbours  $n_{10}$  and  $n_{30}$  (within a 10m and 30m distance, respectively, in line with the BYM settings), the distance  $d_1$  to the camp centroid, and the distance  $d_2$  to the centroid of the connected

<sup>1</sup>Calculated using a moving window of size 14, in line with the lag parameter from the LSTM (see Table 4.4).

component to which each sensor belongs to <sup>2</sup>. We highlight that these connected components correspond to blocks of buildings, by visual inspection of the overlay with OSM data in Figures 4.2 and 5.9.

Table 5.3: Correlation analysis of test RMSE (models with spatial feature) ( $*p < 0.05$ ,  $**p < 0.01$ ).

	Var	$\sigma(\text{Var}_t)$	$n_{10}$	$n_{30}$	$d_1$	$d_2$
LSTM	0.34**	0.32**	-0.1	-0.1	0.02	0.13*
LSTM+AutoARIMA	0.34**	0.34**	-0.12	-0.12	0.04	0.16*
LSTM+AutoARIMA+BYM30	0.33**	0.34**	-0.12	-0.12	0.04	0.15*
LSTM+AutoARIMA+BYM10	0.33**	0.34**	-0.12	-0.12	0.04	0.15*

Table 5.4: Correlation analysis of test RMSE (models without spatial feature) ( $*p < 0.05$ ,  $**p < 0.01$ ).

	Var	$\sigma(\text{Var}_t)$	$n_{10}$	$n_{30}$	$d_1$	$d_2$
LSTM	0.43**	0.38**	-0.15*	-0.15*	0.02	0.09
LSTM+AutoARIMA	0.40**	0.38**	-0.13	-0.13	0.01	0.11
LSTM+AutoARIMA+BYM30	0.40**	0.38**	-0.13	-0.13	0.01	0.11
LSTM+AutoARIMA+BYM10	0.40**	0.38**	-0.13	-0.13	0.01	0.11

According to this results, throughout the eight models, the error is significantly related with the global and local variability of the time series. Moreover, it is interesting to note the different meaningful correlations with the four spatial key features  $n_{10}$ ,  $n_{30}$ ,  $d_1$ , and  $d_2$ , between both tables. Accordint to Table 5.3, in the models with the spatial feature included, there appears to be a significant dependence of test RMSE on distance to the connected components' centroids. We could interpret this as the fact that the error increases in those sensors at the borders of their connected component, due to the lack of spatial information from their neighbours. On the other hand, looking at Table 5.4, in some meaningful level we could attribute the LSTM baseline error to the lack of spatial context as shown by columns  $n_{10}$  and  $n_{30}$ , that capture a degree of clustering.

---

<sup>2</sup>The connected components were extracted from the the BYM(10m) model settings. This is depicted by the four connected green subgraphs in Figure 4.2.

# Chapter 6

## Future Work and Conclusions

In this final chapter, we provide possible future paths within this research on IoT-ECP, as well as limitations found throughout the project. Finally, we present our main conclusions.

### 6.1 Future Work

As it is visible in Figures 5.2-5.3, the models show an overall slight overfitting. This is visible in the  $R^2$  reduction of more than ten percentage points between train and test metrics. This could be due to the high variability of the data itself. Another possible cause could be the complexity of the LSTM model. In that case, improvement can be done by adding regularization to the loss function. This penalization technique forces sparse weights across the LSTM network (Goodfellow et al., 2016, Chapter 7). For simplicity on a baseline model, this was initially excluded in our project.

In this work, we provided a completely independent workflow that leveraged just the autoregressive data and the spatial components of consumption. Nonetheless, another strategy to overcome the overfitting that might come from training on the regressive data, would be the inclusion of more predictive variables. As it is visible in literature (Cheng et al., 2022), indoor and outdoor weather covariates are also useful to predict consumption. Other authors also include human-related features, such occupancy rates (Zeng et al., 2020).

The correlation analysis shown in the previous section suggests that improvement could be done regarding temporal analysis, due to the great temporal variability of the data. We could leverage further additions on the ARIMA approach, for instance, by avoiding temporal stationarity assumptions and consider some volatility effects. Generalized Autoregres-

sive Conditional Heteroskedasticity (GARCH) models, first presented by Bollerslev (1986), stand out as a possible starting solution.

Lately, DL techniques have been widely adopted in modelling IoT-ECP (Cheng et al., 2022). However, there remains a general lack of more explainable models in this setting. Gilanifar, Parvania, and Hariri (2020) show criticism against neural network models, as these single-task learning algorithms may be inaccurate when there is limited historical training data. Nonetheless, the possibility of leveraging Spatial Statistics in synergy with DL methods is available in recent works, for instance in Wikle and Zammit-Mangion (2022). According to the researchers, the so-called *Deep Learning revolution* has led scientists to disregard the advantageous higher interpretability and uncertainty quantification capabilities of statistical models. As improvements to our project, we consider that a more detailed exploration of this compound approach could enhance the interpretability of the LSTM frameworks in IoT-ECP.

## 6.II Conclusions

In this thesis, we exposed a complete workflow of processing and modelling of power consumption data retrieved by IoT sensors spread in a peacekeeping camp of the UNISFA mission during two years. We deployed eight model combinations that use DL, Autoregressive and Spatial Statistics methods. Our results rely strongly in the long-term predictive power of an LSTM neural network architecture, which was mainly used as a baseline trend model.

We implemented an automatic ARIMA algorithm to further correct LSTM one-day-ahead forecasts made per sensors. These models account for temporal autocorrelation still present in the LSTM predictions. The addition of a spatial feature that captures local mean consumption decreased outlier cases of long-term autocorrelation deduced by the ARIMA method. This showed the positive effect of including spatial features in the recurrent baseline framework.

We extracted the structured and random spatial dependence of errors through the application of a BYM model. Thereupon, we showed the high degree of spatial randomness of forecast errors from the LSTM and AutoARIMA methods. Also, this spatial model showed similar results for two tested different criteria of neighbourhood relationships defined across the sensors.

Through the methods presented we achieved suitable forecast performances and deduced possible causes of lingering forecast errors, namely, variability of the data. How-

ever, we demonstrated the added value of the spatial feature in the baseline LSTM model in terms of stability after adding correcting models and less dependence of performance with spatial clustering of sensors.

The presented methods could be used by IoT users to better understand and identify outlier consumption patterns through the spatial and temporal location of prediction errors. Moreover, a one-day-ahead prediction could be carried out to anticipate for incoming anomalous consumptions. Further advances may be added to the project, regarding uncertainty quantification or inclusion of volatility in temporal models.

## References

- Alberg, D., & Last, M. (2018, 9). Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. *Vietnam Journal of Computer Science*, 5, 241-249. Retrieved from <https://link.springer.com/article/10.1007/s40595-018-0119-7> doi: 10.1007/s40595-018-0119-7
- Besag, J., York, J., & Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43, 1-20. Retrieved from <https://doi.org/10.1007/BF00116466> doi: 10.1007/BF00116466
- Bivand, R., & Wong, D. W. S. (2018). Comparing implementations of global and local indicators of spatial association. *TEST*, 27(3), 716–748. Retrieved from <https://doi.org/10.1007/s11749-018-0599-x> doi: 10.1007/s11749-018-0599-x
- Bivand, R. S., Pebesma, E., & Gómez-Rubio, V. (2013). *Applied Spatial Data Analysis with R* (2nd ed.). Springer New York. Retrieved from <https://doi.org/10.1007/978-1-4614-7618-4> doi: 10.1007/978-1-4614-7618-4
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327. Retrieved from <https://www.sciencedirect.com/science/article/pii/0304407686900631> doi: [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1)
- Brockwell, P. J., & Davis, R. A. (2016). Arma models. In *Introduction to Time Series and Forecasting* (pp. 73–96). Cham: Springer International Publishing. Retrieved from [https://doi.org/10.1007/978-3-319-29854-2\\_3](https://doi.org/10.1007/978-3-319-29854-2_3) doi: 10.1007/978-3-319-29854-2\_3
- Browle, J. (2020). *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future*. Machine Learning Mastery.
- Cheng, Y. L., Lim, M. H., & Hui, K. H. (2022). Impact of internet of things paradigm towards energy consumption prediction: A systematic literature review. *Sustainable Cities and Society*, 78, 103624. Retrieved from <https://www.sciencedirect.com/science/article/pii/S221067072100888X> doi: 10.1016/j.scs.2021.103624
- Chollet, F., Kalinowski, T., & Allaire, J. (2022). *Deep Learning with R, Second Edition* (2nd ed.). Manning.
- Dekking, F. M., Kraaijkamp, C., Lopuhaä, H. P., & Meester, L. E. (2005). *A Modern Introduction to Probability and Statistics* (1st ed.). Springer London. Retrieved from <https://link.springer.com/book/10.1007/1-84628-168-7> doi: <https://doi.org/>

10.1007/1-84628-168-7

- Duchi, J., Hazan, E., & Singer, Y. (2011, July). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12(null), 2121–2159.
- Fang, K., & Shen, C. (2020). Near-Real-Time Forecast of Satellite-Based Soil Moisture Using Long Short-Term Memory with an Adaptive Data Integration Kernel. *Journal of Hydrometeorology*, 21(3), 399 - 413. Retrieved from <https://journals.ametsoc.org/view/journals/hydr/21/3/jhm-d-19-0169.1.xml> doi: 10.1175/JHM-D-19-0169.1
- Fong, S., Li, J., Song, W., Tian, Y., Wong, R. K., & Dey, N. (2018, 8). Predicting unusual energy consumption events from smart home sensor network by data stream mining with misclassified recall. *Journal of Ambient Intelligence and Humanized Computing*, 9, 1197-1221. Retrieved from <https://link.springer.com/article/10.1007/s12652-018-0685-7> doi: 10.1007/s12652-018-0685-7
- Gilanifar, M., Parvania, M., & Hariri, M. E. (2020). Multi-Task Gaussian Process Learning for Energy Forecasting in IoT-Enabled Electric Vehicle Charging Infrastructure. In *2020 ieee 6th world forum on internet of things (wf-iot)* (p. 1-6). Retrieved from <https://ieeexplore.ieee.org/document/9221159> doi: 10.1109/WF-IoT48130.2020.9221159
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Sequence Modeling: Recurrent and Recursive Nets. In *Deep Learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>
- Gómez-Rubio, V. (2020). *Bayesian Inference with INLA* (1st ed.). Chapman & Hall/CRC Press. Retrieved from <https://doi.org/10.1201/9781315175584> doi: 10.1201/9781315175584
- Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.
- Hinton, G. (2012). *Neural networks for machine learning*. Retrieved from [https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf) (Coursera video lectures)
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. Retrieved from <https://doi.org/10.1162/neco.1997.9.8.1735> doi: 10.1162/neco.1997.9.8.1735
- Hultman, L., Kathman, J., & Shannon, M. (2013). United Nations Peacekeeping and Civilian Protection in Civil War. *American Journal of Political Science*, 57(4), 875-

891. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1111/ajps.12036> doi: 10.1111/ajps.12036
- Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., ... Yasmeen, F. (2024). *forecast*: Forecasting functions for time series and linear models [Computer software manual]. Retrieved from <https://pkg.robjhyndman.com/forecast/> (R package version 8.23.0)
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles & Practice* (2nd ed.). OTexts. Retrieved from <https://otexts.com/fpp2/>
- Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The *forecast* Package for R. *Journal of Statistical Software*, 27(3), 1–22. Retrieved from <https://www.jstatsoft.org/index.php/jss/article/view/v027i03> doi: 10.18637/jss.v027.i03
- Kalinowski, T., Allaire, J., & Chollet, F. (2024). *keras3*: R Interface to ‘Keras’ [Computer software manual]. Retrieved from <https://CRAN.R-project.org/package=keras3> (R package version 1.1.0)
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction. *Transactions of the ASME - Journal of Basic Engineering*, 82, 35-45.
- Khan, A. N., Iqbal, N., Ahmad, R., & Kim, D.-H. (2021). Ensemble Prediction Approach Based on Learning to Statistical Model for Efficient Building Energy Consumption Management. *Symmetry*, 13(3). Retrieved from <https://www.mdpi.com/2073-8994/13/3/405> doi: 10.3390/sym13030405
- Kingma, D. P., & Ba, J. (2014, 12). Adam: A Method for Stochastic Optimization. *arXiv preprint: arXiv:1412.6980*. Retrieved from <https://arxiv.org/abs/1412.6980> doi: 10.48550/arXiv.1412.6980
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1), 159-178. Retrieved from <https://www.sciencedirect.com/science/article/pii/030440769290104Y> doi: [https://doi.org/10.1016/0304-4076\(92\)90104-Y](https://doi.org/10.1016/0304-4076(92)90104-Y)
- Körner, M., & Rußwurm, M. (2021, 8). Recurrent neural networks and the temporal component. In *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science and Geosciences* (p. 105-119). Wiley. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119646181.ch8> doi:

10.1002/9781119646181.ch8

- Li, W., Logenthiran, T., Phan, V., & Woo, W. L. (2019). A Novel Smart Energy Theft System (SETS) for IoT based Smart Home. *IEEE Internet of Things Journal*, 6. Retrieved from <https://ieeexplore.ieee.org/document/8661504> doi: 10.1109/JIOT.2019.2903281
- Martino, S., & Riebler, A. (2019, 7). Integrated Nested Laplace Approximations (INLA). *arXiv preprint: arXiv:1907.01248*. Retrieved from <https://arxiv.org/abs/1907.01248>
- Martino, S., & Rue, H. (2009, 9). Implementing Approximate Bayesian Inference using Integrated Nested Laplace Approximation: a manual for the inla program [Computer software manual]. Retrieved from <https://www.r-inla.org/what-is-inla>
- Moritz, S., & Bartz-Beielstein, T. (2017). imputeTS: Time Series Missing Value Imputation in R. *The R Journal*, 9(1), 207–218. Retrieved from <https://doi.org/10.32614/RJ-2017-009> doi: 10.32614/RJ-2017-009
- Morris, M., Wheeler-Martin, K., Simpson, D., Mooney, S. J., Gelman, A., & DiMaggio, C. (2019). Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in stan. *Spatial and Spatio-temporal Epidemiology*, 31, 100301. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1877584518301175> doi: <https://doi.org/10.1016/j.sste.2019.100301>
- Niraula, P., Mateu, J., & Chaudhuri, S. (2022). A Bayesian machine learning approach for spatio-temporal prediction of COVID-19 cases. *Stochastic Environmental Research and Risk Assessment*, 36, 2265-2283. Retrieved from <https://doi.org/10.1007/s00477-021-02168-w> doi: 10.1007/s00477-021-02168-w
- Pebesma, E., & Bivand, R. (2023). *Spatial Data Science: With applications in R*. Chapman and Hall/CRC. Retrieved from <https://r-spatial.org/book/> doi: 10.1201/9780429459016
- Pebesma, E. J. (2004). Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30(7), 683-691. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0098300404000676> doi: 10.1016/j.cageo.2004.03.012
- R-INLA Project. (2025). Besag model documentation [Computer software manual]. Retrieved from <http://www.r-inla.org/> (Accessed on 17 December 2024 via the 'inla.doc('besag')' function.)
- Rabie, A. H., Ali, S. H., Saleh, A. I., & Ali, H. A. (2020, 1). A fog based load forecasting strategy based on multi-ensemble classification for smart grids. *Journal of Ambient Intelligence*

- and Humanized Computing*, 11, 209-236. Retrieved from <https://link.springer.com/article/10.1007/s12652-019-01299-x> doi: 10.1007/s12652-019-01299-x
- Rue, H., Martino, S., & Chopin, N. (2009, 4). Approximate Bayesian Inference for Latent Gaussian models by using Integrated Nested Laplace Approximations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71, 319-392. Retrieved from <https://doi.org/10.1111/j.1467-9868.2008.00700.x> doi: 10.1111/j.1467-9868.2008.00700.x
- Rußwurm, M., & Körner, M. (2017). Temporal Vegetation Modelling Using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-spectral Satellite Images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (p. 1496-1504). Retrieved from <https://ieeexplore.ieee.org/document/8014927> doi: 10.1109/CVPRW.2017.193
- Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications*. Springer International Publishing. Retrieved from <https://doi.org/10.1007/978-3-319-52452-8> doi: 10.1007/978-3-319-52452-8
- Souza, W., Souto, E., Rocha, T., Pereira, N., Kool, P., & Rosengren, P. (2016). Prediction of Electrical Energy Consumption for Internet of Things in Disaggregated Databases. *2016 IEEE Symposium on Computers and Communication (ISCC)*, 429-434. Retrieved from <https://ieeexplore.ieee.org/document/7543777> doi: 10.1109/ISCC.2016.7543777
- UN Geospatial Information Section. (2023, May). *UNISFA May 2023 [cartographic material]*. UN Geospatial Information Section,. Retrieved from <http://digitallibrary.un.org/record/4011934>
- United Nations. (2011, Jun). *Resolution 1990* (2011). United Nations. Retrieved from <http://digitallibrary.un.org/record/706063> (Issued in: Resolutions and decisions of the Security Council, 1 August 2010-31 July 2011 - S/INF/66 - (SCOR, 66th year).)
- United Nations. (2022, Dec). *Does UN Peacekeeping Work? Here's the data.* Retrieved from <https://www.un.org/en/video/does-un-peacekeeping-work-here%E2%80%99s-data>
- United Nations. (2023). *The Sustainable Development Goals Report*. Retrieved from <https://sdgs.un.org/goals/goal17>
- United Nations. (2024, Nov). *Resolution 2760* (2024). United Nations. Retrieved from

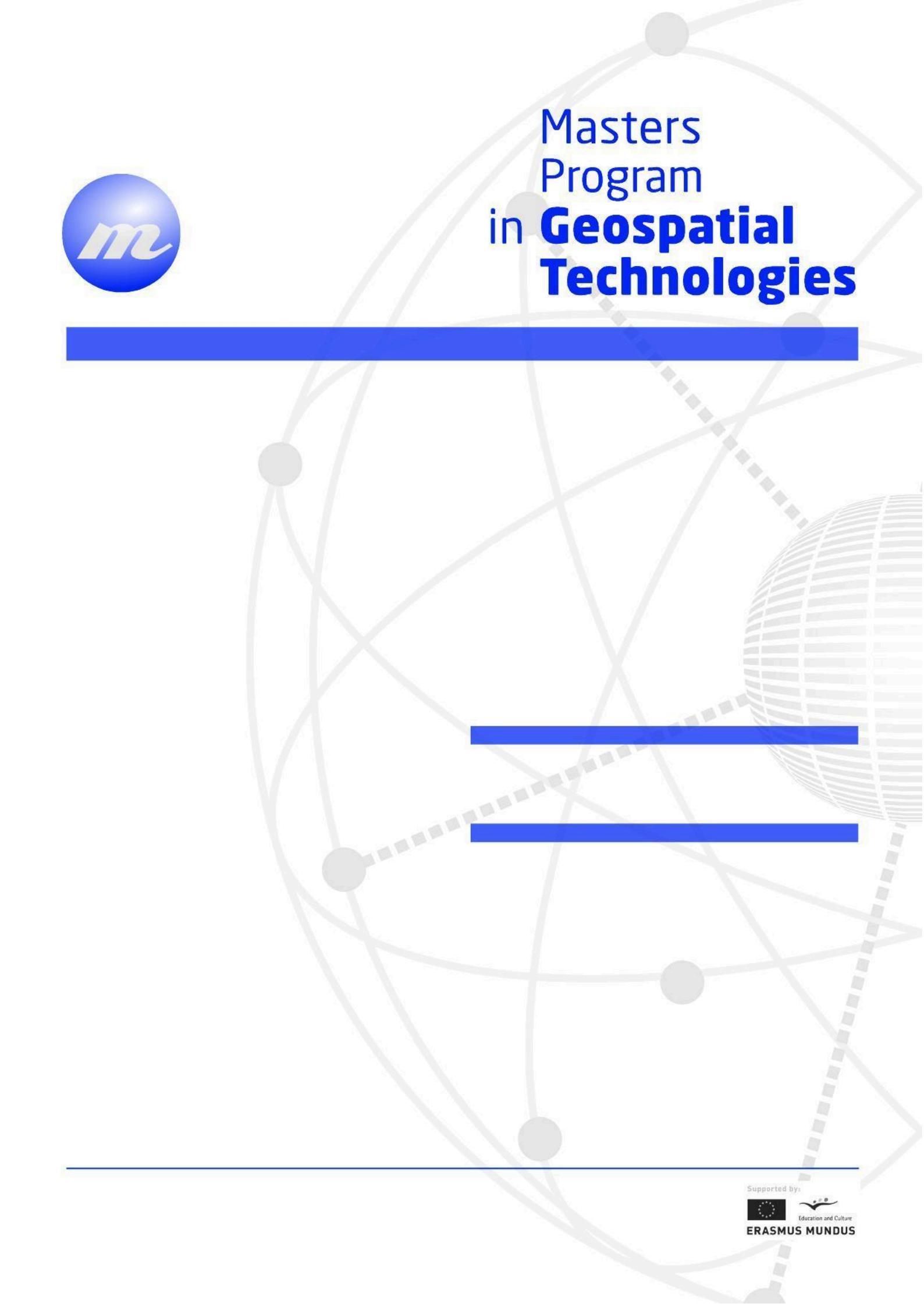
- <http://digitallibrary.un.org/record/4066426> (Issued in: Resolutions and decisions of the Security Council, 2024 - S/INF/79 - (SCOR, 79th year).)
- United Nations Global Service Centre. (2023, Nov 28). *Regional Partnerships for the UN's digital transformation: The advanced FRIM capacity building workshop, Regional Service Centre Entebbe (RSCE), 13-17 November.* Retrieved from <https://www.ungsc.org/news/regional-partnerships-uns-digital-transformation-advanced-frim-capacity-building-workshop>
- United Nations Global Service Centre. (2024, May 10). *From UN Smart Camps towards a Smart Mission: UNISFA marks a new milestone with the inauguration of "Highway Smart Camp".* Retrieved from <https://www.ungsc.org/news/un-smart-camps-towards-smart-mission-unisfa-marks-new-milestone-inauguration-highway-smart>
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. Retrieved from <https://ggplot2.tidyverse.org>
- Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). dplyr: A Grammar of Data Manipulation [Computer software manual]. Retrieved from <https://CRAN.R-project.org/package=dplyr> (R package version 1.1.4)
- Wikle, C., & Zammit-Mangion, A. (2022). Statistical Deep Learning for Spatial and Spatio-Temporal Data. *arXiv preprint: arXiv:2206.02218v1*, 1-27. Retrieved from <https://arxiv.org/abs/2206.02218>
- Wikle, C., Zammit-Mangion, A., & Cressie, N. (2019). *Spatio-Temporal Statistics with R* (1st ed.). S. Chapman and Hall/CRC. Retrieved from <https://spacetimewithr.org/> doi: 10.1201/9781351769723
- Zeng, A., Ho, H., & Yu, Y. (2020). Prediction of building electricity usage using Gaussian Process Regression. *Journal of Building Engineering*, 28, 101054. Retrieved from <https://www.sciencedirect.com/science/article/pii/S235271021930662X> doi: 10.1016/j.jobe.2019.101054
- Ziheng Sun, L. D., & Fang, H. (2019). Using long short-term memory recurrent neural network in land cover classification on Landsat and Cropland data layer time series. *International Journal of Remote Sensing*, 40(2), 593–614. Retrieved from <https://doi.org/10.1080/01431161.2018.1516313> doi: 10.1080/01431161.2018.1516313

2025

*Energy Consumption Modelling in IoT-surveyed Peacekeeping Missions*

Rodrigo Malagón Rodríguez





# Masters Program in **Geospatial Technologies**

---

