REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical-Electronic Engineering

# DEVELOPMENT OF REGRESSION MODEL FOR ELECTRICAL LOAD FORECASTING DURING THE COVID-19 PANDEMIC LOCKDOWN PERIOD

Master Thesis

**Saif Mohammed Salman Al-azzawi**

Supervisor

**Asst. Prof. Dr. Yusuf Gürcan SAHIN**

**Istanbul – 2023**

# THESIS INTRODUCTION FORM

| | | |
|---|---|---|
| **Name and Surname** | **:** | Saif Mohammed Salman Al-azzawi |
| **Language of the Thesis** | **:** | English |
| **Name of the Thesis** | **:** | Development Of Regression Model For Electrical Load Forecasting During The Covid-19 Pandemic Lockdown Period |
| **Institute** | **:** | Istanbul Gelisim University Institute of Graduate Studies |
| **Department** | **:** | Department of Electrical-Electronic Engineering |
| **Thesis Type** | **:** | Master |
| **Date of the Thesis** | **:** | 21.06.2023 |
| **Page Number** | **:** | 120 |
| **Thesis Supervisors** | **:** | Asst. Prof. Dr. Yusuf Gürcan SAHIN |
| **Index Terms** | **:** | |
| **Turkish Anstract** | : | Yeni koronavirüs (COVID-19) salgını, dünya çapında kamu hizmetleri ve şebeke operatörleri için benzeri görülmemiş zorluklar yarattı. Bu tezde, yük tahmini problemine odaklanıyoruz. Katı sosyal mesafe kısıtlamaları nedeniyle, dünya çapındaki güç tüketimi profilleri hem büyüklük hem de günlük kalıplar açısından değişti. Bu değişiklikler, kısa vadeli yük tahmininde önemli zorluklara neden olmuştur. Algoritmalar tipik olarak hava durumunu, zamanlama bilgilerini ve önceki tüketim seviyelerini girdi değişkenleri olarak kullanır, ancak pandemi sırasında sosyoekonomik davranıştaki büyük ve ani değişiklikleri yakalayamazlar. Bu tezde, tahmin algoritmalarının mevcut yapı taşlarını tamamlamak için ekonomik faaliyetlerin bir ölçüsü olarak bir regresyon modeli sunuyoruz. Böyle bir veri kümesiyle ilgili en büyük zorluk, son pandemi ile yalnızca sınırlı hareketlilik kayıtlarının ilişkili olmasıdır. |
| Anahtar Kelimeler | **:** | Elektrik Güç Sistemi, Güç Tüketimi, Coronavirüs (COVID-19), Elektrik Yükü Tahmini. |
| **Distribution List** | **:** | |

*Signature*

*Saif Mohammed Salman Al-azzawi*

REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical-Electronic Engineering

# DEVELOPMENT OF REGRESSION MODEL FOR ELECTRICAL LOAD FORECASTING DURING THE COVID-19 PANDEMIC LOCKDOWN PERIOD

Master Thesis

**Saif Mohammed Salman Al-azzawi**

Supervisor

**Asst. Prof. Dr. Yusuf Gürcan SAHIN**

**Istanbul – 2023**

## DECLARATION

I hereby declare that in the preparation of this thesis, scientific and ethical rules have been followed, the works of other persons have been referenced in accordance with the scientific norms if used, there is no falsification in the used data, any part of the thesis has not been submitted to this university or any other university as another thesis.

Saif Mohammed Salman Al-azzawi
20/06/2023

**TO ISTANBUL GELISIM UNIVERSITY**
**THE DIRECTORATE OF GRADUATE EDUCATION INSTITUTE**

The thesis study of Saif Mohammed Al-azzawi, titled as Development Of Regression Model For Electrical Load Forecasting During The Covid-19 Pandemic Lockdown Period has been accepted as MASTER THESIS in the department of Electrical-Electronic Engineering by out jury.

| | |
|---|---|
| Director | Asst. Prof. Dr. Ercan AYKUT |
| Member | Asst. Prof. Dr. Yusuf Gürcan SAHIN (Supervisor) |
| Member | Asst. Prof. Dr. Ahmed Amin Ahmed SOLYMAN |

APPROVAL

I approve that the signatures above signatures belong to the aforementioned faculty members.

*Prof. Dr. Izzet Gumus*
Director of the Institute

# SUMMARY

The coronavirus (COVID-19) outbreak has presented unprecedented challenges to utilities and grid administrators worldwide. One area of particular concern is load forecasting, as the pandemic has caused significant changes in the magnitude and daily power consumption patterns due to strict social distancing measures. These changes have introduced complexities in accurately predicting short-term electricity demand. Traditionally, load forecasting algorithms rely on weather conditions, timing information, and historical consumption levels to make predictions. However, these algorithms struggle to capture the large and abrupt shifts in socioeconomic behavior during the pandemic. This limitation is due to the lack of available data on mobility, a crucial factor in understanding economic activity during this period. To address this challenge, this dissertation proposes introducing a regression model as an additional measure of economic activity. By incorporating this model into existing forecasting algorithms, we aim to enhance their accuracy and robustness in capturing the effects of the pandemic on electricity demand. However, one of the main obstacles in utilizing such a dataset is the scarcity of mobility records specifically associated with the recent pandemic. By exploring alternative data sources and developing innovative approaches, this dissertation seeks to overcome the limitations posed by the lack of mobility records. The goal is to improve load management forecasting capabilities during significant socioeconomic shifts, such as the ongoing pandemic. This research aims to contribute to advancing load forecasting methodologies and provide valuable insights for utilities and grid administrators grappling with the challenges posed by the COVID-19 pandemic.

**Key Words:** Electrical Power System, Power Consumption, Coronavirus (COVID-19), Electrical Load Forecasting.

# ÖZET

Yeni koronavirüs (COVID-19) salgını, dünya çapında kamu hizmetleri ve şebeke operatörleri için benzeri görülmemiş zorluklar yarattı. Bu tezde, yük tahmini problemine odaklanıyoruz. Katı sosyal mesafe kısıtlamaları nedeniyle, dünya çapındaki güç tüketimi profilleri hem büyüklük hem de günlük kalıplar açısından değişti. Bu değişiklikler, kısa vadeli yük tahmininde önemli zorluklara neden olmuştur. Algoritmalar tipik olarak hava durumunu, zamanlama bilgilerini ve önceki tüketim seviyelerini girdi değişkenleri olarak kullanır, ancak pandemi sırasında sosyoekonomik davranıştaki büyük ve ani değişiklikleri yakalayamazlar. Bu tezde, tahmin algoritmalarının mevcut yapı taşlarını tamamlamak için ekonomik faaliyetlerin bir ölçüsü olarak bir regresyon modeli sunuyoruz. Böyle bir veri kümesiyle ilgili en büyük zorluk, son pandemi ile yalnızca sınırlı hareketlilik kayıtlarının ilişkili olmasıdır.

**Anahtar kelimeler:** Elektrik Güç Sistemi, Güç Tüketimi, Coronavirüs (COVID-19), Elektrik Yükü Tahmini.

# TABLE OF CONTENTS

**SUMMARY** ........................................................................................................ **i**
**ÖZET**................................................................................................................ **ii**
**TABLE OF CONTENTS**........................................................................................ **iii**
**ABBREVIATIONS** ............................................................................................ **v**
**LIST OF TABLES** .............................................................................................. **vi**
**LIST OF FIGURES** ............................................................................................ **vii**
**PREFACE**.......................................................................................................... **viii**

## CHAPTER ONE
## INTRODUCTION

1.1. Introduction.............................................................................................. 1
1.2. Problem Formulation And Definition................................................................ 3
1.3. Objectives And Aims ......................................................................................... 3
1.4. Thesis Questions ............................................................................................... 3
1.5. The Proposed Model.......................................................................................... 4
1.6. Thesis Conclusions ............................................................................................ 4
1.7. Thesis Contents & Layouts................................................................................. 4

## CHAPTER TWO
## LITERATURE REVIEW

2.1. Introduction.......................................................................................................... 5
2.2. COVID-19 Outbreak ............................................................................................ 6
2.3. Global Lockdowns................................................................................................ 8
2.4. Literature Review ............................................................................................... 10

## CHAPTER THREE
## THEORETICAL BACKGROUND

3.1. Forecasting Energy Load.................................................................................... 19
3.2. Time Series Analysis .......................................................................................... 21
3.3. Statistical Approaches ....................................................................................... 23
    3.3.1. Autoregression (AR) model........................................................................ 23
    3.3.1 Moving average (MA) model ........................................................................ 24
    3.3.2. Autoregressive moving average (ARMA) model ....................................... 25
3.4. Linear Models Approaches.................................................................................. 27
3.5. Machine Learning Approaches............................................................................ 29
    3.5.1. Random forest model .................................................................................. 30
    3.5.2. The K-nearest neighbor model..................................................................... 32
    3.5.3. Support Vector Machine .............................................................................. 33

## CHAPTER FOUR
## OPTIMIZATION METHODS OVERVIEW

4.1 The Basic Model of the PSO algorithm ................................................................ 36
    4.1.1 Global Best PSO ( gbest PSO) ...................................................................... 38

4.1.2 Local Best PSO (lbest PSO) ........ 40
4.2 Genetic Algorithms ........ 42
4.2.1 The genetic information ........ 42
4.2.2 Fitness Function Selection ........ 43
4.2.4 Reproduction ........ 45
4.2.5 Crossover ........ 45
4.2.6 Mutation ........ 47

## CHAPTER FIVE
## METHODS AND RESULTS

5.1. Materials: ........ 48
5.2. Methods : ........ 49
I. Time Series Forecasting techniques ........ 50
II. Supervised Machine Learning Forecasting techniques ........ 51
5.3. Random Forest (RF) ........ 53
5.4. Implementation ........ 57
5.5. Results ........ 57

## CHAPTER SIX
## CONCLUSIONS AND FUTURE WORK

6.1 Conclusion: ........ 69
6.2 Future work : ........ 70
**REFERENCE ........ 72**
**APPENDIXES ........ 83**
Python code ........ 83
**RESUME ........ 107**

YÖK watermark

# ABBREVIATIONS

**AR** : Autoregression model

**MA** : Moving average model

**ARMA** : Autoregressive moving average model

**RF** : Random Forest

**WHO** : World Health Organization

**IEA** : International Energy Agency

**GVA** : gross value-added ()

**SVR** : support vector regression

**EWT** : empirical wavelet transform

**VMD** : variational mode decomposition techniques

**EMD** : empirical mode decomposition

**VSTF** : very short-term forecasting

**STF** : short-term forecasting

**MTF** : medium-term forecasting

**LTF** : long-term forecasting

**ACF** : autocorrelation function

**PACF** : partial autocorrelation function

**SVM** : support vector machine

**RF** : Random Forest

**KNN** : K-nearest neighbor

**Sac** : Sacramento

**LA** : Los Angeles

**NY** : New York

# LIST OF TABLES

Table 1: Forecasting horizons........20
Table 2: Search space for the hyper-parameters of regressor models........56
Table 3: The initial hyperparameters values for PSO........58
Table 4: The initial hyperparameters values for GA........58
Table 5: Hyperparameter values after GA tuning........59
Table 6: Hyperparameter values after PSO tuning........59
Table7: Performance results of applying regression models........60
Table 8: Performance results of applying GA optimization........61
Table 9: Performance evaluation of applying the PSO algorithm........61
Table 10: Optimal Hyperparameter values........62
Table 11: Performance evaluation of KNN-PSO and ARIMA models........65

# LIST OF FIGURES

Fig.3.1. ASHRAE’s change point model. ........................................ 28
Figure 4.1: Plot of the functions $f1$ and $f2$. ........................................ 37
Figure 4.2: gbest PSO........................................ 40
Figure 4.3: lbest PSO........................................ 41
Figure 4.5: A basic flowchart of a genetic algorithm. ........................................ 42
Figure 4.6: Example of single-point crossover........................................ 45
Figure 4.7: Example of uniform crossover........................................ 46
Fig. 5.1. Average Hourly electricity usage in three metropolitan areas........................................ 48
Fig. 5.2. Overall prediction methodology........................................ 49
Fig. 5.3. The flow chart for the main process of Hyper-parameter optimization. ........................................ 55
Fig.5.4. Predictions results for the KNN-PSO and ARIMA models........................................ 64
Fig.5.5. KNN-PSO model predictions results........................................ 67

# PREFACE

I thank God for granting me success and helping me to complete my scientific thesis.

*My father, my dear mother, my wife, my brothers, and my friends*. I can never forget your support for me.

I also offer the highest verses of thanks and gratitude to *Dr. Ahmed Amin Ahmed Solyman*, who gave me his precious time. From his information and extensive experience, I ask God Almighty to reward him with the best reward.

Thanks and gratitude to *Dr. Üyesi Yusuf Gürcan Şahin* for accepting to supervise my thesis, which constituted a great addition to the thesis, as his guidance and advice were the beacons I used in my entire research work.

I also extend my sincere thanks for accepting the discussion of the master's thesis to all members of the thesis's jury members who added much information to my thesis with their accurate interventions.

# CHAPTER ONE

# INTRODUCTION

## 1.1. Introduction

Using electricity demand forecasts for cities can assist in planning power generation resources, evaluating energy efficiency programs, monitoring greenhouse gas emissions, analyzing grid infrastructure, and conducting reserve analysis. Understanding buildings' energy usage on a city-wide scale is crucial for promoting global urban sustainability, reducing carbon emissions, and improving energy efficiency (Kontokosta & Tull, 2017). The consumption of electricity in metropolitan areas is influenced by temperature. In addition to factors like population and income, ambient temperature plays a significant role in determining energy consumption on a city level (Deschênes & Greenstone, 2011). This is primarily due to the substantial energy associated with heating and cooling city buildings (Vázquez-Canteli et al., 2019), which heavily relies on outdoor temperature conditions. With more frequent, severe, and prolonged extreme weather events, such as heat waves, due to climate change, analyzing temperature-sensitive electricity consumption at the city scale has become crucial. Consequently, researchers, energy administrators, and policymakers must prioritize climate change adaptation to develop effective solutions (Davoudi et al., 2013).

Furthermore, stakeholders in the energy sector must also prioritize a comprehensive understanding of how electricity generation and transmission infrastructure can be effectively equipped to handle high-demand scenarios, thereby improving energy security and resilience in the face of climate change. In line with this objective, Z. Wang et al. conducted a study examining the effects of rising ambient temperatures on electricity consumption in Shanghai (Z. Wang et al., 2021). Yi-Ling et al. stated that the projected rise in temperatures would increase summer electricity demand while reducing winter electricity demand, assuming no changes occur in the current energy consumption patterns(Yi-Ling et al., 2014). Similarly, Sathaye et al. estimated that by 2099, California would require an additional peak generation capacity of up to 38% and an additional transmission capacity of up to 31% due to atmospheric warming and the associated increase in peak demand (Sathaye et al., 2013). In August 2020, a heat wave in California caused a shortage in power supply due to the increased usage of air conditioning, resulting in intermittent power disruptions for residents. Reflecting on this disruptive event, policymakers highlighted the importance of refining electricity demand forecasts by considering climate change, extreme weather events, and their impact on energy loads as the primary course of action (Lu et al., 2021; Z. Wang et al., 2021).

In addition to weather conditions, unforeseen public health events can significantly impact citywide electricity consumption. Research conducted in Brazil has shown that the COVID-19 pandemic caused a decrease in electricity usage among Brazilians, ranging from 7% to 20%, depending on the local economic structure. These changes affected areas with a predominant industrial sector (Carvalho et al., 2021). Similarly, a European study revealed that the severity and extent of lockdown measures directly influenced electricity consumption patterns. Countries with stringent restrictions, such as Spain and Italy, exhibited electricity consumption profiles during the pandemic that resembled pre-pandemic weekends in 2019.

Conversely, countries with less strict measures, like Sweden, experienced a comparatively smaller decrease in power consumption (Bahmanyar et al., 2020). Monitoring electricity consumption can serve as a real-time indicator of the economic consequences of lockdown measures. For instance, Switzerland witnessed an overall decrease in electricity consumption by 4.6%, while the Canton of Ticino, which implemented more stringent reduction measures in addition to federal regulations, observed a larger decrease of 14.3% (Janzen & Radulescu, 2020).

## 1.2. Problem Formulation And Definition

(1) The literature review reveals that most researchers exclusively evaluated the prediction model's accuracy, ignoring its stability.

(2) Previous research on the electricity demand forecasting ignored significant global events such as COVID-19, indicating that weather and other factors may not be as influential during such times. Consequently, the applicability of these models to significant global events may be limited.

## 1.3. Objectives And Aims

The following is a summary of the thesis' numerous significant contributions:

(1) Introduction of a hybrid model for daily electricity demand forecasting during the COVID-19 pandemic.

(2) Comparison of the proposed model to benchmark models in terms of accuracy and stability of prediction.

(3) Examine the effects of denoising techniques and optimizers on prediction results.

(4) Discuss the viability of incorporating COVID-19-related factors as inputs to the prediction model.

(5) Comparing the prediction outcomes of various forecasting models.

## 1.4. Thesis Questions

In conjunction with the previous review of related literature, problem formulation, and the study objectives, we can summarize the main questions addressed in this thesis as follows:

(1) Which model yields the best performance for Electrical Load Forecasting?

(2) What are the optimal hyperparameters for the prediction models?

(3) What is the correlation among the features utilized in our study?

## 1.5. The Proposed Model

This study utilizes three machine learning models (SVM, KNN, and RF) to forecast electricity consumption in 3 United States metropolitan areas (New York, Sacramento, and Los Angeles). To increase prediction performance, hyperparameter tuning was performed on each machine learning model, resulting in greater accuracy than PSO and GA, two metaheuristic optimization techniques. Each Supervised Machine Learning Forecasting technique's performance (SVM, RF, and KNN) was evaluated using the default and tuned hyperparameters values. In addition, the proposed forecasting model was compared with another forecasting technique, namely, Time series forecasting techniques, including (the ARIMA model).

## 1.6. Thesis Conclusions

This thesis presents an innovative method for load forecasting that addresses the unique challenges posed by the abrupt and global COVID-19 pandemic. The proposed method is rigorously evaluated on global load forecasting assignments comprising diverse regions. In light of the ongoing impact of the pandemic on power grids, the methodologies devised in this study have the potential to provide grid operators with valuable information about future load patterns.

## 1.7. Thesis Contents & Layouts

In the rest of this thesis, Chapter 2 presents the literature review; Chapter 3 discusses the theoretical background; chapter 4 provides an in-depth characterization of the machine learning algorithms utilized in this study, while Chapter 5 delves into the methods employed and presents the corresponding results. Finally, Chapter 6 consolidates the findings and outlines potential future directions for research.

# CHAPTER TWO

# LITERATURE REVIEW

The most pertinent research on our study question is reviewed in this chapter. First, consider how electricity use in 2020 might have served as a leading economic indicator. We then research studies on how changes in electricity use during the pandemic.

## 2.1. Introduction

Global energy demand and consumption are rising consistently due to the combined effects of economic growth and population expansion. The progression of technology and the development of economic conditions contribute to an increase in energy consumption (Y. H. Chen et al., 2016). One factor influencing a country's degree of development is ensuring that energy supply and demand are balanced. To satisfy demand, it is crucial to use energy efficiently (Elattar et al., 2020). For emerging nations to manage their economies effectively, accurate energy forecasting is essential (Matijaš et al., 2013). Globally, the coronavirus epidemic has produced exceptional circumstances. The coronavirus is a virus that causes respiratory illnesses, and it was only recently discovered (Malec et al., 2021).

Covid-19 has been categorized as a pandemic by the World Health Organization (WHO) due to its distinguishing characteristics compared to other coronaviruses. People's living situations have quickly changed due to pandemic measures, affecting energy production and consumption (Özbay & Dalcali, 2021). In response to the pandemic, measures such as curfews and the closure of public spaces were implemented, resulting in increased household electricity consumption and decreased usage in commercial and industrial sectors (Malec et al., 2021). In all the countries where it is found, the virus has negatively affected every aspect of life, including education, health, money, economy, and energy (Elattar et al., 2010). The International Energy Agency (IEA) reports that global electricity consumption experienced a decline of 2.5% during the initial three months of 2020 ( Scarabaggio et al., 2020; Said et al., 2021; Alasali et al., 2021;).

In the beginning, some nations imposed partial quarantine. Under complete quarantine, electricity use was cut by at least 20%, and under partial curfew, to a lesser level. On March 11, 2020, the first coronavirus case in Turkey was noted (Özbay & Dalcali, 2021). As a result, limitations were imposed nationwide. As a result, there have been significant changes in energy consumption. In comparison to 2019, subscriber-based electricity billing increased by 1.67% in 2020 (Sonmez & Bagriyanik, 2021). Large businesses such as hotels, retail centers, and the Istanbul Airport faced reduced operational capacities, resulting in an 11% decline in commercial electricity consumption. Conversely, residential electricity consumption witnessed a 6.6% increase. Industrial consumption initially decreased due to a decline in industrial production during the year's second quarter. However, with industrial production recovering during the third quarter, industrial consumption saw an overall increase of 5.62% throughout the year  (Özbay & Dalcali, 2021; Sonmez & Bagriyanik, 2021).

## 2.2. COVID-19 Outbreak

In late December 2019 and early January 2020, as we contemplated New Year resolutions, a highly transmissible virus emerged in a distant part of the world, catching our immune system completely off guard (Kucukali & Baris, 2010). At the time, we never would have imagined that a faraway virus could have spread and resulted in many issues for people's health and economic systems, individually and collectively(Kong et al., 2017). However, the world situation has drastically changed in only two months, and we have had to adjust and fulfill the new needs. In November 2019, the novel coronavirus Sars-CoV-2 began its transmission in China, specifically in Wuhan, a bustling center for trade and commerce. Initially, the true nature of the virus remained unknown, as a few cases of unusual pneumonia with unidentified causes started to surface. It was not until December 31, when local health authorities officially announced these distinct cases, that the narrative of the novel coronavirus began to unfold. The city had discovered scores of cases at the start of January

2020, and hundreds of people were being watched. During the initial investigations, it was revealed that the individuals who had contracted the infection had frequent contact with the Huanan Seafood Wholesale Market in Wuhan, which has been closed since January 1, 2020. This led to the hypothesis that the infection might have originated from an animal-based product sold in the market. The responsible pathogen was identified as a new strain of coronavirus, belonging to the same family as the coronaviruses that caused SARS and MERS. However, it was noted by Chinese authorities on January 9 that this new strain was distinct and more potent than its counterparts. The World Health Organization released this information on January 10, providing relevant advice (such as avoiding contact with symptomatic individuals) and stating that there were no recommended travel restrictions to or from China. At that time, the cases were localized primarily in Wuhan, and it remained uncertain whether the virus was highly contagious (unlike MERS and SARS, which were more severe but less transmissible) (Deif, Hammam, Ahmed, Mehrdad  Kamarposhti et al., 2021).

As a result of Wuhan's isolation, Chinese New Year celebrations were postponed not only in Wuhan but also in other cities like Beijing and Macau. In late January, the risk of the epidemic spreading transitioned from a low level to a high level, prompting the World Health Organization (WHO) to declare on January 27th that the danger was "extremely high for China" and elevated on a regional and global scale. Three days later, in response to the substantial risk, the WHO declared a "public health emergency of international concern," resulting in Italy suspending the only flight that connected Europe and China. However, positive developments were already emerging in China. By February 8, the WHO reported that infections were stabilizing or experiencing a steady decline in the daily number of new cases. While COVID-19 had not yet been classified as a pandemic by the WHO, the number of infected individuals outside of China, particularly in Italy, Iran, and South Korea, was significantly high in February.

Nonetheless, as February transitioned into early March 2020, more cases were detected in countries across Europe and beyond. Italy emerged as a prominent player in combating the virus, with support from the WHO. On March 11, 2020, during a briefing on the coronavirus epidemic, Tedros Adhanom Ghebreyesus, the director-general of the WHO, officially declared COVID-19 as a pandemic, urging all governments to take decisive measures to contain its spread.

### 2.3. Global Lockdowns

In response to the COVID-19 pandemic, numerous countries and territories have introduced curfews, quarantines, and similar measures. Various non-pharmaceutical interventions were implemented to curb the spread of SARS-CoV-2, the virus responsible for COVID-19. By April 2020, governments across over 90 countries or territories had issued advisories or mandated stay-at-home orders, affecting over 3.9 billion people. Approximately half of the world's population was placed under lockdown. Lockdowns have been enacted to varied degrees by countries around the world. Some have complete mobility control, while others have imposed time-based constraints. In most cases, only necessary enterprises were allowed to continue operating(Y. H. Chen et al., 2016).

We all use the phrase "lockdown," but until 2020, it was solely used to describe convicts and their cells. Due to the circumstances, its meaning has changed and is now understood(Ucar & Korkmaz, 2020):

- Directives to stay at home and enforce movement restrictions
- Closure of kindergartens and schools
- The shutdown of non-essential businesses and Non-essential production are shut off
- leisure centers are closed, and public spaces are closed
- Social movement limits and social separation measures
- Limits on travel.
- Additional non-pharmaceutical anti-pandemic strategies include obligatory post-travel quarantines, self-quarantines, and social seclusion.

At the end of January, China became the first nation to impose a quarantine and place cities under lockdown, followed by entire provinces. On January 23, 2020, China issued a mandate to impose strict isolation measures on Wuhan and other cities within the Hubei province to stop the sickness from spreading. This marked the first instance in modern history where a densely populated metropolis of 11 million inhabitants was completely cut off. Following Wuhan's example, several other cities swiftly implemented similar policies. Within hours, quarantine measures and movement restrictions were enforced in nearby cities such as Huanggang and Ezhou, extending to 15 additional cities. As a result, approximately 57 million people were affected by these measures. Italy was the first nation in Europe to enter a state of general lockdown in the interim: "There will not be a red zone, a zone one, or a zone two; instead, there will be an Italy-protected region. Except in three circumstances, proven work-related concerns, cases of necessity, and health-related issues movements will be avoided. Giuseppe Conte, the prime minister, used this phrase to herald the signing of a historic decree on March 9, 2020. The COVID-19 epidemic would have put the entire country of Italy into lockdown starting the next morning. Following the Second World War, curfews were initially implemented, prohibiting marriage and funeral celebrations. Theatrical, film, and swimming facilities were shuttered.

All sporting activities, such as licensure tests, museums, cultural centers, wellness centers, discos, and ski resorts, are postponed. Straightforwardly, Conte's declaration on March 9 served as the initial introduction of the Coronavirus pandemic to the consciousness of Italians, initiating a "new reality" that we still encounter in our daily lives today. These two months were exceptionally peculiar. Starting from March 10 until the first glimmer of normalcy emerged two months later on May 18, when businesses resumed operations, we endured two challenging months that left a lasting impact on every aspect of life.

By the end of March, almost all European Union member states, starting with Spain and France and spreading northwards, had implemented quarantine measures, following the example set by Italy (except for Sweden, which maintained a more open approach during the pandemic). As of March 26, 2020, around 1.7 billion individuals were under some form of confinement. By the first week of April, over half of the global population, approximately 3.9 billion people, were also adhering to the directives issued by the Indian government.

In the United States, a unique situation arose when President Trump delegated the authority to implement curfews to individual states. Due to the wide-ranging powers granted to states, restrictions were not uniformly enforced nationwide. As of April 6, five states (North and South Dakota, Iowa, Nebraska, and Arkansas) had not implemented significant measures to combat the pandemic. Approximately nine out of ten Americans were subject to varying degrees of restrictions in four states (Oklahoma, Wyoming, Utah, and South Carolina). The remaining 41 states encouraged everyone to stay indoors.

The United States quickly surpassed other nations regarding COVID-19 cases and fatalities, with over 30 million confirmed cases and 550,500 reported deaths. (Özbay & Dalcali, 2021).

## 2.4. Literature Review

(Ferguson et al., 2000) This study raises doubts about the relationship between total energy consumption and economic activity. The researchers analyzed correlations between GDP and electricity consumption in over 100 countries and between total primary energy supply and GDP. The data used in this analysis were adjusted for purchasing power parity and presented on a per-person basis. The study focuses on time series data from 1960 (or 1971 in certain cases) to 1995. According to the research findings, most developed nations exhibit correlation coefficients of at least 0.9 between GDP and electricity consumption, except major oil producers or refiners.

Furthermore, the study reveals that this association becomes stronger as a nation's wealth increases, indicating that as the economy expands, people tend to utilize more electricity. As this report was published in 2000 and the analysis ended in 1995, the empirical findings may be now dated. However, the dynamics of the differences in correlation across the nations are important to note. It is also clear that the countries we focus on in our thesis have had divergent development since 1995. Moreover, it makes no mention of the long-/short-run dynamics.

S.-T. Chen et al. (2007) conducted an extensive analysis of the connection between GDP and energy consumption in ten Asian nations, building upon the earlier research by Ferguson et al. (2000). This study goes beyond the previous work by exploring the long- and short-term dynamics of the relationship between electricity consumption and GDP growth in greater depth. By examining ten Asian countries, namely China, Hong Kong, India, Indonesia, Korea, Malaysia, the Philippines, Singapore, Taiwan, and Thailand, the authors provide additional evidence that supports the findings of Ferguson et al. (2000) regarding the existence of a long-term association between electricity consumption and GDP. To investigate this association, the researchers conduct cointegration tests. Additionally, they employ Granger causality analysis to determine whether there is a statistical indication of one variable causing the other or vice versa. The panel tests reveal a long-term bi-directional Granger causality and a short-term unidirectional Granger causality from economic growth to energy consumption, although the specific results vary across different countries in the study.

The correlation over the long term for the USA is examined in (Hirsh & Koomey, 2015)2015 published research. They found that the relationship between GDP and electricity has steadily deteriorated since the middle of the 1990s. In their conclusion, the researchers acknowledge that they did not consider yearly fixed effects, such as technological advancements leading to a decrease in energy intensity. Their analysis did not consider this factor, which may be the main driver behind their observed results. The fact that there is still a correlation, albeit declining, supports the idea that electricity was still an economic indicator in 2015. So it might be important to consider power usage's long-term, far-off importance as an economic indicator.

The study (Hirsh & Koomey, 2015)examined the relationship between electricity and GDP growth rates in the United States. However, they did not explore potential adjustments that could be made to control for certain factors. Building upon this research, (Arora & Lieskovsky, 2016) carried out a subsequent study that picked up where (Hirsh & Koomey, 2015) left off.

The findings of Arora & Lieskovsky (2016) reveal a baseline correlation of 76 percent between electricity and GDP growth rates from the mid-1970s until 2013. However, after controlling for seasonality and reducing energy intensity, the correlation increases to 86 percent for the entire period. Therefore, it is reasonable to infer that the long-term connection between electricity and economic activity still holds, even with these adjustments.

Furthermore, the data series analyzed by Arora & Lieskovsky (2016) demonstrates that electricity consumption and GDP growth rates move in tandem during recessions. Additionally, the growth rates in power consumption tend to increase before the end of economic downturns.

While previous studies have consistently identified a strong long-term association between electricity consumption and economic activity, (S.-T. Chen et al., 2007) deviate from this trend by suggesting a short-run relationship between economic growth to electricity consumption. It is worth considering that previous research may have overlooked the potential significance of electricity as a short-term indicator in capturing fluctuations in economic activity.

In 2020, (Cicala, 2020a) estimated the sole impact of Covid-19 on electricity usage. To accomplish this, he utilized a regression analysis approach, correlating power consumption with several established measures of electricity usage. By employing consumption data from most of the European Union (EU), he demonstrates a notable decrease in energy consumption across all countries during the early stages of the pandemic (up until April 6/2020, the last data point considered).

Cicala's projections align closely with the timelines of lockdown implementations, providing evidence of consumption declines that precisely correspond to these measures. Additionally, his analysis suggests that the European economy experienced a historic low during this period. The overall decline in energy consumption across the EU was estimated at 10%. Furthermore, Cicala raises a pertinent question regarding the alignment between consumption data and economic statistics, highlighting the need to examine how closely these two metrics correspond.

(Leach et al., 2020) contribute to the existing knowledge by examining the changes in electrical markets during the Covid-19 crisis. Their study focuses specifically on the events related to the pandemic within Canada, analyzing the data at a regional level. While they also explore supply-side adjustments, we will only provide further details on this aspect if it is relevant to the topic of this essay. When examining four distinct regions, they can see changes in the pandemic's severity, the timing of declines relative to the events, and the number of shocks due to the regions' mixed economies. The report asks whether electricity is a good real-time measure of economic activity, but it stops there in terms of research for the time being. They also draw attention to the amount of information that may be gathered from electrical data, which is one of the potential advantages beyond the temporal one. This data analysis showcases the potential for increased data utilization as it enables the differentiation of various consumer classes, including commercial, industrial, residential, and distinct industrial sectors, based on their consumption patterns. It will provide decision-makers with a more thorough picture of a comparable circumstance than other comparable proxies can.

The publications mentioned above serve as a clear and illustrative example of how changes in the power market can be effectively utilized to monitor shifts in demand during the 2020 pandemic. The data can be used to estimate when the slump began and how well the recovery may progress. Still, they do not provide additional information about how this might monitor economic activity.

In contrast to the majority of existing literature that focuses on a single country's analysis of electricity as an economic indicator during the Covid-19 pandemic, (Fezzi & Fanghella, 2021) aims to present a general technique applicable to multiple countries. They draw inspiration from studies by (Beyer et al., 2021; Janzen & Radulescu, 2020; Menezes et al., 2021) propose a simplified approach with significant findings.

Their study examines twelve nations: Belgium, Switzerland, Austria, France, Germany, Italy, Norway, The Netherlands, Sweden, Spain, Denmark, and The United Kingdom. Utilizing comparable "prefiltering" techniques, similar to those presented in (Cicala, 2020b)and (Leach et al., 2020), with minor modifications, they estimate the counterfactual "normal" power consumption values for 2020. Assuming that all demand outside the residential sector fell, they analyze the recession's impact on the economy while considering the proportion of residential load in each country's total load. This approach allows them to estimate real-time changes in GDP, exhibiting a correlation coefficient of 0.98 with actual data for the first and second quarters of 2020.

Due to data availability until the end of August 2020, the study focuses solely on the first two quarters, referred to as "the first wave of the pandemic." (Fezzi & Bunn, 2010), evaluate the selected nations based on their Non-Pharmaceutical Interventions (NPIs) and their impact on GDP and power consumption to determine the most effective and least effective "measure strategies."

(Beyer et al., 2021)add to the collection of studies looking at the effects of Covid-19 using electricity data and data on the intensity of evening light. However, since they are implementing it in India, it provides information about the technique's applicability in a less developed nation than the USA and Europe. In contrast to previous approaches that primarily utilize GDP as an economic indicator, their method distinguishes itself by employing gross value-added (GVA) statistics. When analyzing a sample of 123 countries, they observe a consistent long-term correlation of 0.95 between these two variables, aligning with previous research findings (Ferguson et al., 2000).

Furthermore, the coefficient obtained from their regression analysis of GVA on energy consumption is similar to the findings of other European studies. The previous examples are followed for modeling power consumption, with a few small variations to account for the geography of their subject, India. They also look at regional variations and national data to show the variety within India's economy.

The study conducted by (Menezes et al., 2021) offers valuable insights into the utilization of power data as an economic indicator in Brazil. Similar to previous articles in this field, they employ a similar methodology to determine the baseline or "normal" electricity consumption. In addition to conventional quarterly GDP statistics, they consider a monthly indicator called IBC-Br, provided by the Brazilian Central Bank. This expanded approach allows for a comprehensive analysis of the relationship between Brazil's power consumption and economic activity. Their study affirms the use of power data for developing nations like Brazil and developed nations in the EU. Their findings are pretty solid, given that the correlation between the movement of the real GDP and their indicator is roughly 0.98 between February 2020 and May 2020. Including the IBC-Br proxy is particularly significant because it encompasses all forms of consumption, including formal and informal activities. This aspect is crucial for a country like Brazil, where the informal sector contributes to nearly 40% of the total economic activity. By considering the comprehensive nature of the indicator, the study recognizes the importance of capturing the entire economic landscape and providing a more accurate representation of the relationship between power consumption and economic activity in Brazil. The study also discusses the distinctions between the residential, industrial, and commercial customer segments.

The study (Janzen & Radulescu, 2020) focuses on documenting the electricity consumption patterns during the designated lockdown period in Switzerland. Unlike previous studies, they adopt a unique approach by analyzing the hourly load data, specifically during the seven weeks leading up to the lockdown and the five weeks following its initiation. The authors introduce dummy variables to capture the changes in electricity consumption attributed to the Covid-19 situation. These dummy variables are incorporated for each week, seven weeks prior and five weeks after the commencement of the lockdown. The regression analysis, which includes the logarithm of load and factors such as temperature and temporal dummies, helps estimate the coefficients associated with these dummies.

Furthermore, the study examines the relationship between these coefficient values and markers indicating the severity of the pandemic, such as the number of cases per capita and movement data. Additionally, (Janzen & Radulescu, 2020) delve into regional variations within Switzerland by considering different cantons as distinct political regions. It is important to note that the study assumes that economic output accounts for 67 percent of total electricity consumption, but no direct comparison is made with actual economic data.

The four publications mentioned above share a common approach of using power data as an economic indicator to assess the impact of the pandemic on consumption. However, there is a distinction in the methodology employed by (Janzen & Radulescu, 2020), as they specifically isolate the time-fixed impacts during the lockdown weeks. Nevertheless, these studies highlight the significance of electricity usage data in capturing the market shock caused by Covid-19 and its clear implications as a tool for identifying impacts during economic crises. The evidence presented in these studies consistently supports the reliability of using empirical data to gauge the effects, regardless of the specific economic conditions, particularly in the near term.

(Kaynar et al., 2017)  Utilized a hybrid algorithm that combined the Support Vector Regression (SVR) algorithm with chaotic approaches to conducting a forecasting investigation. Their study aimed to predict future load data based on historical data from 2006 to 2009, as analyzed by (Baghel et al., 2016). Similarly, (Türkay & Demren, 2011) conducted a demand forecasting study using the Library for Support Vector Machines (LibSVM) algorithm. They utilized load data to develop a predictive model. In a different approach, (Elattar et al., 2010) conducted a forecasting investigation using a hybrid model that combined the Least Square Support Vector Regression (LWSVR) algorithm. Their study aimed to forecast future outcomes based on historical data. Furthermore, (Al Mamun & Kermanshahi, 2006) conducted a prediction study comparing the outcomes of using Support Vector Machines (SVM) and Artificial Neural Network (ANN) approaches. Their study focused on analyzing the predictive capabilities of these two algorithms.

In their study, (S. Zheng et al., 2017) employed data from smart meters connected to a smart grid to assess the effectiveness and accuracy of three forecasting methods: Seasonal Autoregressive Integrated Moving Average (SARIMA), Nonlinear Autoregressive Neural Network (NARX), and Support Vector Regression (SVR). The objective was to evaluate these methods' performance in predicting electricity consumption using the data obtained from the smart meters.

The performance of the CNN model was compared to other models, including decision tree, Random Forest (RF), Support Vector Machine (SVM), Short-Term Long Memory (LSTM), and multiple linear regression. The results demonstrated that the CNN model outperformed the other models regarding predictive accuracy. The CNN model exhibited high prediction rates and required less training memory, making it a preferred choice in Short-Term Load Forecasting (STLF) investigations.

Wang et al. developed three models: Long Short-Term Memory-CNN (LSTM-CNN), CNN, and hybrid CNN-LSTM. Among these models, the hybrid CNN-LSTM model demonstrated superior performance and yielded the best results (J. Wang et al., 2010). Models based on residual and dense networks were employed by (H. Zheng et al., 2017). In addition to the models mentioned, the input data underwent preprocessing using empirical wavelet transform (EWT) and variational mode decomposition (VMD) techniques (ALTAN & KARASU, 2020). It was found to produce superior outcomes compared to other conventional techniques. The study demonstrated the superiority of a hybrid forecasting model that combined a deep neural network with empirical mode decomposition (EMD) over more traditional approaches.

Empirical Mode Decomposition (EMD) helps mitigate issues related to slow convergence and local minimum problems in artificial neural networks by decomposing the data into intrinsic mode functions (Bessec & Fouquau, 2008; Huang & Kunoth, 2013; H. Zheng et al., 2017) EMD, unlike wavelet decomposition, do not rely on pre-defined basis functions but rather adapts to the local characteristics of a signal sequence. This makes the EMD method more flexible and easily adaptable (Bessec & Fouquau, 2008).

EMD operates at multiple resolutions, allowing for the decomposition of a signal into different components at different scales. In contrast, selecting a wavelet-based function for wavelet transform involves choosing an appropriate wavelet-basis function that suits the specific characteristics of the signal (Acikgoz, 2022; Acikgoz et al., 2021).

# CHAPTER THREE

# THEORETICAL BACKGROUND

The theoretical underpinnings of forecasting are presented in this chapter in several areas. There is a discussion on time series analysis's fundamental definitions. Based on a literature review, this chapter also gives the history of various mathematical formulations of the chosen methodologies. A comparative analysis of these statistical, traditional, and deep learning approaches is briefly offered.

## 3.1. Forecasting Energy Load

Foreseeing the short-, medium-, and long-term demand for energy is known as energy data forecasting. It entails developing informed predictions about how much energy will be needed by residences, companies, and other entities. The large quantities of data about energy use, including specific measurements of energy use like electricity demand, PV generation, wind, gas, steam, and heating load, are referred to as energy data. Energy is generated from various sources and is measured using different time scales and units of measurement. Historical energy data is used to train forecasting algorithms and establish time series for future periods. Accurate energy data forecasting relies on the availability of relevant information. To evaluate the efficacy of forecasting across various energy data applications, this study focuses on using datasets linked to PV generation and electrical demand in residential usage. Based on the duration of the forecasting interval, the forecasting horizons in the energy sector can be broadly divided into four groups: very short-term forecasting (VSTF), short-term forecasting (STF), medium-term forecasting (MTF), and long-term forecasting (LTF). The forecasting of energy data for both business and residential use has been studied using a variety of methodologies and models in the literature (Kuo & Huang, 2018). VSTF, which stands for Very Short-Term Forecasting, specifically focuses on predicting energy data within a period of up to an hour in advance.

It aims to provide accurate and timely forecasts for immediate or near-future energy demands. VSTF techniques and models are designed to handle short-term energy consumption and generation variations, enabling better planning and decision-making in real-time energy management. This type of forecasting is particularly valuable for optimizing energy distribution, load balancing, and ensuring grid stability (Taylor, 2008).

Demand sight's management system frequently uses STF, which takes a range of time into account up to a day in advance(Kaytez et al., 2015). VSTF and STF are primarily utilized for the daily operation and scheduling of electricity and spot price calculation. These applications require a significantly higher accuracy level than long-term forecasts (Kuo & Huang, 2018). This type of forecasting is essential to ensure that the scarce electricity in developing nations is used more effectively (Hong et al., 2010). The term MTF, referring to "month ahead," has long been employed (Ghiassi et al., 2006)for scheduling maintenance and grid system development.

In contrast, LTF encompasses a timeframe ranging from months to years ahead, focusing on power supply arrangement and resource planning. When it is important to predict power demand over a longer time, this form of forecasting is used to establish system design (Kuo & Huang, 2018). MTF and LTF are often only used to estimate peak loads since they frequently experience forecast mistakes over time (B.-J. Chen et al., 2004).

Table 1: Forecasting horizons

| The duration of forecasting intervals | Time scale |
|---|---|
| **Long-term** | months-year |
| **Medium-term** | 24h- weeks |
| **Short-term** | One h- 24h |
| **Very short-term** | 5min-1h |

Xia and Wang's study revealed that factors such as temperature, humidity, and wind speed significantly influenced the accuracy of short-term forecasting (Xia et al., 2010). As a result, various variables, including historical load data, meteorological conditions (humidity, temperature), seasonality, day of the week, time of day, and even specific holidays or festivals, can impact the forecasting process.

## 3.2. Time Series Analysis

A time series refers to a collection of observations recorded over a specific duration, whether a short or long period. This dataset typically comprises numerical values and corresponding time stamps, which are recorded at regular intervals (Hyndman & Athanasopoulos, 2018). Time series analyses examine and glean information from a dataset gathered through time. The time interval between each observation in a time series is determined based on a predetermined frequency, which can be set at various intervals such as hourly, daily, or weekly. This frequency dictates the regularity at which data points are recorded and allows for consistent measurement and analysis of the time series. The decomposition of the time series is one method for analyzing these structures. Four parts can be separated from the time series. The terms "trend," "cyclical," "seasonal," and "irregular" are used to describe these elements.

- **Trend:** A long-term change in the variability of a time series, characterized by either an increase or decrease. "changing direction" can describe a trend that transitions upward to downward.
- **Seasonal:** Seasonal elements, which are variations in pattern over the year's seasons, can impact time series data. The impact of these seasonal elements causes periodic fluctuations or seasonal patterns in the time series (Deif, Solyman, & Hammam, 2021). Examples of important factors producing this seasonal fluctuation include weather and customs.
- **Cyclical:** This term describes how variations or patterns might recur throughout a time series. It denotes the existence of cyclically repeated, non-

periodic fluctuations. These cycles typically last over two years (Lee & Ko, 2011). The cyclic and trend components are combined and called the overall trend.

The random component within the time series is called the irregular component or residual. The element that depicts the random variations at each instant is also known as the noise component. A flawless forecast is not feasible since time series contain these erratic components. There are two models for time series decomposition: additive decomposition and multiplicative decomposition. The existence or absence of a trend in the dataset determines whether a time series is stationary or non-stationary. The mean and covariance of the observations are constant across time in a stationary time series.

On the other hand, the dataset is regarded as non-stationary if it exhibits an upward or downward trend. The time series is regarded as stagnant when the dataset shows no trends. Non-stationary time series must have the proper treatment before modeling.

Cleaning the raw data to remove anomalies and translating it to a new scale are the first steps in processing time series data. Differentiating, which calculates the differences between successive observations to create a new time series, is one method for processing time series data. The autocorrelation function (ACF) can detect non-stationarity in time series. To put it simply, correlation looks at the linear relationship between two variables. Statistical methods for finding patterns in time series include the partial autocorrelation function (PACF) and the autocorrelation function (ACF). Various statistical software programs enable the visual representation of ACF and PACF through correlograms. These correlograms can also assist in determining the parameter range for autoregressive (AR) models using the PACF plot and moving average (MA) models using the ACF plot.

## 3.3. Statistical Approaches

Linear statistical techniques, including the autoregressive (AR) model, moving average (MA) model, and their derived models, such as ARMA (autoregressive moving average) and ARIMA (autoregressive integrated moving average), have long been influential in the field of forecasting. These models have gained significant popularity among forecasting academics over the years. These models are referred to as Box-Jenkins models at times. Exponential smoothing techniques, such as the Holt-Winters method and simple exponential smoothing, are another statistical category. The explicit modeling of error, trend, and seasonality is often called the ETS model. This part will discuss these forecasting techniques from Hyndman and Athanasopoulos's book Forecasting: Principle and Practice(Brockwell & Davis, 2002).

### 3.3.1. Autoregression (AR) model

By regressing a value against earlier values from the same series, an autoregression model forecasts a value in a time series. When a single measurement is taken at regular intervals, univariate time series without a trend or seasonality is especially well suited for this modeling technique. The autoregression model uses a linear combination of the current observations and the anticipated variable based on the past p observations of that variable and takes uncertainty into account (Hamilton, 2020). The anticipated output variable yt in the autoregressive process depends linearly on its initial values, such as (yt1, yt2,... ytp), and the white noise t. The white noise comprises a set of uniformly distributed, uncorrelated random variables with a limited variance of 2. It is represented as WN(0, 2). (Brockwell & Davis, 2002).

The following equation describes how the $AR(p)$ model can be expressed:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \qquad (3.1)$$

In the given equation, $\varepsilon_t$ represents white noise with a distribution of $WN(0, \sigma^2)$, $C$ and $\varphi$ are constant parameters of the model, and $\varphi$ takes values from 1 to $p$. The model uses the order p to determine the number of previous observations needed to forecast the present value. The *AR(1)* model, which is a first-order autoregression model, serves as a basic example and can be expressed as follows:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t \tag{3.2}$$

The autocorrelation coefficients gradually decrease when dealing with an autoregressive (AR) process. This makes it difficult to determine the order of the model using the autocorrelation function (ACF). However, the partial autocorrelation function (PACF) plot offers a potential solution, displaying a sharp cut-off after the *lag p* (Shumway et al., 2000). Since static data is the norm for autoregressive models, some parameter value limitations are necessary (Shumway et al., 2000).

### 3.3.1 Moving average (MA) model

Single-variable time series are typically expressed using moving average models. A stationary time series can be described as a process that incorporates prior prediction errors in a regression-like model rather than relying solely on historical values of the forecast variable. This approach, known as a moving average process, helps maintain the stationarity of the time series. To put it another way, the component in the $MA(q)$ process depicts a model's error series as a linear mix of the current observation and earlier $q$ innovations (Hamilton, 2020). A moving average model of order $q$, known as the $MA(q)$, can be represented in the following manner (Li et al., 2020):

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \tag{3.3}$$

where $c$, and $\{\theta = 1, \cdots, q\}$ are the model's constant parameters, and $\varepsilon_t$ is a white noise error term. In this context, each $y_t$ value represents a weighted moving average of the recent forecast errors, according to (Yao et al., 2013). An example of a first-order moving average $MA(1)$ is:

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}. \tag{3.4}$$

The ACF plot is typically used to identify the MA model's order. The ACF plot typically exhibits a severe cut-off at lag q, indicating that the autocorrelation coefficients are nearly zero for lags beyond q. On the contrary, the PACF figure for the moving average process exhibits a slow decay to zero (Zhang et al., 2013).

### 3.3.2. Autoregressive moving average (ARMA) model

One of the most popular models, the ARMA, combines the benefits of moving average MA$(q)$ and autoregressive AR$(p)$ models. As a result, the stationary time series $AR(p)$and $MA(q)$ models are combined to create the autoregressive moving average model, abbreviated as ARMA $(p, q)$. The initial model was introduced by Peter Whittle in 1951 in his paper on "Hypothesis testing in time series analysis," later refined by (Box et al., 2015). An ARMA $(p, q)$ model of order *(p,q)* can be expressed as:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{3.5}$$

In the equation, $y_t$ represents the original series, and $\varepsilon_t$ represents a series of random errors that adhere to a normal probability distribution. The coefficients $\{\phi = 1, \cdots, p\}$ and $\{\theta = 1, \cdots, q\}$ correspond to the AR and MA terms, respectively.

Determining the model's order (p,q) can be done by analyzing the graphical representations of the ACF and PACF. However, estimating the appropriate values for the ARMA model's parameters p and q based solely on the ACF and PACF plots can be challenging. An alternative approach is to utilize the Akaike information criterion (AIC) to select a model that better fits the input time series. The AIC measures the model's fit quality and penalizes adding more parameters to models. Consequently, this penalty deters overfitting the model. Therefore, the optimum ARMA model is obtained by minimizing the AIC (Bisgaard & Kulahci, 2011).

$$AIC = -2\log(L) + 2(p + q + k + 1) \tag{3.6}$$

Where the value of $k = 0$ if $c = 0$ and $k = 1$ if $c \neq 0$. The last phrase in this parenthesis indicates the model's parameter number. $\mathcal{L}$ indicate the likelihood of the data,

## ARIMA model

The Autoregressive Integrated Moving Average (ARIMA) model predicts a variable based on a linear relationship with its previous values. For stationary time series analysis, the AR, MA, and ARMA models that we covered in earlier sections are preferable (Zhang et al., 2013). Time series, however, are typically non-stationary in the real world. To fit stationary models, removing the variance originating from non-stationary sources in time series is essential. (Chatfield & Xing, 2019). The ARIMA model, introduced by (Box et al., 2015), effectively addresses the issue of non-stationary data by incorporating a differencing mechanism. This approach overcomes the limitation associated with non-stationarity. (Box George et al., 1976; Moghram & Rahman, 1989), offered one remedy for this.

The first step in ARIMA models is to use differencing to eliminate this non-stationarity. To achieve this, the current observation is subtracted from the observation from the previous time step. One method to perform first-order differencing is by substituting $y_t = y_t - y_{t-1}$. The stationary model fitted to the differenced data needs to be summed or integrated to obtain a model for the original non-stationary data. The ARIMA model is therefore referred to as "Integrated" ARMA. The ARIMA $(p, d, q)$ process is described in its generic form (Hyndman & Athanasopoulos, 2018).

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \qquad (3.7)$$

In the differenced new series, denoted as $y'_t$, the right-hand side "predictors" consist of lagged errors and lagged values of $y_t$. The term $\varepsilon_t$ represents white noise with a distribution of $WN(0, \sigma^2)$. The coefficients φ and θ represent the values of the AR and MA terms, respectively, where φ ranges from 1 to p and θ ranges from 1 to q.

Utilizing the backshift notation simplifies working with numerous complex models that combine these elements. For instance, the following is how Equation (3.8) can be written in backshift notation:

$$\left(1-\phi_1 B-\cdots-\phi_p B^p\right)(1-B)^d y_t=c+\left(1+\theta_1 B+\cdots+\theta_q B^q\right)\varepsilon_t \qquad (3.8)$$

The approach to estimating the model parameters (p, q) in the ARIMA model follows a similar method to Akaike's Information Criterion (AIC) approach defined in the ARMA model. Once the number of differences is determined, the ARIMA(p, d, q) model can be defined as follows:

- AR: The autoregression model captures the dependent relationship between the observation and lag observations.
- p: The number of lagged observations included in the model.
- I: Integration is performed to differentiate the raw observations and make the time series stationary.
- d: The number of differences applied to achieve stationarity.
- MA: The model incorporates the relationship between an observation and a residual error derived from a moving average applied to lagged observations.
- $q$: Whether the moving average window or order is large or small.

### 3.4. Linear Models Approaches

Two linear models are discussed to forecast temperature-sensitive electricity use. ASHRAE's change-point model (Sanz-Bobi et al., 2012), which ASHRAE first put forth in the 1990s, is the first linear model. The change point model utilizes five parameters ($\beta base$, $Th$, $Tc$ , $\beta h$, $\beta c$) to describe the relationship between energy consumption and ambient temperature, as depicted in Figure 3.1 and Equation 3.9. The base represents the initial load, which is the level of energy consumption observed when the ambient temperature falls within the range [$Th$, $Tc$]. As the outside temperature drops below the heating change point $Th$, the city-level energy usage increases in response to the rising heating demand.

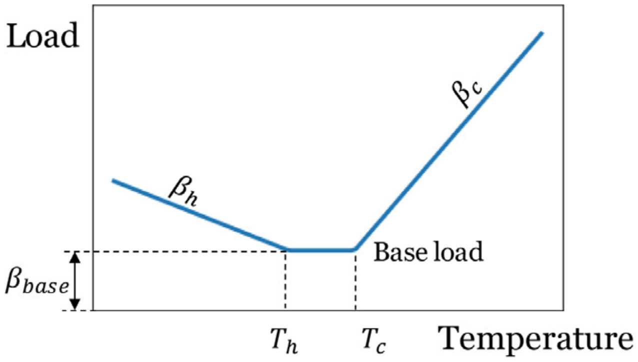


Fig.3.1. ASHRAE's change point model.

Similarly, when the outdoor temperature surpasses the cooling transition point $Tc$, the city-level energy consumption experiences a rise alongside the increasing cooling demand. The slope of the cooling side ($\beta c$) and the heating side ($\beta h$) reflects the sensitivity of the city-level load to temperature variations. In cases where electricity consumption is the primary focus, $\beta h$ would typically be smaller than $\beta c$ since most air-conditioned buildings rely on electricity for cooling, while natural gas is often used for heating. The widely-used five-parameter change point model developed by ASHRAE is a valuable tool for predicting and benchmarking building-level energy performance (Mantel et al., 2019). In this study, we employed the five-parameter change point (5-p) model to forecast energy consumption at the city level.

$$\text{load}(T) = \begin{cases} \beta_{\text{base}} + \beta_h \times (T_h - T), \text{ if } T < T_h \\ \beta_{\text{base}}, \text{ if } T_h < T < T_C \\ \beta_{\text{base}} + \beta_c \times (T - T_c), \text{ if } T_C < T \end{cases} \tag{3.9}$$

The second linear model employed in this study is the Heating/Cooling Degree Hour (HCDH) model. The heating/cooling degree hour method is widely recognized in the heating, ventilation, and air conditioning (HVAC) sector for estimating heating and cooling energy requirements (Runge et al., 2020). Heating and cooling degree days have also been extensively used as proxy variables to assess the impact of climate change on power demand and determine climate zones in the United States (Bahmanyar et al., 2020).

Following Equation 3.10, the difference between the ambient temperature and the heating base temperature ($Tb_h$) and cooling base temperature ($Tb_c$) is summed to calculate the heating degree hours (HDH) and cooling degree hours (CDH). Heating typically begins when the outdoor temperature falls below $Tb_h$, and the amount of heating required can be determined by accumulating the differences between the base temperature and the outdoor temperature ($Tb_h - T_i$). Cooling and Heating degree hours are frequently employed in various applications such as evaluating building thermal insulation (Natarajan et al., 2020), estimating energy demand for buildings (Bünning et al., 2020), and more. This study used a linear regression approach to analyze the daily energy consumption at the municipal level, incorporating the HCDH model as one of the variables.

$$
\begin{aligned}
HDH &= \sum_{i=1}^{24} max\big(0, (Tb_h - T_i)\big) \\
CDH &= \sum_{i=1}^{24} max\big(0, (T_i - Tb_c)\big) \\
\text{load}(T) &= \beta_0 + \beta_1 \times HDH + \beta_2 \times CDH
\end{aligned}
\tag{3.10}
$$

When constructing any model, it is crucial to make appropriate selections for the base temperature ($Tb_h$, $Tb_c$ in the HCDH model) and the transition temperature ($T_h$, $T_C$ in the five-parameter model). This decision-making process regarding temperature thresholds is a common challenge in model development (Kontokosta & Tull, 2017).

## 3.5. Machine Learning Approaches

When exploring machine learning methodologies, we have considered Random Forest (RF), K-nearest neighbor (KNN), and Support Vector Machine (SVM) as part of our analysis.

### 3.5.1. Random forest model

Unlike ensemble methods that combine the outputs of multiple weak models to create a stronger model, Random Forest is an ensemble learning technique that constructs multiple decision trees and aggregates their predictions (Dudek, 2011). The classification and regression tree (CART) model, introduced by Breiman (Zhao et al., 2012)) is a collection of binary decision trees that are combined in this approach. In the random forest model, an individual tree is formed using a bootstrap sample randomly selected at each node, along with a subset of learning points and features (predictors or input variables) at each node (Dudek, 2011). The bootstrap technique is a sampling method that involves random sampling with replacement. It is used to obtain data from a large dataset by resampling. By creating each tree using a randomly selected dataset through bootstrapping, this approach helps mitigate bias and enables the evaluation of solution stability (Dudek, 2011). A binary decision tree is structured so that each internal node has two outgoing edges corresponding to a left child and a right child. These split nodes contain test functions that are applied to incoming input. The final nodes of the tree, referred to as leaves, store the test results.

The Random Forest (RF) technique utilizes decision trees of this form, specifically using the CART (Classification and Regression Tree) algorithm, to tackle classification and regression problems (Di Leo et al., 2020). The subset of data sets and the decision tree for each dataset are created from random bootstrapped data. The final decision is made by combining the ensemble, which is achieved in the regression case by averaging the output or by voting in classification. In his research, Breiman observed that increasing the number of trees in a Random Forest (RF) model prevents overfitting and limits generalization errors, which can be evaluated using an out-of-bag (OOB) error. The OOB score, or out-of-bag prediction, is a proprietary validation method used in Random Forest.

Since only a portion of the training points is included in each bootstrap training set, this group is known as out-of-bag (OOB) samples. For testing reasons, these unused training data might be used. As a result, the proportion of correctly classified out-of-bag samples serves as a proxy for evaluating the Random Forest model's correctness. On the other hand, the percentage of out-of-bag samples that the model incorrectly categorizes is known as the OOB error. Similar to N-fold cross-validation, this error is estimated (Deif, Solyman, Alsharif, et al., 2021). Once the OOB error has been resolved, the training can be finished. The out-of-bag error and the variable importance measure are two significant characteristics that set the Random Forest (RF) model apart. The example below can be used to demonstrate the RF algorithm for regression (Kosek & Gehrke, 2016):

1 For $k = I$ to K :

Step (1): From the training set, create a bootstrap $L$ of size $N$.

Step (2): To construct the random-forest tree $T_k$ using the bootstrapped data, the following procedures are recursively repeated for each node in the tree until the minimum node size m is reached:

Step (2.1): From the $n$ variables, choose $F$ variables at random.

Step (2.2): Select the $F$ most optimal variable or split-point.

Step (2.3) Splitting the node results in creating two daughter nodes.

2 Generate the output by aggregating the ensemble of trees $\{T_k\}_{k=1,2,\ldots,K}$. To forecast at a new point $x$:

$$f(x) = \frac{1}{K}\sum_{k=1}^{K} T_k(x) \tag{3.10}$$

To perform a random forest model, two crucial parameters need to be specified: the number of trees (ntree) in the forest and the number of randomly selected input variables (mtry) at each node (Kosek & Gehrke, 2016). Additional trees are incorporated during training until the out-of-bag (OOB) error reaches a stable state (Chetty et al., 2020). Several researchers say the default parameter value can produce satisfactory results (Zhou et al., 2022). The primary advantages of the model include its robustness to parameter values, ability to generalize well, and built-in cross-validation (Dudek, 2011). Outliers for the decision tree and bootstrapping building do not impact this strategy.

### 3.5.2. The K-nearest neighbor model

Fix and Hodges Jr. created the k-nearest neighbor (KNN), which Cover and Hart later formalized for classification applications (Al-Qahtani & Crone, 2013). K-nearest neighbors (KNN) is an instance-based learning algorithm that relies on the differences between features in the labeled dataset.

Based on distance functions, it looks for a set of k samples close to unknown ones. Here, the k examples most similar to and closest to the new data point are found using a labeled dataset bunch. Because of this, the algorithm relies its prediction on how similar the newly entered data are to the training observations. This algorithm keeps the entire training set in memory during learning. Predicting unknown samples in both regression and classification tasks involves comparing the labels or classes of the new input data to instances in the training set. In the regression case, the average of the response variables is used as the predicted value for the unknown samples. The predicted class is determined by selecting the most prevalent class value among the k-nearest samples in classification.

The prediction is the average of the associated targets, and the KNN model employs the k closest cases in the training set for a particular instance X to compute $\hat{Y}$,. The model can be expressed in the most basic manner.

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \qquad (3.11)$$

Where $N_k(x)$ is the collection of the training sample's nearest points, $x_i$, as the parameter k increases to 1, Determining the optimal value for k becomes challenging as the error decreases to 0 on the training set but increases on the test set. This is because the KNN model has a high variance and low bias. Similarity metric (certain distance functions) is used to determine how close together two data points are and to calculate their distance, $d$. The most prevalent function in this area is the Euclidean distance, which calculates the separation between the points x and y as follows:

$$\mathrm{d}(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.12}$$

According to the research paper's benefits and drawbacks list (Deif et al., 2015), KNN is effective for a day or longer when forecasting load profiles. Because it takes advantage of local knowledge, it exhibits highly adaptable behavior. K-nearest neighbor search encounters significant limitations due to the extensive storage requirements for maintaining a database of historical data.

### 3.5.3. Support Vector Machine

Support Vector Machines (SVM) were developed and introduced by Boser, Guyon, and Vapnik in 1992, building upon the principles of statistical learning theory. Originally designed for classification tasks, SVM was later adapted for regression (Hyndman & Athanasopoulos, 2018). In the case of Support Vector Regression (SVR), it is recommended to utilize this approach to establish the mapping relationship between the input and output vectors.

Given training data $(x^1, y_1), (x^2, y_2), \dots, (x^l, y_l)$ $x^i \in \mathbb{R}^{\mathrm{n}}, y^i \in \mathrm{R}$ where $l$ is the number of samples. We are looking for a vector $\omega \in \mathbb{R}^{\mathrm{n}}$ and a scalar b such that the quantities $f(x^i) = w^T x^i + b$ for each $i = 1, \dots, l$ are as close as possible to the target $y^i$. After defining a map $\phi(x): x \in \mathbb{R}^{\mathrm{n}} \mapsto \phi(x) \in \mathrm{F}$

In the new feature space, denoted by F, which can be a finite or infinite-dimensional Hilbert space with a scalar product $<\dots>$, the training data is represented as $(\phi(x^i), y_i)$, where $i$ take values from 1 to $l$. The Support Vector Machine (SVM) with a nonlinear regression function can mathematically describe by Equation 3.13.

$$f(x) = f(x,\omega) = \omega\phi(x) + b = \langle \omega, \phi(x) \rangle + b \tag{3.13}$$

The samples in the low-dimension space exhibit nonlinear properties in real-world situations. Adding a non-linear function can transfer each sample to a high-dimension space. Hence, employing linear regression in a high-dimensional feature space makes it feasible to achieve non-linear regression in a low-dimensional space. By using the insensitive loss function $\varepsilon$, which is defined as follows, support vector regression can be utilized to resolve the regression estimation problem.

$$L(y - f(x), x) = |y - f(x)|_{\varepsilon} = \begin{cases} 0, & \text{if } |y - f(x,\omega)| \leqslant \varepsilon \\ |y - f(x,\omega)| - \varepsilon, & \text{otherwise} \end{cases} \tag{3.14}$$

Minimizing Equation 3.14 makes it possible to determine the unknown values of $\omega$ and $b$ in Equation 3.15.

$$R(C,\varepsilon) = \frac{1}{m} C \sum_{i=1}^{m} L_i(y_i - f(x_i), x_i) + \frac{1}{2}\omega \cdot \omega \tag{3.15}$$

where:

$\frac{1}{2}\omega \cdot \omega =$ regularization term

$C =$ regulation parameter

The regression issue can be transformed into computing the least value of Equation 3.15 by including the slack variables $\xi$ and $\xi^*$ in Equation 2.16.

$$R(\omega, b, \xi, \xi^*) = \frac{1}{2}\omega \cdot \omega + C \sum_{i=1}^{m} (\xi + \xi^*) \tag{3.16}$$

$$\text{s.t.} = \begin{cases} y_i - f(x_i) \leq \xi + \varepsilon \\ f(x_i) - y_i \leq \xi^* + \varepsilon \\ \xi, \xi^* \geq 0 \end{cases} \begin{pmatrix} i = 1,2,\cdots,m \\ x_i \in R^n \end{pmatrix}$$

Equation 3.17 dual problem can be created by adding the Lagrange multipliers $\alpha_i$ and $\alpha_i^*$:

$$\text{m}\ \frac{1}{2}\sum_{i=1}^{m} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(x_i, x_j) + \varepsilon \sum_{i=1}^{m} (\alpha_i + \alpha_i^*) - \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \tag{3.17}$$

$$\text{s.t. } = \begin{cases} \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) = 0 & i = 1,2,\cdots,m \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

A vector comprising several Lagrange multipliers serves as the solution. Some of the values in these Lagrange multipliers are zero, whereas others are not zero. The support vectors, which can be used to calculate regression, are input vectors with non-zero Lagrange multipliers. Therefore, Equation 3.18 can be used to represent the nonlinear regression function.

$$f(x) = \sum_{i=1}^{m} (\bar{\alpha}_i - \bar{\alpha}_i^*)\langle\phi(x_i) \cdot \phi(x)\rangle + b \tag{3.18}$$

Because the non-linear function is challenging to solve. By choosing the Kernel function $K(x_i, x)$, which converts the initial data space into a new space with a higher dimension and is the inner product of $\phi(x_i)$ and $\phi(x)$: $K(x_i, x) = \phi(x_i)\phi(x)$, the regression function expression of SVM can be solved quickly. The regression function can then be written as Equation 3.19.

$$f(x) = \sum_{i=1}^{m} (\bar{\alpha}_i - \bar{\alpha}_i^*)K(x_i, x) + b \tag{3.19}$$

Where $\alpha_i$ is the Lagrange multiplier, and $x$ is support vector data.

# CHAPTER FOUR

# OPTIMIZATION METHODS OVERVIEW

This chapter provides a conceptual introduction to the Particle Swarm Optimization (PSO) algorithm and Genetic Algorithm (GA). We also discuss their parameter selection methods and representations of neighborhood topology and provide mathematical justifications for these algorithms.

## 4.1 The Basic Model of the PSO algorithm

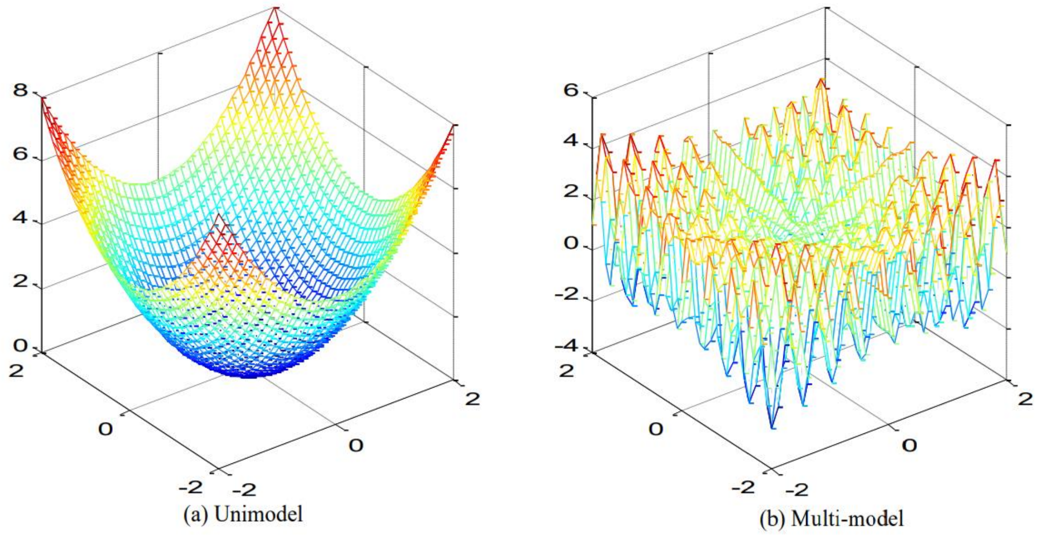
The pioneering work of Kennedy and Eberhart involved tackling the complex problem of non-linear optimization by simulating the behavior of bird flocks. They introduced the concept of function optimization using a particle swarm (Gad, 2022). Think about an n-dimensional function's global optimal, which is defined by

$$f(x_1, x_2, x_3, \dots, x_n) = f(X) \tag{4.1}$$

Let $x_i$ represent the set of independent variables in the given function, which serves as the search space variable. The objective is to find the value $x^*$ that maximizes or minimizes the function $f(x^*)$ within the search space. Consider the functions presented by

$$f_1 = x_1^2 + x_2^2$$

$$\text{and } f_2 = x_1 \sin(4\pi x_2) - x_2 \sin(4\pi x_1 + \pi) + 1 \tag{4.2}$$



(a) Unimodel
(b) Multi-model

Figure 4.1: Plot of the functions $f_1$ and $f_2$.

The minimum value of the function $f_1$ is globally located at $(x_1, x_2) = (0,0)$, or at function $f_1$ origin in the search space, as shown in figure 4.1(a). This indicates that the function is unimodal, with a single minimum. With several local minima, multi-model functions make it more challenging to determine the global optimum. Given the rugged search space and multiple peaks in the function $f_2$, as illustrated in Figure 4.1 (b), it is necessary for numerous agents to explore the space from different initial positions until at least one agent discovers the optimal global position. Throughout this process, all agents are free to communicate and exchange information (Deif, Attar, Amer, Elhaty, et al., 2022).

The Particle Swarm Optimization (PSO) algorithm is an approach for multi-agent parallel search. It involves a particle swarm, each representing a potential solution. In a multidimensional search space, these particles move and update their positions based on their own experiences and neighbors' experiences.

The vector can represent the update of each particle's location in the search space $x_i^t$, where *i* denotes the particle index and *t* represents the time step:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \text{ with } x_i^0 \sim U(x_{\min}, x_{\max}) \tag{4.3}$$

The velocity vector of particle $i$, denoted as $v_i^t$, plays a crucial role in the optimization process by incorporating information from both the particle's own experience and the collective experience of all particles. It is influenced by the uniform distribution $U(x_{\min}, x_{\max})$, where $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values, respectively.

In the PSO procedure, the fitness of each particle is computed, and both the personal best (best value for each particle) and the global best (best value among all particles in the swarm) are determined. The algorithm enters a loop from random positions to search for the optimal solution. The velocities of the particles are updated using information from the personal and global bests, and these updated velocities are then used to update the positions of each particle. The loop continues until a predefined stopping condition is met (Deif, Hammam., Hammam, & Solyman, 2021). Essentially, two variations of the PSO algorithm have been developed: the Global Best ($g_{\text{best}}$) PSO and the Local Best ($l_{\text{best}}$) PSO. These variations differ in the size of their communities or neighborhoods.

### 4.1.1 Global Best PSO ( gbest PSO)

In the PSO approach, each particle (denoted by $i$) in the swarm, where $i$ ranges from 1 to $n$ (where $n > 1$), is influenced by the global best-fitting particle. Each particle has its current position in the search space, $x_i$, current velocity, $v_i$, and a personal best position in the search space, $P_{best}$, $i$. The personal best position, $P_{best,i}$, represents the location in the search space where particle $i$ achieves the lowest value according to the objective function $f$ in the case of a minimization problem. The global best position, denoted by $G_{\text{best}}$, is the position with the lowest value among all the personal best positions $P_{\text{best, i}}$ within the swarm. The personal and global best values are updated using the formulas (4.4) accordingly. For a minimization problem, the best personal position, $P_{best,i}$, at time step $t+1$ (where $t$ ranges from $0$ to $N$), can be determined as

$$P_{\text{best, }i}^{t+1} = \begin{cases} P_{\text{best, i}}^{t} & \text{if } f(x_i^{t+1}) > P_{\text{best,}i}^{t} \\ x_i^{t+1} & \text{if } f(x_i^{t+1}) \leq P_{\text{best, }i}^{t} \end{cases} \tag{4.4}$$

Where the fitness function is $f: \mathbb{R}^n \rightarrow \mathbb{R}$.The calculation for the global best position $G_{\text{best}}$ at time step $t$ is

$$G_{\text{best}} = \min\{P_{\text{best,}i}^{t}\}, \text{ where } i \in [1, \ldots, n] \text{ and } n > 1 \tag{4.5}$$

The personal best position ($P_{best,i}$) represents the best position a specific particle i has achieved since the initial time step, an important individual distinction. On the other hand, the global best position ($G_{\text{best}}$ ) represents the best position any particle in the swarm finds. In the gbest PSO method, the velocity of particle $i$ is determined by considering the difference between its current position and the $G_{\text{best}}$ position:

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t}[P_{best,i}^{t} - x_{ij}^{t}] + c_2 r_{2j}^{t}[G_{best} - x_{ij}^{t}] \tag{4.6}$$

In the gbest PSO algorithm, the velocity vector ($v_{ij}^{t}$) of particle *i* in dimension *j* at time *t* is determined based on various factors. These include the personal best position ($P_{best,i}^{t}$) of the particle found from initialization through time t, the global best position ($G_{\text{best}}$ ) of particle *i* in dimension *j* found from initialization through time t, and positive acceleration constants ($c_1$ and $c_2$ ). These constants help balance the contributions of the particle's best position and the global best position in the equation. Figure 4.2 illustrates the gbest PSO algorithm.

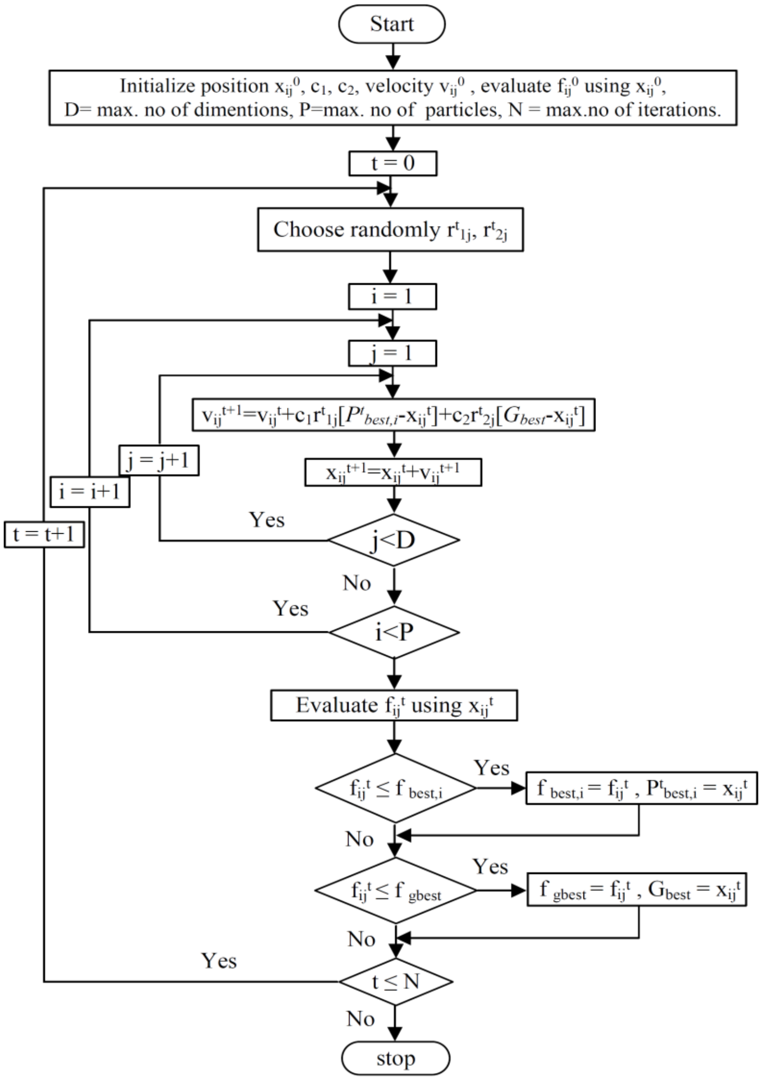

Start
Initialize position $x_{ij}^0$, $c_1$, $c_2$, velocity $v_{ij}^0$ , evaluate $f_{ij}^0$ using $x_{ij}^0$,
D= max. no of dimentions, P=max. no of particles, N = max.no of iterations.
t = 0
Choose randomly $r^t_{1j}$, $r^t_{2j}$
i = 1
j = 1
$v_{ij}^{t+1}=v_{ij}^t+c_1r^t_{1j}[P^t_{best,i}-x_{ij}^t]+c_2r^t_{2j}[G_{best}-x_{ij}^t]$
$x_{ij}^{t+1}=x_{ij}^t+v_{ij}^{t+1}$
j = j+1
i = i+1
t = t+1
Yes
j<D
No
Yes
i<P
Evaluate $f_{ij}^t$ using $x_{ij}^t$
$f_{ij}^t \leq f_{best,i}$
Yes
$f_{best,i} = f_{ij}^t$ , $P^t_{best,i} = x_{ij}^t$
No
$f_{ij}^t \leq f_{gbest}$
Yes
$f_{gbest} = f_{ij}^t$ , $G_{best} = x_{ij}^t$
No
Yes
$t \leq N$
No
stop

Figure 4.2: gbest PSO

### 4.1.2 Local Best PSO (lbest PSO)

In the local best PSO approach, the best-fitting particle selected from its neighborhood influences each particle's behavior. The velocity of particle i is determined by considering the personal best position ($P_{best,i}^{t}$) of the particle and the best position ($P_{best,n}^{t}$) of the neighboring particles within its defined neighborhood. This neighborhood-based interaction helps guide the optimization process and update the particle's velocity for improved convergence.

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t} \left[ P_{best,i}^{t} - x_{ij}^{t} \right] + c_2 r_{2j}^{t} \left[ L_{best,i} - x_{ij}^{t} \right] \tag{4.7}$$

Where $L_{best,i}$ denotes the best position each particle has been in since initiation through time $t$ in the vicinity of particle $i$. The optimal PSO method is summarized in the following Figure 4.3:

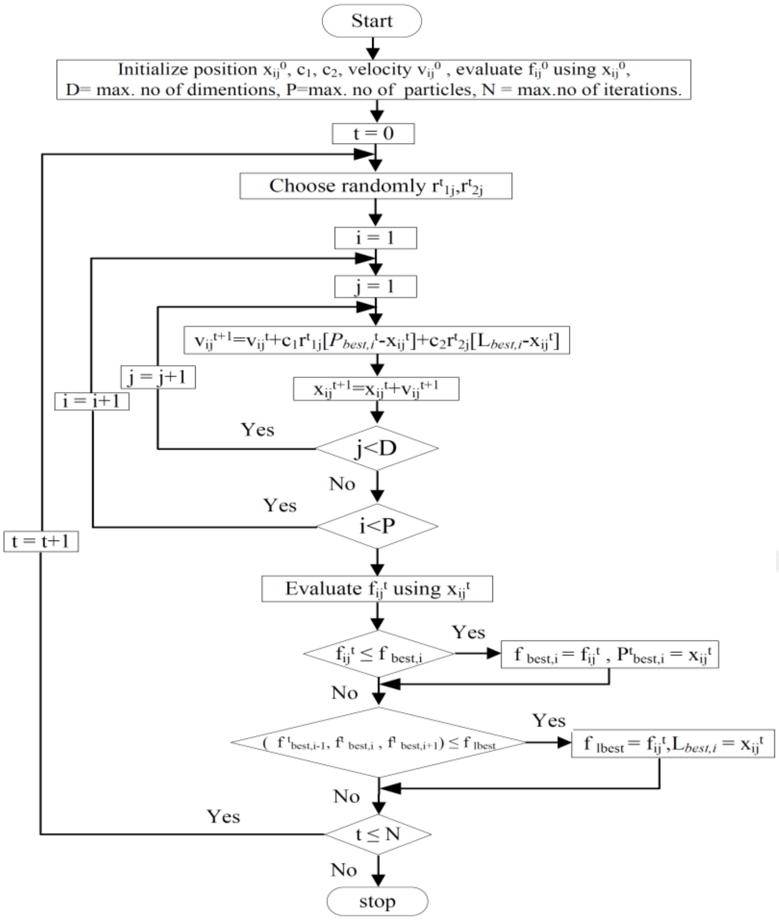

Start
Initialize position x_ij^0, c_1, c_2, velocity v_ij^0 , evaluate f_ij^0 using x_ij^0,
D= max. no of dimentions, P=max. no of particles, N = max.no of iterations.
t = 0
Choose randomly r^t_1j,r^t_2j
i = 1
j = 1
v_ij^t+1=v_ij^t+c_1r^t_1j[P_best,i^t-x_ij^t]+c_2r^t_2j[L_best,i-x_ij^t]
x_ij^t+1=x_ij^t+v_ij^t+1
j = j+1
i = i+1
Yes
j<D
No
Yes
i<P
t = t+1
Evaluate f_ij^t using x_ij^t
Yes
f_ij^t ≤ f best,i
f best,i = f_ij^t , P^t best,i = x_ij^t
No
Yes
( f^t best,i-1, f^t best,i , f^t best,i+1) ≤ f lbest
f lbest = f_ij^t,L_best,i = x_ij^t
No
Yes
t ≤ N
No
stop

Figure 4.3: lbest PSO

Finally, it is clear from Sections 4.1.1 and 4.1.2, respectively, that each particle in the gbest PSO algorithm receives information from the best particle in the entire swarm. In contrast, each particle in the lbest PSO algorithm only receives information from its immediate swarm neighbors.

## 4.2 Genetic Algorithms

There are numerous various ways that genetic algorithms have been implemented. Researchers typically prefer to apply a reliable traditional method rather than construct their solution to each new problem due to the difficulty in generating acceptable parameters. According to Davis, for a Genetic Algorithm to function successfully, it must be the simplest possible for a given situation (Davis, 1991). Figure 4.5 presents a fundamental flowchart outlining the steps involved in a genetic algorithm.

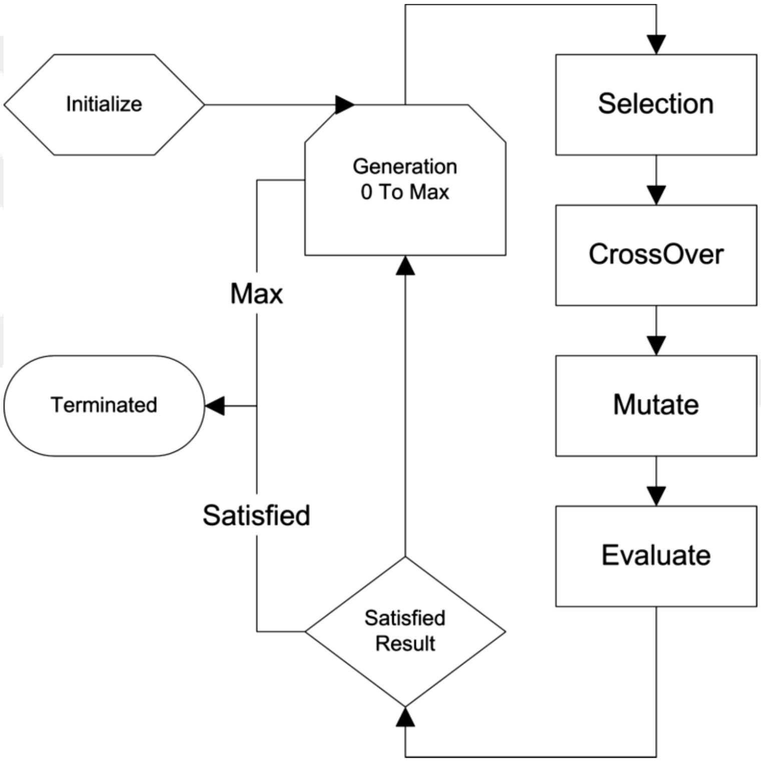


Figure 4.5: A basic flowchart of a genetic algorithm.

### 4.2.1 The genetic information

A DNA chain in an individual correlates to the genetic data in a genetic algorithm. The solution, or more precisely, its properties, are described by the DNA chain. Chromosomes construct the genetic material, which is typically encoded in binary form. (but it can also be in another form of data). A solution is commonly referred to as an individual, and the population of a genetic algorithm is the collection of individuals that make up a generation.

### 4.2.2 Fitness Function Selection

The genetic algorithm's fitness or evaluation function ranks alternative solutions by determining their effectiveness concerning the present issue domain.

The evolutionary algorithm can advance toward promising areas of the search space through selection. Selecting high-fitness individuals is crucial as they are more likely to survive and propagate to the next generation. It is important to balance the selection pressure to ensure optimal genetic exploration. Too much selection pressure can prematurely halt genetic exploration, while insufficient pressure can result in slower evolution. It is generally recommended to apply greater selection pressure towards the end of the genetic search to narrow the search space (1995 Spears). The roulette wheel selection algorithm is frequently utilized. The following steps are included in the roulette wheel selection: (See equations 4.8)

(1) Compute the entire fitness of the population
(2) Compute the likelihood of selection $p_1$ for each individual $v_1$
(3) Compute the accumulative likelihood $q_1$ for each individual $v_t$
(4) Develop a random number from the range [0.1]

$$F = \sum_{i=1}^{\text{PopulationSize}} \text{evaluate } (v_i) \tag{4.8}$$

$$p_1 = \frac{\text{evaluate}(v_i)}{F} \tag{4.8}$$

$$q_i = \sum_{j=1}^{i} p_j \tag{4.10}$$

#### *4.2.3.1 Stochastic Sampling*

During the selection phase, the number of copies of each chromosome is determined based on its probability of survival. The selection phase consists of two steps: figuring out the predicted values for the chromosomes and converting those values to the number of offspring. A chromosome's anticipated value is a precise figure representing the specific number of offspring. The actual predicted value is converted to the number of offspring using the sampling technique. The roulette wheel selection is a widely adopted stochastic selection method. We picture a roulette wheel with slots that vary in size depending on the players' fitness level. A new member of the population is chosen for every turn of the roulette wheel. According to the Schema Theorem, it is acceptable if one person is chosen more than once because the most incredible people tend to multiply, the ordinary people stay the same, and the worst people have the highest mortality rates. (Michalewicz, 1994). For the mathematical application, see equations 4.8.

#### *4.2.3.2 Deterministic Sampling*

Deterministic sampling involves selecting the best chromosomes from both parents and offspring using a deterministic approach. Elitism is one deterministic selection method frequently used with a stochastic selection method. In general selection, the population's best individual will not reach the following generation. Including elitism in genetic algorithms can accelerate the dominance of a superior individual within the population, leading to potential performance improvements in specific areas. Elitism involves selecting and preserving the fittest members of the population, allowing them to directly pass on to the next generation without undergoing any genetic modifications. This approach ensures that the maximum fitness of the population does not decrease from one generation to the next. Typically, elitism promotes faster convergence of the population. However, the impact of elitism on the likelihood of finding the optimal candidate can vary depending on the specific application (Goldberg, 1989).

#### *4.2.3.3 Mixed Sampling*

Mixed sampling combines deterministic and random characteristics.

### 4.2.4 Reproduction

A shared characteristic among all evolutionary algorithms is the requirement for populations to evolve and improve over time. Genetic algorithms often achieve this by using the two techniques of mutation and crossover.

### 4.2.5 Crossover

Crossover, derived from the biological concept, presents a complex process of exchanging chromosomal segments to produce diverse genetic combinations. In genetic algorithms, a crossover operator is commonly employed to facilitate recombination by manipulating pairs of chromosomes. This operation involves exchanging genetic material between parents, resulting in the creation of new offspring. Essentially, crossover combines the genetic material of the two fittest individuals within a population, giving rise to novel genetic strings.

#### *4.2.5.1 1-point crossover*

Genetic algorithms have traditionally used a one-point crossover, the most basic type of crossover. The way a one-point crossover operates is as follows. Figure 4.6 illustrates how the offspring's DNA is formed when a random DNA point is selected. The resulting offspring inherits DNA from the first parent up to that point and DNA from the second parent after that point. It is simple to extrapolate the one-point crossover to any number of crossover points. A 2-point crossover is the least disruptive crossover approach, according to studies. (Spears, 1995).

DNA for Parent 1: 01100101

DNA for Parent 2: 00011110

Crossover point: 3

DNA for Child 1: 00010100

DNA for Child 2: 01101111

Figure 4.6: Single-point crossover example

#### 4.2.5.2 Uniform crossover

Several empirical studies have provided evidence for the benefits of increasing the number of crossover points (Spears, 1995). Instead of selecting random locations, a predetermined template is utilized for the crossover operation. The template consists of randomly chosen bit values, which are then adjusted to match the positions in the string. Figure 4.7 provides a visual representation of this process. Uniform crossover, which involves an average of L/2 crossover points for a string of length L, has also been explored in the literature.

Additionally, studies have shown that this instance is specific to the problem at hand and could be better in general. The most disruptive crossover approach is uniform crossover. (Spears, 1995)

DNA for Parent 1: 01100101
DNA for Parent 2: 00011110
Template: 01010111
DNA for Child 1: 0111111
DNA for Child 2: 00010100

Figure 4.7: Uniform crossover example

#### 4.2.5.3 Adaptive crossover

A genetic algorithm called adaptive crossover chooses the best type of crossover as it runs. There are two methods to accomplish this: locally and globally. This is accomplished locally by extending the DNA strand by one extra chromosome. The 2-point crossover is utilized if both parents have zero chromosomes (0), while the uniform crossover is used if both parents have one (1) chromosome. Additionally, the crossover form employed if the parents have differing chromosomal numbers is chosen randomly. This is done globally by taking into account the entire population. If the chromosome within the population consists mostly of zeros (0), a 2-point crossover technique will be used for the entire population. On the other hand, if the chromosome contains more ones (1), the uniform crossover will be applied to the entire population. This additional chromosome is subject to genetic manipulation by all genetic operators, including mutation and crossover, at both the global and local levels.

#### *4.2.5.4 Guided Crossover*

The guided crossover is a commonly used crossover technique that offers several advantages. This operator is specifically designed to enhance the quality of the solutions generated by the genetic algorithm, increasing the likelihood of approaching the global optima. The guided crossover process involves selecting two individuals from the population. The offspring resulting from the crossover operation is randomly determined along the line connecting the two parents closest to the individual with the best fitness. This approach was introduced by (Rasheed in 1999).

### 4.2.6 Mutation

A genetic operation called mutation modifies one or more gene values on a chromosome. The genetic search includes mutation as a crucial component since it keeps the population from plateauing at any local optimum. The mutation is the mechanism by which random changes are introduced to one or more chromosomes within the DNA string. This process involves altering specific elements, such as flipping the value of a 0 to 1 or vice versa, in a stochastic manner.

# CHAPTER FIVE

# METHODS AND RESULTS

This chapter provides a detailed description and rationale for the dataset and methods employed in predicting daily electricity consumption at the city level. Furthermore, we present and analyze the outcomes obtained by applying these methods to our specific country case datasets.

## 5.1. Materials:

This study used the dataset published in a research paper (Z. Wang et al., 2021). The dataset comprises ambient temperature data and city-level electricity usage data for three major metropolitan areas in the United States: New York (NY), Sacramento (Sac), and Los Angeles (LA). The data spans from July 2015 to September 2020. The ambient temperature data was obtained from the National Oceanic and Atmospheric Administration (NOAA), while the city-level electricity usage data was collected from the Energy Information Administration (EIA). The raw electricity consumption in three cities is plotted in Fig. 5.1.

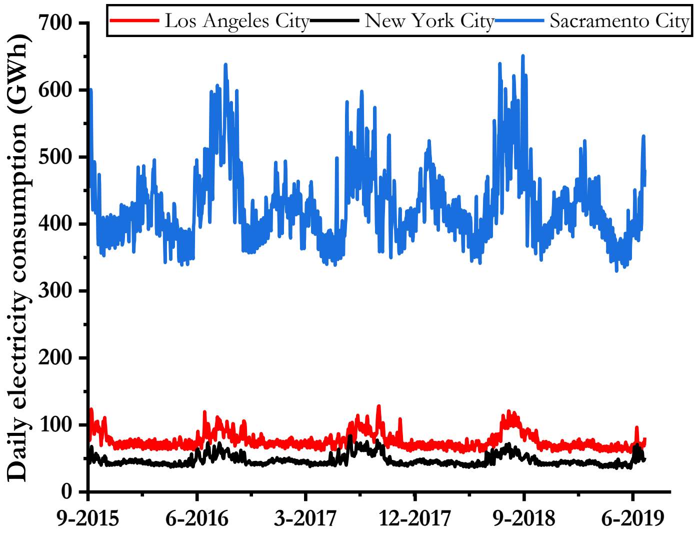


Fig. 5.1. The average hourly electricity consumption in three metropolitan areas.

## 5.2. Methods :

Figure 1 presents an overview of the methodology used to forecast daily electricity consumption at the city level, which includes the following Five phases: (1) Data preparation, (2) Prediction phase, (3) Hyper-parameter optimization, (4) performance evaluation and (5) Predict energy consumption during the pandemic. Each phase is explained in the next subsections.

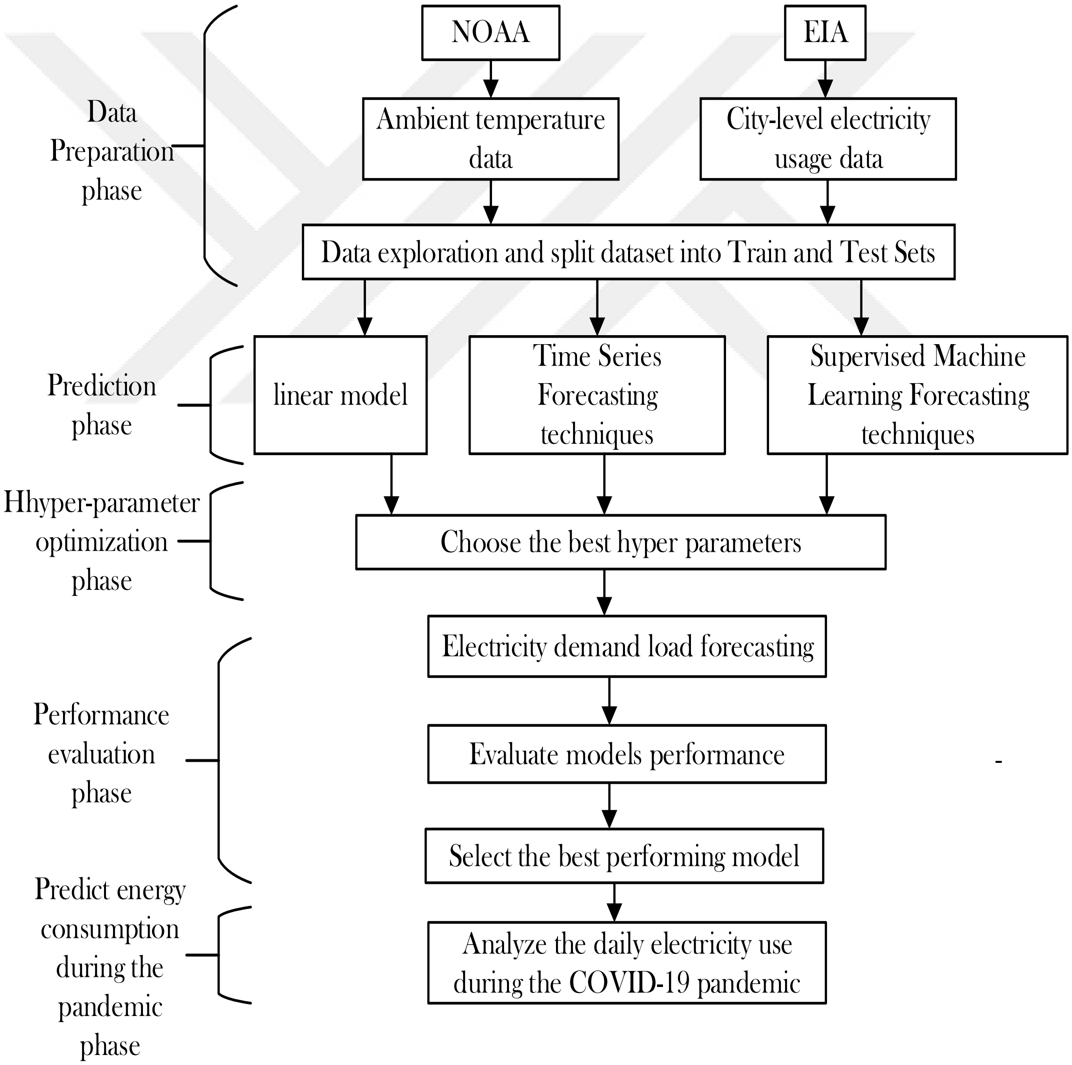

Data Preparation phase
NOAA
EIA
Ambient temperature data
City-level electricity usage data
Data exploration and split dataset into Train and Test Sets
Prediction phase
linear model
Time Series Forecasting techniques
Supervised Machine Learning Forecasting techniques
Hhyper-parameter optimization phase
Choose the best hyper parameters
Electricity demand load forecasting
Performance evaluation phase
Evaluate models performance
Select the best performing model
Predict energy consumption during the pandemic phase
Analyze the daily electricity use during the COVID-19 pandemic

Fig. 5.2. Overall prediction methodology

### Phase (1): Data preparation

The primary dataset was divided into power data, including (Date, time, electricity load, and Day Type (Working Day / non-Working Day)) and weather data, including (Date, time, Ambient temperature, and humidity). Therefore, the power and weather data were merged to be appropriate for the study phases.

The dataset underwent several data preprocessing steps. The daily power load, demand peaks, and average ambient temperature were calculated each day and included as new features in the combined dataset. Any measures with missing values were carefully examined and removed from the dataset. Furthermore, the variables' scale was verified to ensure consistency. Following these data preparation steps, the dataset was randomly split, with 80% allocated for training prediction models and the remaining 20% used for model validation testing.

### Phase (2): Prediction phase

Four forecasting models were developed to predict daily electricity usage in Sacramento, Los Angeles, and New York. These forecasting models belong to two different modeling techniques: Time series forecasting techniques (ARIMA model) and Supervised Machine Learning Forecasting techniques (SVM, RF, and KNN). The details of six forecasting models are illustrated in the following

## I. Time Series Forecasting techniques

The ARIMA model, widely used for forecasting univariate time series, was introduced by (Deif, Solyman, & Hammam, 2021). This linear statistical model decomposes time series into current and past values and random errors. ARIMA combines autoregression (AR) with $p$ past observations, moving average (MA) with $q$ random errors, and differencing ($d$) to achieve stationarity. The representation of ARIMA$(p, q, d)$ is as follows:

$$\Delta^d y(t) = c + \sum_{j=1}^{p} \alpha_j \times y(t-j) + \epsilon(t) + \sum_{j=1}^{\bar{q}} \beta_j \times \epsilon(t-j) \tag{4.1}$$

Let $\Delta = (1 - B)$, where $B$ is the backward operator, and $By(t) = y(t-1), y(t)$ represents the observation data at time $t$. The ARIMA model is defined by a constant $c$, auto-regressive parameters$\alpha_1, \ldots, \alpha_p$, white noise $\epsilon(t)$, and moving average coefficients $\beta_1, \ldots, \beta_q$. The fitting process of an ARIMA model involves four steps:

Step (1): The ARIMA$(p, d, q)$ structure Identification.

Step (2): Parameters Estimation.

Step (3): Examine the estimated residuals through diagnostic checking.

Step (4): Predicting future values using the existing data.

The order q and p of the ARIMA model are determined by analyzing the autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the data, as proposed by Box and Jenkins (1976).

## II. Supervised Machine Learning Forecasting techniques

### Support Vector Machine (SVM)

The SVM algorithms aim to map data points from a low-dimensional space to a higher-dimensional space to achieve linear separability. Given n data points, the objective function of SVM is formulated as follows (Deif, Solyman, Alsharif, et al., 2021; Hammam, Attar, et al., 2022a):

$$\arg\min_{\mathbf{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \text{m} \ \{0, 1 - y_i f(x_i)\} + C\mathbf{w}^T\mathbf{w} \right\} \qquad (4.2)$$

In the equation, **w** represents a normalization vector, and *C* is the penalty parameter for the error term. It is an essential hyperparameter in all SVM models.

The kernel function *f(x)* is used to assess the similarity between two data points$x_i$ and $x_j$. There are various kernel functions available in SVM models, and the choice of kernel type becomes a crucial hyperparameter that needs to be adjusted. Some common kernel types in

SVM include linear kernels, radial basis function (RBF), polynomial kernels, and sigmoid kernels(Hammam, Solyman, et al., 2022)(Ayyarao et al., 2022):

These different kernel functions can be represented as follows:

1 Linear kernel:

$$f(x) = x_i^T x_j \tag{4.3}$$

2 Polynomial kernel:

$$f(x) = \left(\gamma x_i^T x_j + r\right)^d \tag{4.4}$$

3 RBF kernel:

$$f(x) = \exp(-\gamma \lVert x - x' \rVert^2) \tag{4.5}$$

4 Sigmoid kernel:

$$f(x) = \left(\tanh\left(\gamma x_i^T x_j + r\right)\right) \tag{4.6}$$

After selecting a kernel type, several other hyperparameters need to be tuned, as evident in the kernel function equations. One such hyperparameter is the coefficient $\gamma$, denoted as 'gamma' in sklearn. It is a conditional hyperparameter associated with kernel types such as polynomial, RBF, or sigmoid. Additionally, the hyperparameter r is represented by 'coef0' in Sklearn, which is specific to polynomial and sigmoid kernels. Furthermore, the polynomial kernel introduces an additional conditional hyperparameter, *d*, representing the kernel function's' degree'. In the case of support vector regression (SVR) models, there is an additional hyperparameter called 'ε,' which denotes the tolerance or allowable distance error in its loss function(Deif, Hammam., Hammam, & Solyman, 2021).

## 5.3. Random Forest (RF)

The Random Forest algorithm can be summarized as follows:

1. From a training dataset containing m samples and n variables (features), independently construct T decision trees.

2. Each decision tree model uses a bootstrap sample set from the training dataset.

3. At each internal node of the decision tree, a random subset of n' variables (where n' << n) is considered, and the best split is determined based on these selected variables.

4. The decision trees are built without pruning, allowing them to grow to their maximum depth.

The prediction value y for each tree is obtained, and the final prediction value is obtained by aggregating the results from all T trees in the forest. The Random Forest prediction is determined by combining the predictions of each tree.

$$\hat{y} = \frac{1}{T}\sum_{i=1}^{T} \hat{f}_i(x) \qquad (4.7)$$

Let $x \in R^n$ represent a new input, where n is the number of variables. $T$ denotes the total number of trees in the random forest. The prediction $\hat{f}_i(x)$ The ith tree generates the unknown value y for the input x (where *i* ranges from *1 to T*).

During the construction of each tree, there is an important tuning parameter known as mtry. This parameter determines the number of candidate variables selected for node splitting at each iteration. It is crucial to note that mtry must satisfy condition 1 < mtry < n. By selecting mtry to be less than the total number of variables (n), the objective is to improve computational efficiency and reduce the processing time.

### K-Nearest Neighbors (KNN)

According to a similarity metric known as a distance function, the K-nearest neighbors (KNN) method predicts new data points based on them (Deif, Hammam, Solyman, Alsharif, et al., 2021). This method uses a new data sample's proximity to points in the training set to determine its value. In more detail, we determine the distance (such as the Euclidean, Manhattan, or Minkowski distance) between each sample in the test set and every sample in the training set. The K nearest training data samples are then chosen.

In the case of regression, the KNN prediction is obtained by averaging the outcomes of the K nearest neighbors:

$$y = \frac{1}{K}\sum_{i=1}^{k} y_i \tag{4.8}$$

In the equation, $y_i$ represents the outcome or value of the ith example sample, while y represents the prediction or outcome of the query point.

### Phase (3) Hyper-parameter optimization

In order to find the best hyperparameters for regression models, it is essential to explore the hyperparameter space while creating machine learning models efficiently. A search space, also referred to as the configuration space, a search or optimization method used to find hyperparameter combinations, and an evaluation function used to compare the performance of various hyperparameter configurations are the four main components of the hyperparameter optimization process. Obtaining the following is often the main objective of a hyperparameter optimization problem (Deif, Attar, Amer, Elhaty, et al., 2022; Deif, Attar, Amer, Issa, et al., 2022):

$$x^* = \arg\min_{x \in X} f(x) \tag{4.9}$$

where $f(x)$ represents the objective function that needs to be minimized, such as the error rate or the root mean squared error (RMSE). The hyperparameter configuration $x^*$ Corresponds to the optimal value of $f(x)$. Each hyperparameter x can assume any value within the search space $X$. The primary steps of the hyperparameter optimization process are illustrated in Figure 5.3.

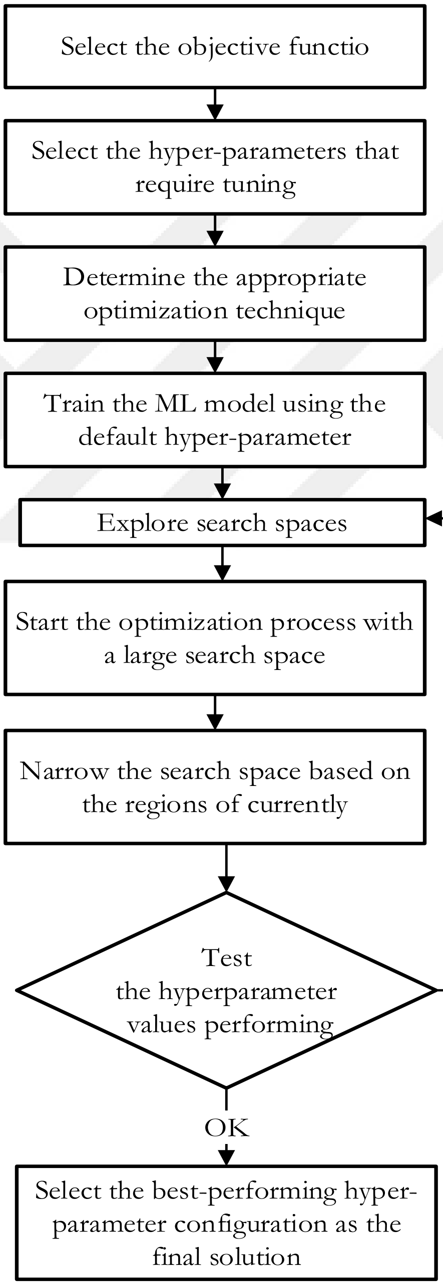

Select the objective functio
Select the hyper-parameters that require tuning
Determine the appropriate optimization technique
Train the ML model using the default hyper-parameter
Explore search spaces
Start the optimization process with a large search space
Narrow the search space based on the regions of currently
Test the hyperparameter values performing
No
OK
Select the best-performing hyper-parameter configuration as the final solution

Fig. 5.3. The flow chart for the main process of Hyper-parameter optimization.

Table 2 shows the search space for the hyper-parameters of Machine Learning Forecasting models

Table 2: Search space for the hyper-parameters of regressor models

| **MACHINE LEARNING Model** | **Hyper-parameter** | **Search Space** |
|---|---|---|
| RF Regressor | n estimators | [10,100] |
| | max depth | [5,50] |
| | Min samples split | [2,11] |
| | min samples leaf | [1,11] |
| | criterion | [’mse’, ’mae’] |
| | max features | [1,13] |
| SVM Regressor | C | [0.1,50] |
| | kernel | [’ linear’, ’poly’, ’rbf’, ’sigmoid’] |
| | epsilon | [0.001,1] |
| KNN Regressor | n neighbors | [1,20] |

**Phase (4) performance evaluation**

The Root Mean Square Error (RMSE) indicators are used to verify the effectiveness of the forecasting model established in this paper.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - x_i)^2} \quad (4.10)$$

In the equation, $y_i$ represents the actual observed value, $x_i$ represents the corresponding forecasted value, and n represents the number of predicted samples.

**Phase (4) Predict energy consumption during the pandemic.**

To address these two pertinent concerns, we selected the best-performing models to investigate: 1) The impact of ambient air temperature, especially during heatwaves, on the daily electricity consumption in the city, and 2) The influence of unexpected public health crises, such as the COVID-19 pandemic, on the daily electricity usage in the city. In this study, our goal is to forecast the daily electricity consumption at a city-wide scale, encompassing various sectors such as buildings, transportation (including electric vehicles and public electric transportation systems), industries, and other public services within the city and its surrounding rural areas.

## 5.4. Implementation

The experiments were performed on an HP machine with an Intel (R) Core (TM) i5-8250U CPU running at 1.60 GHz and Windows 10 operating system, utilizing Python 3.8. Various machine learning models and optimization algorithms were assessed using several open-source Python libraries (Sklearn, Optunity, Hyperband, DEAP ) (Deif, Hammam, Ahmed, Mehrdad Kamarposhti, et al., 2021; Deif & Hammam, 2020; Hammam, Attar, et al., 2022b)

## 5.5. Results

Four experiments were conducted to predict the daily electricity usage in Sacramento, Los Angeles, and New York. In the first experiment, the performance of each Supervised Machine Learning Forecasting technique (SVM, RF, and KNN) was evaluated in predicting daily electricity usage using their default hyperparameter values. The second experiment involved optimizing the hyperparameters of each machine learning model using the PSO and GA algorithms, and their performance was assessed accordingly. Based on the results from experiments 1 and 2, the best-performing forecasting model was selected and compared with another forecasting technique, specifically, Time series forecasting techniques such as the ARIMA model. Lastly, the impact of the COVID-19 pandemic on city-scale daily electricity usage was analyzed.

The initial hyperparameter values for the PSO and GA algorithms used in the experiments can be found in Tables 3 and 4. These values were set to their default values as specified by the Python sci-kit-learn library package (Baghdadi et al., 2022).

Table 3: The initial hyperparameters values for PSO

| PSO Parameter | Parameter values |
|---|---|
| Number of particles | 10 |
| Number of generations | 5 |
| Maximum velocity of each particle | None |
| Local acceleration coefficient (c1) | 1.5 |
| global acceleration coefficient (c2) | 1.5 |

Table 4: The initial hyperparameters values for GA

| GA Parameter | Parameter values |
|---|---|
| population_size | 10 |
| gene_mutation | 0.10 |
| gene_crossover | 0.5 |
| generations_number | 1.5 |

Tables 5 and 6 show the best hyperparameters obtained with different optimization techniques (GA and PSO) for machine learning models. In our experiment, we employed 5-fold cross-validation to determine the optimal parameters of each model in Sacramento, Los Angeles, and New York City. We observed that the convergence was achieved well before reaching 100 iterations, and we decided to stop the process when the variation between iterations dropped below 0.5%.

Table 5: Hyperparameter values after GA tuning

| Machine learning model | Hyper-parameter | Los Angeles | Sacramento | New York |
|---|---|---|---|---|
| RF Regressor | n estimators | 24 | 32 | 80 |
| | max depth | 50 | | |
| | min samples split | 2 | | |
| | min samples leaf | 5 | 2 | 1 |
| | criterion | 'mse' | 'mae' | 'mse' |
| | max features | 10 | | |
| SVM Regressor | C | 30 | | |
| | kernel | 'poly' | | |
| | epsilon | 0.001 | | 0.01 |
| KNN Regressor | n neighbors | 6 | 15 | 10 |

Table 6: Hyperparameter values after PSO tuning

| **Machine learning model** | **Hyper-parameter** | Los Angeles | Sacramento | New York |
|---|---|---|---|---|
| RF Regressor | n estimators | 22 | 36 | 82 |
| | max depth | 40 | | |
| | min samples split | 6 | | |
| | min samples leaf | 1 | 2 | 1 |

| | | | | |
|---|---|---|---|---|
| | criterion | 'mse' | | |
| | max features | 8 | | |
| SVM Regressor | C | 50 | | |
| | kernel | rbf | | |
| | epsilon | 0.001 | | 0.01 |
| KNN Regressor | n neighbors | 6 | 5 | 7 |

Table 7 presents a tabular format displaying the performance of each regressor using the default hyperparameters. On the other hand, Tables 8 and 9 showcase the performance of each optimization algorithm applied to RF, SVM, and KNN regressors evaluated on the MNIST dataset after a comprehensive optimization process. The dataset used in our study spanned from July 2015 to June 2019, covering four years. The first three years were utilized for training the model, while the final year was reserved for validation. We excluded the 2020 data due to the distortion in electricity consumption behavior caused by the COVID-19 curtailment measures.

Table7: Performance results of applying regression models with default Hyperparameter values

| | **RMSE (GWh)** | | |
|---|---|---|---|
| | RF | SVM | KNN |
| **Los Angeles** | 8.383 | 8.850 | 7.4786 |
| **Sacramento** | 5.343 | 6.4569 | 4.9525 |
| **New York** | 25.221 | 31.476 | 24.565 |

Table 8: Performance results of applying GA optimization algorithm to the regression models

| | RMSE (GWh) | | |
|---|---|---|---|
| | RF-GA | SVM-GA | KNN-GA |
| **Los Angeles** | 7.7128 | 8.1427 | 6.8804 |
| **Sacramento** | 4.9158 | 5.940 | 4.5563 |
| **New York** | 23.204 | 28.958 | 22.600 |

Table 9: Performance evaluation of applying the PSO algorithm to the regressor models

| | RMSE (GWh) | | |
|---|---|---|---|
| | RF-PSO | SVM-PSO | KNN-PSO |
| **Los Angeles** | 6.7067 | 7.0806 | 5.9829 |
| **Sacramento** | 4.2746 | 5.1655 | 3.9620 |
| **New York** | 20.17 | 25.181 | 19.652 |

Tables 7, 8, and 9 demonstrate the overall impact of hyperparameter tuning using optimization algorithms (PSO and GA) on enhancing the regression performance of machine learning models. The results generally indicate a notable improvement achieved through the optimization process.

When comparing the performance of PSO and GA in tuning the hyperparameters of machine learning models, PSO showed superior performance, with an almost 20% reduction in the RMSE value for all regression machine learning models. While it’s reduced by almost 8% when using GA.

Among the three regression models tuned using PSO, the KNN model outperformed the RF and SVM model in all three metropolitan areas. KNN performs the best in Sacramento (RMSE=3.96). But performs the worst in New York (RMSE=19.65).

Additionally, the KNN regressor is easier to implement because it has the lowest number of Hyper-parameter (one Hyper-parameter) compared with three Hyper-parameters for the SVM regressor and six Hyper-parameters for RF regressor. Consequently, the KNN regression model's optimization step typically takes substantially less time to compute than other models.

As the number of hyperparameters increases, the search space's dimensionality and the problems' complexity grow exponentially. Consequently, the total time required to evaluate the objective function also increases exponentially (Alshehri et al., 2021; Mahajan et al., 2022). Hence, it becomes crucial to mitigate the impact of large search spaces on execution time by enhancing current hyperparameter optimization methods.

Based on the previous experiment, the best-performing forecasting model achieved by using KNN with the PSO method will Compare with the ARIMA model that belongs to Time series forecasting techniques. The comparison results have recorded in Table 1.

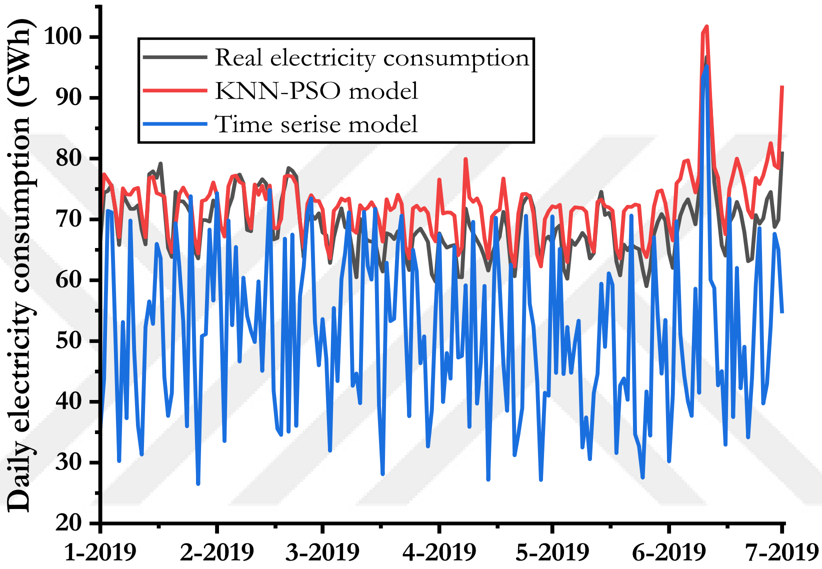
We used the auto.arima() function (Natarajan et al., 2020; Oyelade et al., 2022) to determine the best hyper-parameters values ARIMA$(p, q, d)$ for the ARIMA model. The optimal parameters results and RMSE value for all the three areas are shown in table .10

Table 10: Optimal Hyperparameter values
for the ARIMA model after the optimization process

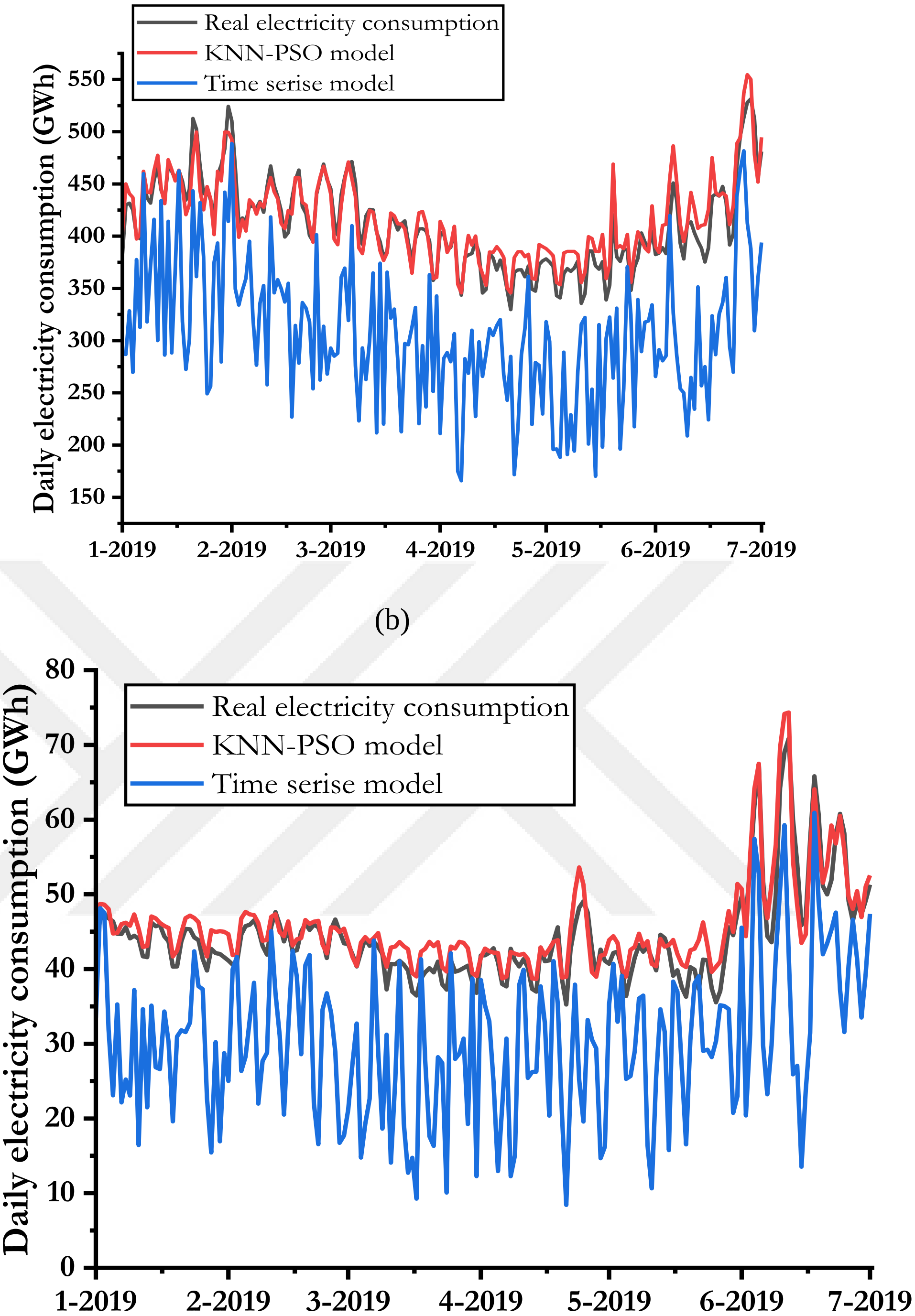
| **City** | **ARIMA$(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{d})$** |
|---|---|
| **Los Angeles** | (1,1,5) |

| | |
|---|---|
| **Sacramento** | (5,3,5) |
| **New York** | (1,3,5) |

Similar to the previous experiment, the two models have trained for the first three years (from July 2015 to June 2018) and kept the last year (2019) for forecasting. The results for the daily electricity consumption forecast in 2019 are shown in Fig. 5.4.



(a)



(b)

(c)

Fig.5.4. Predictions results for the KNN-PSO and ARIMA models on the test dataset (Jan.2019–June 2019). (a) Los Angeles, (b)New York, and (c) Sacramento city.

Figure 5.4 illustrates that the KNN-PSO model's forecasting curve aligns more closely with the actual daily electricity consumption curve than the ARIMA model. This suggests that the KNN-PSO model exhibits greater robustness in handling missing data. Additionally, time-series modeling relies on the sequential order of input data to capture temporal information. Therefore, when data is missing, imputation becomes necessary, introducing further complexity in data preprocessing. In contrast, tabular data models utilize additional features (such as Holiday Day and Day type) to encode temporal information, mitigating the impact of missing data. Table 11 presents the RMSE values for the KNN-PSO and ARIMA models, allowing for a comparison of their performances. The results indicate that the KNN-PSO model outperforms the ARIMA model, exhibiting the lowest RMSE across all metropolitan areas. Consequently, it can be inferred that the proposed KNN-PSO model offers the highest prediction accuracy among the evaluated models.

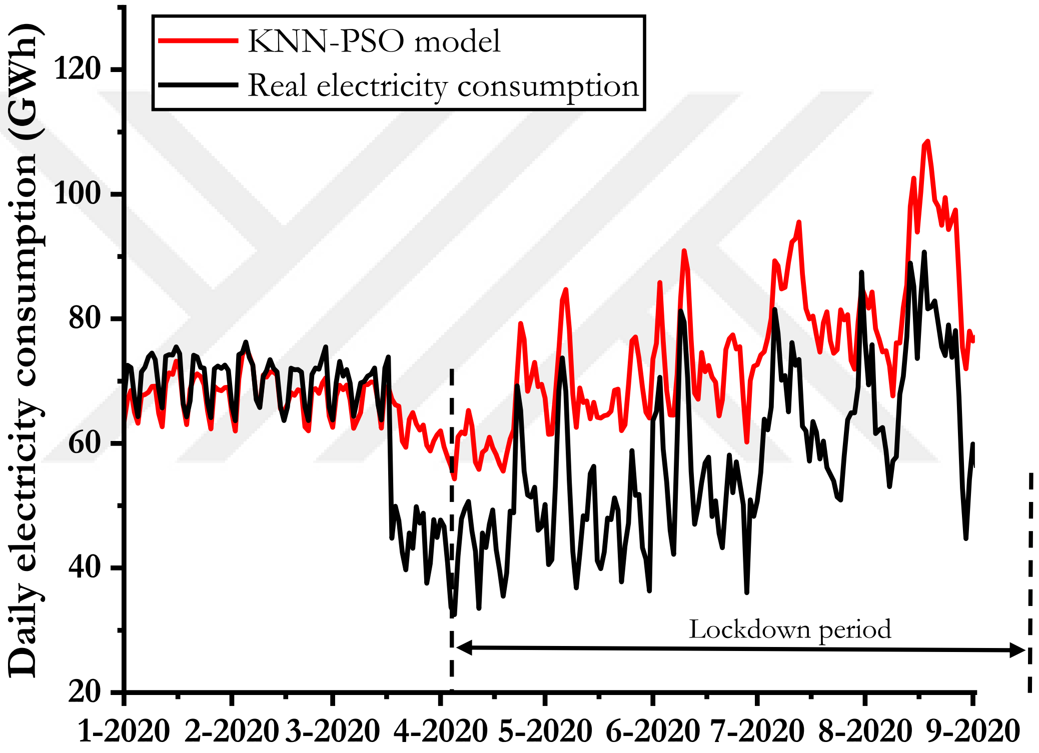
Table 11: Performance evaluation of KNN-PSO and ARIMA models

| City | ARIMA | KNN-PSO |
|---|---|---|
| **Los Angeles** | 25.142 | 5.982958 |
| **Sacramento** | 16.472 | 3.962065 |
| **New York** | 40.232 | 19.65249 |

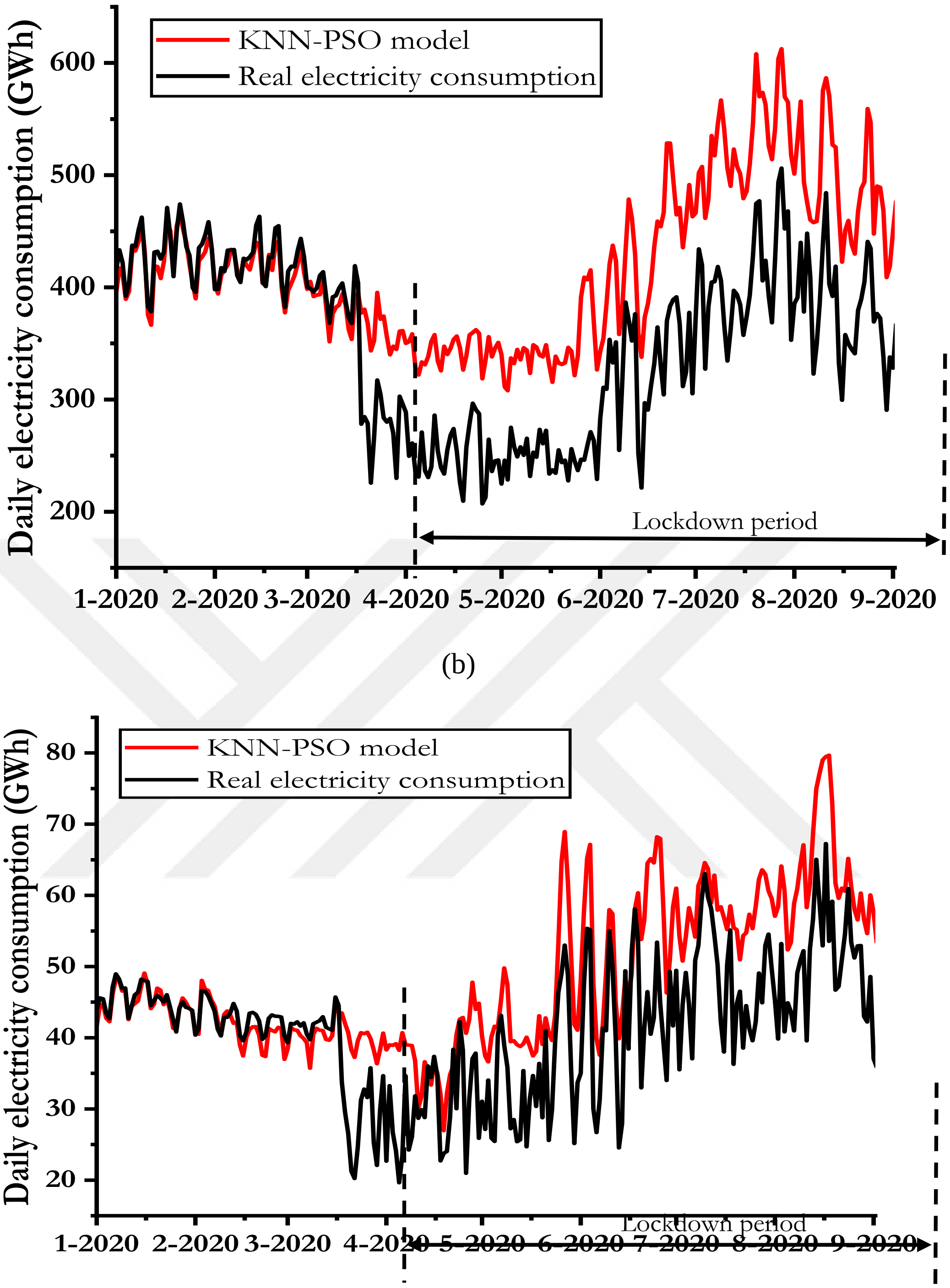
Our final inquiry sought how governmental and individual actions, such as stay-at-home directives, business closures, reduced activity, and shelter-in-place requirements, affected city-level electricity usage in American cities during the COVID-19 epidemic. Electricity consumption on a city-wide scale can act as a real-time gauge of how severe shutdown measures are.

We used the KNN-PSO model due to its higher accuracy compared to other examined models to analyze the impact of COVID-19 on city-scale electricity usage. Data from the pre-pandemic era, from July 2015 to March

2020, was used to train the model. We then used the trained model to predict how much electricity would be used following the outbreak. According to our theory, the model developed using pre-pandemic data might not be able to effectively forecast power consumption in the post-pandemic period if COVID-19 caused changes in consumption patterns. Figure 5.5 shows the predicted outcomes.



(a)



(b)

(c)

Fig.5.5. KNN-PSO model predictions results before and after the COVID-19 lockdown. (a) Los Angeles, (b)New York, and (c) Sacramento city.

Figure 5 illustrates that the forecasted electricity consumption following the end of the lockdown period exceeded the actual electricity usage. In contrast, the electricity demand forecasting before the lockdown period aligned with the demand (indicated by the dotted orange line). It is possible that the COVID-19 epidemic had an effect on electricity demand behavior, given the discrepancy between anticipated and actual electricity usage after mid-March 2020. Particularly, the demand for electricity at the city level was significantly lower than it had been for the same month in 2019. The Sacramento, Los Angeles, and New York urban areas saw a 10% to 20% decrease in daily electricity usage during the COVID-19 epidemic.

# CHAPTER SIX

# CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusion:

This long study examined electricity usage in three significant American metropolises: New York, Sacramento, and Los Angeles. We aimed to analyze the COVID-19 pandemic's effects on electricity consumption and create precise prediction models using machine learning techniques.

We used three well-known machine learning models: Random Forest (RF), Support Vector Machine (SVM), and k-Nearest Neighbors (KNN) to get reliable predictions. However, these models' default hyperparameter settings might not produce the best results. So, to fine-tune the hyperparameters of each model, we used two powerful metaheuristic optimization algorithms: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

We evaluated the performance of each machine learning model using both the default hyperparameter values and the optimized values obtained through PSO and GA. By comparing the results, we identified the KNN regressor model as the most accurate and reliable for predicting electricity consumption. The KNN model, particularly when combined with PSO Optimization, demonstrated the lowest Root Mean Squared Error (RMSE) score, indicating superior predictive capabilities.

Additionally, we compared our proposed machine learning-based forecasting model with a traditional time series forecasting technique, namely the Autoregressive Integrated Moving Average (ARIMA) model. Through this comparison, we aimed to highlight the advantages of our approach in capturing complex patterns and dynamics in electricity consumption data.

Furthermore, we examined the impact of the COVID-19 pandemic on electricity usage. Our analysis revealed a significant reduction in electricity consumption during the pandemic lockdown period in 2020. The Sacramento, Los Angeles, and New York metropolitan areas experienced a 2% to 12% decrease in electricity consumption compared to the corresponding months in 2019. This reduction can be attributed to stay-at-home orders, company shutdowns, and decreased operations due to the pandemic.

Our study provides valuable insights into accurately predicting electricity consumption using advanced machine learning models and optimizing their hyperparameters. We also shed light on the influence of unprecedented events like the COVID-19 pandemic on energy usage patterns, enabling a better understanding of the dynamics and potential opportunities for efficient energy management in urban areas.

### 6.2 Future work :

In the healthcare sector, machine learning has become a powerful technology with the potential to change many facets of healthcare delivery drastically. Future research should concentrate on developing and improving machine learning techniques to handle the unique requirements and difficulties in the healthcare domain.

Creating ensemble and calibrated machine learning methods is one topic of study. These approaches can overcome current algorithms' restrictions and offer quicker and more precise answers to challenging healthcare issues. We can increase performance and reliability in healthcare applications by tweaking models and combining the predictions of various algorithms. This could improve patient outcomes, more precise illness diagnoses, and better treatment planning.

Incorporating AI-based solutions in healthcare offers considerable promise in addition to algorithmic improvements. These applications can aid in identifying and detecting diseases by utilizing several sensors and features. AI systems can find patterns and signs by analyzing enormous volumes of patient data that may not be obvious to human healthcare experts. This may result in early disease detection, individualized treatment programs, and better patient care.

The creation of prediction systems that consider socioeconomic and cultural issues in healthcare is another crucial subject that has to be explored in the future. We can learn important lessons about the emergence and spread of illnesses by including these variables in predictive models. This can help with preemptive planning, resource allocation, and identifying populations that may be more susceptible or in danger. We can better prepare for and lessen the effects of emerging diseases, so protecting public health, by understanding the intricate interplay between social and health aspects.

In conclusion, using machine learning in the healthcare industry has enormous prospects for improving illness detection, prognosis, and general healthcare delivery. We can realize the full potential of machine learning in the healthcare sector by concentrating on calibrated and ensemble techniques, AI-based applications, and predictive models that consider socioeconomic and cultural elements. This may result in better patient outcomes, better decision-making, and a future healthcare system that is more proactive and efficient.

# REFERENCE

Acikgoz, H. (2022). A novel approach based on integration of convolutional neural networks and deep feature selection for short-term solar radiation forecasting. *Applied Energy*, *305*, 117912.

Acikgoz, H., Budak, U., Korkmaz, D., & Yildiz, C. (2021). WSFNet: An efficient wind speed forecasting model using channel attention-based densely connected convolutional neural network. *Energy*, *233*, 121121.

Al-Qahtani, F. H., & Crone, S. F. (2013). Multivariate k-nearest neighbour regression for time series data—A novel algorithm for forecasting UK electricity demand. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Al Mamun, M., & Kermanshahi, B. (2006). *Implementation of an intelligent method to forecast long-term electric demand*.

Alasali, F., Nusair, K., Alhmoud, L., & Zarour, E. (2021). Impact of the COVID-19 pandemic on electricity demand and load forecasting. *Sustainability*, *13*(3), 1435.

Alshehri, M., Kumar, M., Bhardwaj, A., Mishra, S., & Gyani, J. (2021). Deep learning based approach to classify saline particles in sea water. *Water*, *13*(9), 1251.

Altan, A., & Karasu, S. (2020). Ayr{\i}{\c{s}}t{\i}rma yöntemlerinin derin ö{\u{g}}renme algoritmas{\i} ile tan{\i}mlanan rüzgâr h{\i}z{\i} tahmin modeli ba{\c{s}}ar{\i}m{\i}na etkisinin incelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, *20*, 844–853.

Arora, V., & Lieskovsky, J. (2016). *Electricity use as an indicator of US economic activity*.

Ayyarao, T. S. L. V, Ramakrishna, N. S. S., Elavarasan, R. M., Polumahanthi, N., Rambabu, M., Saini, G., Khan, B., & Alatas, B. (2022). War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization. *IEEE Access*, *10*, 25073–25105.

Baghdadi, N., Maklad, A. S., Malki, A., & Deif, M. A. (2022). Reliable Sarcoidosis Detection Using Chest X-rays with EfficientNets and Stain-Normalization Techniques. *Sensors*, *22*(10), 3846.

Baghel, M., Ghosh, A., Singh, N. K., & Singh, A. K. (2016). Short-term electric load forecasting using SVR implementing LibSVM package and Python code. *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, 485–489.

Bahmanyar, A., Estebsari, A., & Ernst, D. (2020). The impact of different COVID-19 containment measures on electricity consumption in Europe. *Energy Research \& Social Science*, *68*, 101683.

Bessec, M., & Fouquau, J. (2008). The non-linear link between electricity consumption and temperature in Europe: A threshold panel approach. *Energy Economics*, *30*(5), 2705–2721.

Beyer, R. C. M., Franco-Bedoya, S., & Galdo, V. (2021). Examining the economic impact of COVID-19 in India through daily electricity consumption and nighttime light intensity. *World Development*, *140*, 105287.

Bisgaard, S., & Kulahci, M. (2011). *Time series analysis and forecasting by example*. John Wiley \& Sons.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley \& Sons.

Box George, E. P., Jenkins Gwilym, M., Reinsel Gregory, C., & Ljung Greta, M. (1976). Time series analysis: forecasting and control. *San Francisco: Holden Bay*.

Brockwell, P. J., & Davis, R. A. (2002). *Introduction to time series and forecasting*. Springer.

Bünning, F., Heer, P., Smith, R. S., & Lygeros, J. (2020). Improved day ahead heating demand forecasting by online correction methods. *Energy and*

*Buildings*, *211*, 109821.

Carvalho, M., de Mello Delgado, D., de Lima, K. M., de Camargo Cancela, M., dos Siqueira, C. A., & de Souza, D. L. B. (2021). Effects of the COVID-19 pandemic on the Brazilian electricity consumption patterns. *International Journal of Energy Research*, *45*(2), 3358–3364.

Chatfield, C., & Xing, H. (2019). *The analysis of time series: an introduction with R*. CRC press.

Chen, B.-J., Chang, M.-W., & others. (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. *IEEE Transactions on Power Systems*, *19*(4), 1821–1830.

Chen, S.-T., Kuo, H.-I., & Chen, C.-C. (2007). The relationship between GDP and electricity consumption in 10 Asian countries. *Energy Policy*, *35*(4), 2611–2621.

Chen, Y. H., Hong, W.-C., Shen, W., & Huang, N. N. (2016). Electric load forecasting based on a least squares support vector machine with fuzzy time series and global harmony search algorithm. *Energies*, *9*(2), 70.

Chetty, R., Friedman, J. N., Hendren, N., Stepner, M., & Team, T. O. I. (2020). *How did COVID-19 and stabilization policies affect spending and employment? A new real-time economic tracker based on private sector data* (Vol. 91). National Bureau of Economic Research Cambridge, MA.

Cicala, S. (2020a). Early economic impacts of covid-19 in europe: A view from the grid. *Unpublished Manuscript*.

Cicala, S. (2020b). *Powering work from home*.

Davoudi, S., Brooks, E., & Mehmood, A. (2013). Evolutionary resilience and strategies for climate adaptation. *Planning Practice \& Research*, *28*(3), 307–322.

Deif, M. A., Attar, H., Amer, A., Elhaty, I. A., Khosravi, M. R., & Solyman, A. A. A. (2022). *Diagnosis of Oral Squamous Cell Carcinoma Using Deep Neural Networks and Binary Particle Swarm Optimization on Histopathological*

*Images: An AIoMT Approach*.

Deif, M. A., Attar, H., Amer, A., Issa, H., Khosravi, M. R., & Solyman, A. A. A. (2022). A New Feature Selection Method Based on Hybrid Approach for Colorectal Cancer Histology Classification. *Wireless Communications and Mobile Computing*, *2022*.

Deif, M. A., Eldosoky, M. A. A., El-Garhy, A. M., Gomma, H. W., & El-Azab, A. S. (2015). Parasympathetic Nervous Signal Damping Using the Adaptive Neuro-Fuzzy Inference System Method to Control Overactive Bladder. *Journal of Clinical Engineering*, *40*(4), 197–201.

Deif, M. A., Hammam., R. E., Hammam, R. E., & Solyman, A. A. A. (2021). Gradient Boosting Machine Based on PSO for prediction of Leukemia after a Breast Cancer Diagnosis. *International Journal on Advanced Science, Engineering and Information Technology*, *11*(12), 508–515.

Deif, M. A., & Hammam, R. E. (2020). Skin lesions classification based on deep learning approach. *Journal of Clinical Engineering*, *45*(3), 155–161.

Deif, M. A., Hammam, R. E., Ahmed, S., Mehrdad Kamarposhti, A., Shahab, S. B., & Rania, E. H. (2021). A deep bidirectional recurrent neural network for identification of SARS-CoV-2 from viral genome sequences. *Mathematical Biosciences and Engineering, 18*(6), AIMS-Press.

Deif, M. A., Hammam, R. E., Solyman, A., Alsharif, M. H., & Uthansakul, P. (2021). Automated Triage System for Intensive Care Admissions during the COVID-19 Pandemic Using Hybrid XGBoost-AHP Approach. *Sensors*, *21*(19), 6379.

Deif, M. A., Solyman, A. A. A., Alsharif, M. H., Jung, S., & Hwang, E. (2021). A hybrid multi-objective optimizer-based SVM model for enhancing numerical weather prediction: a study for the Seoul metropolitan area. *Sustainability*, *14*(1), 296.

Deif, M. A., Solyman, A. A. A., & Hammam, R. E. (2021). ARIMA Model Estimation Based on Genetic Algorithm for COVID-19 Mortality Rates.

*International Journal of Information Technology & Decision Making*, *20*(6), 1775–1798.

Deschênes, O., & Greenstone, M. (2011). Climate change, mortality, and adaptation: Evidence from annual fluctuations in weather in the US. *American Economic Journal: Applied Economics*, *3*(4), 152–185.

Di Leo, S., Caramuta, P., Curci, P., & Cosmi, C. (2020). Regression analysis for energy demand projection: An application to TIMES-Basilicata and TIMES-Italy energy models. *Energy*, *196*, 117058.

Dudek, G. (2011). *Short-Term Load Forecasting using Random Forests*.

Elattar, E. E., Goulermas, J., & Wu, Q. H. (2010). Electric load forecasting based on locally weighted support vector regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(4), 438–447.

Elattar, E. E., Sabiha, N. A., Alsharef, M., Metwaly, M. K., Abd-Elhady, A. M., & Taha, I. B. M. (2020). Short term electric load forecasting using hybrid algorithm for smart cities. *Applied Intelligence*, *50*, 3379–3399.

Ferguson, R., Wilkinson, W., & Hill, R. (2000). Electricity use and economic development. *Energy Policy*, *28*(13), 923–934.

Fezzi, C., & Bunn, D. (2010). Structural analysis of electricity demand and supply interactions. *Oxford Bulletin of Economics and Statistics*, *72*(6), 827–856.

Fezzi, C., & Fanghella, V. (2021). Tracking GDP in real-time using electricity market data: Insights from the first wave of COVID-19 across Europe. *European Economic Review*, *139*, 103907.

Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. *Archives of Computational Methods in Engineering*, *29*(5), 2531–2561.

Ghanbari, A., Naghavi, A., Ghaderi, S. F., & Sabaghian, M. (2009). Artificial Neural Networks and regression approaches comparison for forecasting Iran's annual electricity load. *2009 International Conference on Power Engineering, Energy and Electrical Drives*, 675–679.

Ghiassi, M., Zimbra, D. K., & Saidane, H. (2006). Medium term system load forecasting with a dynamic artificial neural network model. *Electric Power Systems Research*, *76*(5), 302–316.

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

Hammam, R. E., Attar, H., Amer, A., Issa, H., Vourganas, I., Solyman, A., Venu, P., Khosravi, M. R., & Deif, M. A. (2022a). Prediction of Wear Rates of UHMWPE Bearing in Hip Joint Prosthesis with Support Vector Model and Grey Wolf Optimization. *Wireless Communications and Mobile Computing*, *2022*.

Hammam, R. E., Attar, H., Amer, A., Issa, H., Vourganas, I., Solyman, A., Venu, P., Khosravi, M. R., & Deif, M. A. (2022b). *Research Article Prediction of Wear Rates of UHMWPE Bearing in Hip Joint Prosthesis with Support Vector Model and Grey Wolf Optimization*.

Hammam, R. E., Solyman, A. A. A., Alsharif, M. H., Uthansakul, P., & Deif, M. A. (2022). Design of Biodegradable Mg Alloy for Abdominal Aortic Aneurysm Repair (AAAR) Using ANFIS Regression Model. *IEEE Access*, *10*, 28579–28589.

Hirsh, R. F., & Koomey, J. G. (2015). Electricity consumption and economic growth: a new relationship with significant consequences? *The Electricity Journal*, *28*(9), 72–84.

Hong, T., Gui, M., Baran, M. E., & Willis, H. L. (2010). Modeling and forecasting hourly electric load by multiple linear regression with interactions. *Ieee Pes General Meeting*, 1–8.

Huang, B., & Kunoth, A. (2013). An optimization based empirical mode decomposition scheme. *Journal of Computational and Applied Mathematics*, *240*, 174–183.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Janzen, B., & Radulescu, D. (2020). Electricity use as a real-time indicator of the economic burden of the COVID-19-related lockdown: Evidence from Switzerland. *CESifo Economic Studies*, *66*(4), 303–321.

Kaynar, O., Özekicio\uglu, H., & Demirkoparan, F. (2017). Forecasting of Turkey's electricity consumption with support vector regression and chaotic particle swarm algorithm. *Yönetim Bilimleri Dergisi*.

Kaytez, F., Taplamacioglu, M. C., Cam, E., & Hardalac, F. (2015). Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power \& Energy Systems*, *67*, 431–438.

Khan, A. R., Razzaq, S., Alquthami, T., Moghal, M. R., Amin, A., & Mahmood, A. (2018). Day ahead load forecasting for IESCO using artificial neural network and bagged regression tree. *2018 1st International Conference on Power, Energy and Smart Grid (ICPESG)*, 1–6.

Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, *10*(1), 841–851.

Kontokosta, C. E., & Tull, C. (2017). A data-driven predictive model of city-scale energy use in buildings. *Applied Energy*, *197*, 303–317.

Kosek, A. M., & Gehrke, O. (2016). Ensemble regression model-based anomaly detection for cyber-physical intrusion detection in smart grids. *2016 IEEE Electrical Power and Energy Conference (EPEC)*, 1–7.

Kucukali, S., & Baris, K. (2010). Turkey's short-term gross annual electricity demand forecast by fuzzy logic approach. *Energy Policy*, *38*(5), 2438–2445.

Kuo, P.-H., & Huang, C.-J. (2018). A high precision artificial neural networks model for short-term energy load forecasting. *Energies*, *11*(1), 213.

Leach, A., Rivers, N., & Shaffer, B. (2020). Canadian electricity markets during the COVID-19 pandemic: An initial assessment. *Canadian Public Policy*, *46*(S2), S145--S159.

Lee, C.-M., & Ko, C.-N. (2011). Short-term load forecasting using lifting scheme and ARIMA models. *Expert Systems with Applications*, *38*(5), 5902–5911.

Li, R., Jiang, P., Yang, H., & Li, C. (2020). A novel hybrid forecasting scheme for electricity demand time series. *Sustainable Cities and Society*, *55*, 102036.

Lu, H., Ma, X., & Ma, M. (2021). A hybrid multi-objective optimizer-based model for daily electricity demand prediction considering COVID-19. *Energy*, *219*, 119568.

Mahajan, S., Abualigah, L., Pandit, A. K., & Altalhi, M. (2022). Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks. *Soft Computing*, *26*(10), 4863–4881.

Malec, M., Kinelski, G., & Czarnecka, M. (2021). The impact of COVID-19 on electricity demand profiles: A case study of selected business clients in Poland. *Energies*, *14*(17), 5332.

Mantel, C., Villebro, F., dos Reis Benatto, G. A., Parikh, H. R., Wendlandt, S., Hossain, K., Poulsen, P., Spataru, S., Sera, D., & Forchhammer, S. (2019). Machine learning prediction of defect types for electroluminescence images of photovoltaic panels. *Applications of Machine Learning*, *11139*, 1113904.

Matijaš, M., Suykens, J. A. K., & Krajcar, S. (2013). Load forecasting using a multivariate meta-learning system. *Expert Systems with Applications*, *40*(11), 4427–4437.

Menezes, F., Figer, V., Jardim, F., Medeiros, P., & others. (2021). *Using electricity consumption to predict economic activity during COVID-19 in Brazil*.

Moghram, I., & Rahman, S. (1989). Analysis and evaluation of five short-term load forecasting techniques. *IEEE Transactions on Power Systems*, *4*(4), 1484–1491.

Natarajan, K., Bala, P. K., & Sampath, V. (2020). Fault detection of solar PV system using SVM and thermal image processing. *International Journal of Renewable Energy Research (IJRER)*, *10*(2), 967–977.

Oyelade, O. N., Ezugwu, A. E.-S., Mohamed, T. I. A., & Abualigah, L. (2022). Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. *IEEE Access*, *10*, 16150–16177.

Özbay, H., & Dalcali, A. (2021). Effects of COVID-19 on electric energy consumption in Turkey and ANN-basedshort-term forecasting. *Turkish Journal of Electrical Engineering and Computer Sciences*, *29*(1), 78–97.

Runge, J., Zmeureanu, R., & Le Cam, M. (2020). Hybrid short-term forecasting of the electric demand of supply fans using machine learning. *Journal of Building Engineering*, *29*, 101144.

Said, A. Ben, Erradi, A., Aly, H. A., & Mohamed, A. (2021). Predicting COVID-19 cases using bidirectional LSTM on multivariate time series. *Environmental Science and Pollution Research*, *28*(40), 56043–56052.

Sanz-Bobi, M. A., San Roque, A. M., De Marcos, A., & Bada, M. (2012). Intelligent system for a remote diagnosis of a photovoltaic solar power plant. *Journal of Physics: Conference Series*, *364*(1), 12119.

Sathaye, J. A., Dale, L. L., Larsen, P. H., Fitts, G. A., Koy, K., Lewis, S. M., & de Lucena, A. F. P. (2013). Estimating impacts of warming temperatures on California's electricity system. *Global Environmental Change*, *23*(2), 499–511.

Scarabaggio, P., La Scala, M., Carli, R., & Dotoli, M. (2020). Analyzing the effects of covid-19 pandemic on the energy demand: the case of northern italy. *2020 AEIT International Annual Conference (AEIT)*, 1–6.

Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2000). *Time series analysis and its applications* (Vol. 3). Springer.

Sonmez, M. A., & Bagriyanik, M. (2021). Generating manageable electricity demand capacity for residential demand response studies by activity-based load models. *Advances in Electrical and Computer Engineering*, *21*(1), 99–108.

Taylor, J. W. (2008). An evaluation of methods for very short-term load forecasting using minute-by-minute British data. *International Journal of Forecasting*, *24*(4), 645–658.

Türkay, B. E., & Demren, D. (2011). Electrical load forecasting using support vector machines. *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, I--49.

Ucar, F., & Korkmaz, D. (2020). COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. *Medical Hypotheses, 140*, 109761.

Vázquez-Canteli, J. R., Ulyanin, S., Kämpf, J., & Nagy, Z. (2019). Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities. *Sustainable Cities and Society*, *45*, 243–257.

Wang, J., Zhu, S., Zhang, W., & Lu, H. (2010). Combined modeling for electric load forecasting with adaptive particle swarm optimization. *Energy*, *35*(4), 1671–1678.

Wang, Z., Hong, T., Li, H., & Piette, M. A. (2021). Predicting city-scale daily electricity consumption using data-driven models. *Advances in Applied Energy*, *2*, 100025.

Xia, C., Wang, J., & McMenemy, K. (2010). Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks. *International Journal of Electrical Power \& Energy Systems*, *32*(7), 743–750.

Yao, X., Herrera, L., Ji, S., Zou, K., & Wang, J. (2013). Characteristic study and time-domain discrete-wavelet-transform based hybrid detection of series DC arc faults. *IEEE Transactions on Power Electronics*, *29*(6), 3103–3115.

Yi-Ling, H., Hai-Zhen, M., Guang-Tao, D., & Jun, S. (2014). Influences of urban temperature on the electricity consumption of Shanghai. *Advances in Climate Change Research*, *5*(2), 74–80.

Zhang, X., Liu, Y., Yang, M., Zhang, T., Young, A. A., & Li, X. (2013). Comparative study of four time series methods in forecasting typhoid fever incidence in China. *PloS One*, *8*(5), e63116.

Zhao, Y., Yang, L., Lehman, B., de Palma, J.-F., Mosesian, J., & Lyons, R. (2012). Decision tree-based fault detection and classification in solar photovoltaic arrays. *2012 Twenty-Seventh Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 93–99.

Zheng, H., Yuan, J., & Chen, L. (2017). Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies, 10*(8), 1168.

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. *2017 IEEE International Conference on Prognostics and Health Management (ICPHM),* 88–95.

Zhou, X., Ma, H., Gu, J., Chen, H., & Deng, W. (2022). Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Engineering Applications of Artificial Intelligence, 114*, 105139.

# APPENDIXES

## Python code

```
######### utlis.py
import os
import ALL

def median_filter(df, varname = None, window=24, std=3):
    """
    A simple median filter, removes (i.e. replace by
np.nan) observations that exceed N (default = 3)
    tandard deviation from the median over window of
length P (default = 24) centered around
    each observation.
    Parameters

def fiveP_analysis(data, plot_var):
    data = data[['Temperature, daily mean (degC)',
plot_var, 'Day Type']]
    data.rename(columns={'Temperature, daily mean
(degC)':'temp', plot_var:'value'}, inplace=True)

    data_wd = data[data['Day Type']=='Working Day']
    data_nwd = data[data['Day Type']=='Non-Working Day']

    model_wd = InverseModel(data_wd['temp'].values,
data_wd['value'].values, plot_var)
    model_wd.fit_model()

    model_nwd = InverseModel(data_nwd['temp'].values,
data_nwd['value'].values, plot_var)
    model_nwd.fit_model()




data.to_csv(f'C:/Users/Compu50store/Desktop/X/{city_short
Name}_prophet.csv')
```

```
ax.set_zlabel('Daily Electricity Use [GWh]')
plt.gcf().subplots_adjust(bottom=-0.55)
#plt.savefig(generate_fig_path('Figure 11'))
```

In [ ]:

```
data_all = [la, sac, ny]
cities_shortName = ['la', 'sac', 'ny']
heatBaseTemps =
hcdh_result.loc[hcdh_result.index.get_level_values('Day
Type')=='WD']['Heating Base Temp'].values
coolBaseTemps =
hcdh_result.loc[hcdh_result.index.get_level_values('Day
Type')=='WD']['Cooling Base Temp'].values
```

In [ ]:

```
for index in range(3):
    data = data_all[index].copy()
    data = data[['Electricity demand, daily sum,
(GWh)','Temperature, daily mean (degC)',

f'HDH_{heatBaseTemps[index]}',f'CDH_{coolBaseTemps[index]
}','Day Type']]
    data.rename(columns={'Electricity demand, daily sum,
(GWh)': 'y',
                         'Temperature, daily mean
(degC)':'temp',

f'HDH_{heatBaseTemps[index]}':'HDH',

f'CDH_{coolBaseTemps[index]}':'CDH'}, inplace=True)
    data_train, data_test = prepare_data(data,
train_ratio=0.75)
    data['train'] = False
    data.loc[data_train.index,'train'] = True

        ## HCDH model
    model_wd = LinearRegression()
    X_wd = data_train_wd[['HDH','CDH']].values
    y_wd = data_train_wd['y'].values
    model_wd.fit(X_wd,y_wd)
    data_wd['yhat_hcdh'] =
model_wd.predict(data_wd[['HDH','CDH']].values)

    model_nwd = LinearRegression()
    X_nwd = data_train_nwd[['HDH','CDH']].values
    y_nwd = data_train_nwd['y'].values
    model_nwd.fit(X_nwd,y_nwd)
    data_nwd['yhat_hcdh'] =
model_nwd.predict(data_nwd[['HDH','CDH']].values)

    ## 5p model
    model_wd = InverseModel(data_train_wd['temp'].values,
data_train_wd['y'].values, 'y')
    model_wd.fit_model()
    data_wd['yhat_5p'] =
model_wd.piecewise_linear(data_wd['temp'].values,
*model_wd.p)

    model_nwd =
InverseModel(data_train_nwd['temp'].values,
data_train_nwd['y'].values, 'y')
    model_nwd.fit_model()
    data_nwd['yhat_5p'] =
model_nwd.piecewise_linear(data_nwd['temp'].values,
*model_nwd.p)

    linear_predict =
pd.concat([data_wd,data_nwd]).sort_index()
    linear_predict.index =
pd.to_datetime(linear_predict['ds'])


linear_predict.to_csv(f'C:/Users/Compu50store/Desktop/X/l
inear_{cities_shortName[index]}', index=False)
```

In [ ]:

```
plt.rc('figure', titlesize=12)
plt.rc('savefig', dpi=330, bbox='tight')
%matplotlib inline
```

In [ ]:

```
logging.getLogger('Prophet').setLevel(logging.ERROR)
```

In [ ]:

```
### section3.3 time-series model.ipynb

def prophet_analysis(region):
    # read and clean the data
    data =
    data.rename(columns={'Electricity demand, daily sum,
(GWh)': 'y'}, inplace=True)
    ## make sure data has not non:
    for field in data.columns:
        if data[field].isna().sum()>0:
            print(f'Missing entry for {field}:
{data[data[field].isna()].index}')
            data[field] = data[field].interpolate()
        else:
            print(f'No missing values in the column
{field}')
    data_train, data_test = prepare_data(data,
train_ratio=0.75)

    m_noHoliday.fit(data_train)

    future =
m_noHoliday.make_future_dataframe(periods=len(data_test),
freq='1D')
    futures = pd.concat([future, data[['HCDH']]], axis=1)
    futures.index = pd.to_datetime(futures.ds)

    forecast = m_noHoliday.predict(futures)

    f = m_noHoliday.plot_components(forecast)
    #f.savefig(generate_fig_path(f'Figure 13_{region}'))

    data_plot = make_verif(forecast, data_train,
data_test)[['y','yhat', 'train']]

    return data_plot
```

In [ ]:

```
data_train
```

In [ ]:

```
import logging
logging.getLogger('Prophet').setLevel(logging.ERROR)
```

In [ ]:

```
la_prophet = prophet_analysis('la')
sac_prophet = prophet_analysis('sac')
ny_prophet = prophet_analysis('ny')
```

In [ ]:

```
data_all = [la_prophet, sac_prophet, ny_prophet]
cities = ['Los Angeles', 'Sacramento', 'New York']
cities_shortName = ['la', 'sac', 'ny']


for index in range(3):
    data = data_all[index]
    city = cities[index]
    city_shortName = cities_shortName[index]
    # save result for model comparison
    data.rename(columns={'yhat':'yhat_prophet'},
inplace=True)

data.to_csv(f'C:/Users/Compu50store/Desktop/X/prophet_{ci
ty_shortName}.csv')

    train = data[data['train']]
    axes[index].plot(train.index, train.y, 'ko',
markersize=1.5, label='Train ground truth')
```

```
    axes[index].plot(train.index, train.yhat_prophet,
color='steelblue', lw=0.5, label='Train prediction')
    test = data[data['train'] == False]
    axes[index].plot(test.index, test.y, 'ro',
markersize=1.5, label='Test ground truth')
    axes[index].plot(test.index, test.yhat_prophet,
color='coral', lw=0.5, label='Test prediction')

    axes[index].axvline(data[data['train']].index[-1],
color='0.8', alpha=0.7)
    axes[index].set_ylabel(f'{city}\nDaily Electricity
Use [GWh]')
    axes[index].grid(ls=':', lw=0.5)

    # save result for model comparison
    data.rename(columns={'yhat':'yhat_prophet'},
inplace=True)

data.to_csv(f'C:/Users/Compu50store/Desktop/X/prophet_{ci
ty_shortName}.csv')


#plt.savefig(generate_fig_path('Figure 12'))
```

In [ ]:

```
#section3.4 tabular data model.ipynb
```

In [ ]:

```
def lightGBM_train(region, params):

    # read the data
    data =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/{region}_
    data_lgmb['Month'] = data_lgmb.index.month
    data_lgmb['dayOfWeek'] = data_lgmb.index.weekday
    data_lgmb = data_lgmb.dropna()
'Month']].values
    X_test = data_test[['Temperature, daily mean (degC)',
'Temperature, daily peak (degC)',
                        'Holiday', 'dayOfWeek',
'Month']].values
    y_train = data_train['Electricity demand, daily sum,
(GWh)'].values
    y_test = data_test['Electricity demand, daily sum,
(GWh)'].values
    X_all = data_lgmb[['Temperature, daily mean (degC)',
'Temperature, daily peak (degC)',
                       'Holiday', 'dayOfWeek',
'Month']].values

    d_train = lgb.Dataset(X_train,
categorical_feature=[2,3,4], label=y_train)

    # train and print the errors
    regr = lgb.train(params, d_train, 5000)

    rmse_train =
mean_squared_error(regr.predict(X_train), y_train)**0.5
    rmse_test = mean_squared_error(regr.predict(X_test),
y_test)**0.5
    print(f'-------City: {region}--------------')
    print(f'RMSE for Train: {rmse_train}')
    print(f'RMSE for Test: {rmse_test}')

    # make prediction on the whole data set and save the
results for model comparison
    data_lgmb['yhat_lgbm'] = regr.predict(X_all)
    data_lgmb.index = data_lgmb['ds']

data_lgmb.to_csv(f'C:/Users/Compu50store/Desktop/X/lgbm_{
region}.csv', index=False)

    # prepare the data for plotting
    verif_plot = data_lgmb[['Electricity demand, daily
sum, (GWh)','yhat_lgbm']]
    verif_plot.rename(columns={'Electricity demand, daily
sum, (GWh)':'y',
                               'yhat_lgbm':'yhat'},
inplace=True)
    verif_plot['train'] = False
    verif_plot.loc[data_train.ds, 'train'] = True

    return verif_plot
```

In [ ]:

```
print(X_all)
```

In [ ]:

```
# set up the hyper-parameter
```

In [ ]:

```
la_lgbm = lightGBM_train('la', params)
sac_lgbm = lightGBM_train('sac', params)
ny_lgbm = lightGBM_train('ny', params)
```

In [ ]:

```
data_all = [la_lgbm, sac_lgbm, ny_lgbm]
cities = ['Los Angeles', 'Sacramento', 'New York']




for index in range(3):
    data = data_all[index]
    city = cities[index]

    train = data[data['train']]
    axes[index].plot(train.index, train.y, 'ko',
markersize=1.5, label='Train ground truth')
    axes[index].plot(train.index, train.yhat,
color='steelblue', lw=0.5, label='Train prediction')
    test = data[data['train'] == False]
    axes[index].plot(test.index, test.y, 'ro',
markersize=1.5, label='Test ground truth')
    axes[index].plot(test.index, test.yhat,
color='coral', lw=0.5, label='Test prediction')
```

```
    axes[index].axvline(data[data['train']].index[-1],
color='0.8', alpha=0.7)
    axes[index].set_ylabel(f'{city}\nDaily Electricity
Use [GWh]')
    axes[index].grid(ls=':', lw=0.5)


#plt.savefig(generate_fig_path('Figure 14'))
```

In [ ]:

```
#section3.5 other ML models.ipynb
```

In [ ]:

```
def train_test_split(data_array, train_ratio=0.75):
    n_train = int(len(data_array)*train_ratio)
    data_train = data_array[:n_train,:]
    data_test = data_array[n_train:,:]
    return data_train, data_test
```

In [ ]:

```
def otherML_train(region):

    # read the data
    data =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/{region}_
    data_ml = np.concatenate((data[['Electricity demand,
daily sum, (GWh)',
                                    'Temperature, daily
mean (degC)',
                                    'Temperature, daily
peak (degC)']].values,

data[['Holiday']].astype('int').values,

pd.get_dummies(data.index.weekday).values,

pd.get_dummies(data.index.month).values), axis=1)

    # train the RF model
    regr_rf = RandomForestRegressor(max_depth=8,



    rmse_train_rf =
mean_squared_error(regr_rf.predict(X_train),
y_train)**0.5
    rmse_test_rf =
mean_squared_error(regr_rf.predict(X_test), y_test)**0.5
    print(f'-------Random Forest: {region}--------------
')
    print(f'RMSE for Train: {rmse_train_rf}')
    print(f'RMSE for Test: {rmse_test_rf}')

    # train the SVM model

mean_squared_error(regr_svr.predict(X_train),
y_train)**0.5
    rmse_test_svr =
mean_squared_error(regr_svr.predict(X_test), y_test)**0.5
    print(f'-------SVM: {region}--------------')
    print(f'RMSE for Train: {rmse_train_svr}')
    print(f'RMSE for Test: {rmse_test_svr}')


    regr_nn.fit(X_train, y_train)
    rmse_train_nn =
mean_squared_error(regr_nn.predict(X_train),
y_train)**0.5
    rmse_test_nn =
mean_squared_error(regr_nn.predict(X_test), y_test)**0.5
    print(f'-------Neural Network: {region}--------------
')
    print(f'RMSE for Train: {rmse_train_nn}')
    print(f'RMSE for Test: {rmse_test_nn}')

    # make prediction on the whole data set and save the
results for model comparison
```

```
    data['yhat_rf'] = regr_rf.predict(X_all)
    data['yhat_svm'] = regr_svr.predict(X_all)
    data['yhat_nn'] = regr_nn.predict(X_all)

data.to_csv(f'C:/Users/Compu50store/Desktop/X/otherML_{re
gion}.csv')

    # prepare the data for plotting
    verif_plot = data[['Electricity demand, daily sum,
(GWh)','yhat_rf','yhat_svm','yhat_nn']]
    verif_plot.rename(columns={'Electricity demand, daily
sum, (GWh)':'y'}, inplace=True)
    verif_plot['train'] = False
    verif_plot.iloc[:len(data_train),:]['train'] = True

    return verif_plot
```

In [ ]:

```
X_train
```

In [ ]:

```
X_train
```

In [ ]:

```
la_otherML = otherML_train('la')
sac_otherML = otherML_train('sac')
ny_otherML = otherML_train('ny')
```

In [ ]:

```
data_all = [la_otherML, sac_otherML, ny_otherML]
cities = ['Los Angeles', 'Sacramento', 'New York']




for index in range(3):
    data = data_all[index]
    city = cities[index]

    train = data[data['train']]
    axes[index].plot(train.index, train.y, 'ko',
markersize=1.5, label='Train ground truth')
    axes[index].plot(train.index, train.yhat_rf,
color='steelblue', lw=0.5, label='Train prediction')
    test = data[data['train'] == False]
    axes[index].plot(test.index, test.y, 'ro',
markersize=1.5, label='Test ground truth')
    axes[index].plot(test.index, test.yhat_rf,
color='coral', lw=0.5, label='Test prediction')

    axes[index].axvline(data[data['train']].index[-1],
color='0.8', alpha=0.7)
    axes[index].set_ylabel(f'{city}\nDaily Electricity
Use [GWh]')
    axes[index].grid(ls=':', lw=0.5)


#plt.savefig(generate_fig_path('Figure A1_rf'))
```

In [ ]:

```
data_all = [la_otherML, sac_otherML, ny_otherML]
cities = ['Los Angeles', 'Sacramento', 'New York']



for index in range(3):
    data = data_all[index]
    city = cities[index]

    train = data[data['train']]
    axes[index].plot(train.index, train.y, 'ko',
markersize=1.5, label='Train ground truth')
    axes[index].plot(train.index, train.yhat_svm,
color='steelblue', lw=0.5, label='Train prediction')
    test = data[data['train'] == False]
    axes[index].plot(test.index, test.y, 'ro',
markersize=1.5, label='Test ground truth')
```

```
    axes[index].plot(test.index, test.yhat_svm,
color='coral', lw=0.5, label='Test prediction')

    axes[index].axvline(data[data['train']].index[-1],
color='0.8', alpha=0.7)
    axes[index].set_ylabel(f'{city}\nDaily Electricity
Use [GWh]')
    axes[index].grid(ls=':', lw=0.5)

#plt.savefig(generate_fig_path('Figure A2_svm'))
```

In [ ]:

```
data_all = [la_otherML, sac_otherML, ny_otherML]
cities = ['Los Angeles', 'Sacramento', 'New York']


for index in range(3):
    data = data_all[index]
    city = cities[index]

    train = data[data['train']]
    axes[index].plot(train.index, train.y, 'ko',
markersize=1.5, label='Train ground truth')
    axes[index].plot(train.index, train.yhat_nn,
color='steelblue', lw=0.5, label='Train prediction')
    test = data[data['train'] == False]
    axes[index].plot(test.index, test.y, 'ro',
markersize=1.5, label='Test ground truth')
    axes[index].plot(test.index, test.yhat_nn,
color='coral', lw=0.5, label='Test prediction')

    axes[index].axvline(data[data['train']].index[-1],
color='0.8', alpha=0.7)
    axes[index].set_ylabel(f'{city}\nDaily Electricity
Use [GWh]')
    axes[index].grid(ls=':', lw=0.5)
```

```
#plt.savefig(generate_fig_path('Figure A3_nn'))
```

In [ ]:

```
#section4.1 model comparison.ipynb
# some matrics and stats
```

In [ ]:

```
def combine_predict(region):
    lgbm_predict =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/lgbm_{regio
n}.csv',index_col=0)[['yhat_lgbm']]
    otherML_predict =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/otherML_{re
gion}.csv',index_col=0)[['yhat_rf','yhat_svm','yhat_nn']]
    linear_predict =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/linear_{reg
ion}.csv',index_col=0)[['y','yhat_hcdh','yhat_5p']]
    prophet_predict =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/prophet_{re
gion}.csv',index_col=0)[['yhat_prophet','train']]

    predict = pd.concat([lgbm_predict, otherML_predict,
linear_predict, prophet_predict], axis=1)
    predict.index = pd.to_datetime(predict.index)

    predict.rename(columns={'y':'Ground Truth',
                            'yhat_5p':'5-Parameter',
                            'yhat_hcdh':'Degree Hour',
                            'yhat_prophet':'Decomposed',
                            'yhat_lgbm':'lightGBM',
                            'yhat_rf':'Random Forest',
                            'yhat_svm':'Support Vector
Machine',
                             'yhat_nn':'Neural Network'},
inplace=True)
    return predict
```

In [ ]:

```
la_predict = combine_predict('la')
sac_predict = combine_predict('sac')
ny_predict = combine_predict('ny')
```

In [ ]:

```
def calculate_error(predict):
    accCom =
pd.DataFrame(columns=['R2_train','R2_test','MAE_train',


    accCom['R2_train'] =
predict[predict['train']].loc[:,['Ground Truth','5-
Parameter','Degree Hour','Decomposed',
                        'lightGBM','Random
Forest','Support Vector Machine','Neural
Network']].corr().iloc[0,1:]
    accCom['R2_test'] =
predict[~predict['train']].loc[:,['Ground Truth','5-
Parameter','Degree Hour','Decomposed',
                        'lightGBM','Random


    train_real = predict[predict['train']]['Ground
Truth'].values
    test_real = predict[~predict['train']]['Ground
Truth'].values
    for predict_field in accCom.index:
        train_predict =
predict[predict['train']][predict_field].values
        test_predict =
predict[~predict['train']][predict_field].values
        accCom.loc[predict_field,'MAE_train'] =
mean_absolute_error(train_real, train_predict)
```

```
        accCom.loc[predict_field,'MAE_test'] =
mean_absolute_error(test_real, test_predict)
        accCom.loc[predict_field,'RMSE_train'] =
mean_squared_error(train_real, train_predict)**0.5
        accCom.loc[predict_field,'CVRMSE_train'] =
accCom.loc[predict_field,'RMSE_train']/predict['Ground
Truth'].mean()
        accCom.loc[predict_field,'RMSE_test'] =
mean_squared_error(test_real, test_predict)**0.5
        accCom.loc[predict_field,'CVRMSE_test'] =
accCom.loc[predict_field,'RMSE_test']/predict['Ground
Truth'].mean()

    accCom.drop(['R2_train','R2_test'], axis=1,
inplace=True)

    return accCom
```

In [ ]:

```
la_error = calculate_error(la_predict)
sac_error = calculate_error(sac_predict)
ny_error = calculate_error(ny_predict)
```

In [ ]:

```
error_result = pd.concat([la_error, sac_error, ny_error],
keys=cities)
error_result.to_csv('C:/Users/Compu50store/Desktop/X/Tabl
e 3.csv')
```

In [ ]:

```
error_result
```

In [ ]:

```
figsize=(9,5))




'Sacramento', 'New York']):
    labels = error_result.loc[region,:].index
    CVRMSE_train =
error_result.loc[region,'CVRMSE_train']
```

```
#plt.savefig(generate_fig_path('Figure 16'))
```

In [ ]:

```
plot_time = {'summer':['2018-07-01','2018-10-01'],
'winter':['2018-12-31','2019-04-01']}

la_predict_plot = la_predict.drop(['train'], axis=1)
sac_predict_plot = sac_predict.drop(['train'], axis=1)
ny_predict_plot = ny_predict.drop(['train'], axis=1)

data_all =
[la_predict_plot,sac_predict_plot,ny_predict_plot]
cities = ['Los Angeles', 'Sacramento', 'New York']
```

In [ ]:

```
import matplotlib.dates as mdates
```

In [ ]:

```
myFmt = mdates.DateFormatter('%m-%d')

for index in range(3):
    data = data_all[index]
    city = cities[index]
    data_summer =
data.truncate(before=plot_time['summer'][0],
after=plot_time['summer'][1])
    data_winter =
data.truncate(before=plot_time['winter'][0],
after=plot_time['winter'][1])

    axes[index, 0].plot(data_summer.index,
data_summer['Ground Truth'].values, label='Ground Truth',
linewidth=2)
    axes[index, 0].plot(data_summer.index,
data_summer['5-Parameter'].values, '--', label='5-
Parameter')
    axes[index, 0].plot(data_summer.index,
data_summer['Degree Hour'].values, '--', label='HCDH')
    axes[index, 0].plot(data_summer.index,
data_summer['Decomposed'].values, ':',
label='Decomposed')
    axes[index, 0].plot(data_summer.index,
data_summer['lightGBM'].values, ':', label='GBM')
    axes[index, 0].xaxis.set_major_formatter(myFmt)

    axes[index, 1].plot(data_winter.index,
data_winter['Ground Truth'].values, label='Ground Truth',
linewidth=2)
    axes[index, 1].plot(data_winter.index,
data_winter['5-Parameter'].values, '--', label='5-
Parameter')
    axes[index, 1].plot(data_winter.index,
data_winter['Degree Hour'].values, '--', label='HCDH')
    axes[index, 1].plot(data_winter.index,
data_winter['Decomposed'].values, ':',
label='Decomposed')
    axes[index, 1].plot(data_winter.index,
data_winter['lightGBM'].values, ':', label='GBM')
    axes[index, 1].xaxis.set_major_formatter(myFmt)

    axes[index,
0].set_ylabel(f'C:/Users/Compu50store/Desktop/X/{city}\nD
aily Electricity Use\n[GWh]')
```

In [ ]:

```
title = {'la': 'Los Angeles', 'sac':'Sacramento',
'ny':'New York'}
regions = ['la','sac','ny']

temp_plot = np.arange(20, 45, 0.5)

data_all = {}
predict_sum_all = {}
predict_peak_all = {}

for index in range(3):

    region = regions[index]
    data =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/{region}_da
ily.csv', index_col=0)

    # train 5p model
    data_wd = data[data['Day Type']=='Working Day']
    model_sum = InverseModel(data_wd['Temperature, daily
mean (degC)'].values,
                             data_wd['Electricity demand,
daily sum, (GWh)'].values, 'Electricity demand, daily
sum, (GWh)')
    model_sum.fit_model()
    predict_sum_5p = piecewise_linear(temp_plot,
*model_sum.model_p)

    model_peak = InverseModel(data_wd['Temperature, daily
mean (degC)'].values,
                              data_wd['Electricity power,
daily peak, (GW)'].values, 'Electricity power, daily
peak, (GW)')
    model_peak.fit_model()
    predict_peak_5p = piecewise_linear(temp_plot,
*model_peak.model_p)
    # train lightGBM model


    data['dayOfWeek'] = data.index.weekday
    data = data.dropna()
    X_train = data[['Temperature, daily mean (degC)',
'Non-workDay', 'Month']].values
    y_sum_train = data['Electricity demand, daily sum,
(GWh)'].values
    y_peak_train = data['Electricity power, daily peak,
(GW)'].values
    d_sum_train = lgb.Dataset(X_train,
categorical_feature=[1,2], label=y_sum_train)
    d_peak_train = lgb.Dataset(X_train,
categorical_feature=[1,2], label=y_peak_train)
    clf_sum = lgb.train(params, d_sum_train, 5000)
    clf_peak = lgb.train(params, d_peak_train, 5000)
    X_test_temp = pd.DataFrame({'Temperature, daily mean
(degC)':temp_plot})
    X_test_temp['Non-workDay'] = False
    X_test_temp['Month'] = 8
    predict_sum_lgmb =
clf_sum.predict(X_test_temp.values)
    predict_peak_lgmb =
clf_peak.predict(X_test_temp.values)

    predict_sum = pd.DataFrame({'5-parameter':
predict_sum_5p, 'lightGBM': predict_sum_lgmb},
index=temp_plot)
    predict_peak = pd.DataFrame({'5-parameter':
predict_peak_5p, 'lightGBM': predict_peak_lgmb},
index=temp_plot)

    data_all[region] = data_wd
    predict_sum_all[region] = predict_sum
    predict_peak_all[region] = predict_peak
```

In [ ]:

```
    region = regions[index]
    data = data_all[region]
    predict_sum = predict_sum_all[region]
    predict_peak = predict_peak_all[region]
```

```
parameter'].values, label='5-parameter',
color=default_colors[0])
    axes[0, index].plot(temp_plot,
predict_sum['lightGBM'].values, label='GBM',
color=default_colors[1])
    axes[0, index].scatter(data['Temperature, daily mean
(degC)'].values,
                           data['Electricity demand,
daily sum, (GWh)'].values, label='Measured Data',
                           s=1, color=default_colors[2])

    axes[0, index].set_xlim(temp_plot[0], temp_plot[-1])

    axes[1, index].plot(temp_plot, predict_peak['5-
parameter'].values, label='5-parameter',
color=default_colors[0])
    axes[1, index].plot(temp_plot,
predict_peak['lightGBM'].values, label='GBM',
color=default_colors[1])
    axes[1, index].scatter(data['Temperature, daily mean
(degC)'].values,
                           data['Electricity power, daily
peak, (GW)'].values, label='Measured Data',
                           s=1, color=default_colors[2])
    axes[1, index].set_xlabel('Daily Mean Temperature
[$^oC$]')
    axes[1, index].set_xlim(temp_plot[0], temp_plot[-1])
axes[0, 0].set_ylabel('Daily Electricity Use [GWh]')
axes[1, 0].set_ylabel('Daily Peak Demand [GW]')



#plt.savefig(generate_fig_path('Figure 17'))
```

In [ ]:

```
# COVID

# set up the hyper-parameter

title = {'la': 'Los Angeles', 'sac':'Sacramento',
'ny':'New York'}
regions = ['la','sac','ny']

result_all = {}
```

In [ ]:

```
for index in range(3):
    region = regions[index]

    data =
pd.read_csv(f'C:/Users/Compu50store/Desktop/X/{region}_




'Temperature, daily peak (degC)', 'Holiday', 'Weekend',
                      'Electricity demand, daily sum,
(GWh)']].copy()
    data_lgmb['Month'] = data_lgmb.index.month
    data_lgmb['dayOfWeek'] = data_lgmb.index.weekday
    data_lgmb = data_lgmb.dropna()

    # train the model
    data_train, data_test = prepare_data(data_lgmb,
train_ratio=0.9)
    X_train = data_train[['Temperature, daily mean
(degC)', 'Temperature, daily peak (degC)', 'Holiday',
'dayOfWeek', 'Month']].values
    X_test = data_test[['Temperature, daily mean (degC)',
'Temperature, daily peak (degC)', 'Holiday', 'dayOfWeek',
'Month']].values
    y_train = data_train['Electricity demand, daily sum,
(GWh)'].values
    y_test = data_test['Electricity demand, daily sum,
(GWh)'].values




    # make prediction
    X_all = data_lgmb[['Temperature, daily mean (degC)',
'Temperature, daily peak (degC)', 'Holiday', 'dayOfWeek',
'Month']].values
    data_lgmb['yhat_lgbm'] = clf.predict(X_all)
    data_lgmb.index = data_lgmb['ds']

    result = data_lgmb[['Electricity demand, daily sum,
(GWh)','yhat_lgbm']].copy()
```

```
    result.rename(columns={'Electricity demand, daily
sum, (GWh)':'y',
                                'yhat_lgbm':'yhat'},
inplace=True)
    result['train'] = False
    result.loc[data_train.ds, 'train'] = True

    result_all[region] = result
```

In [ ]:

```
cities = ['Los Angeles', 'Sacramento', 'New York']
```

In [ ]:

```
monthly_all = {}

for index in range(3):
    region = regions[index]
    data = result_all[region]
    city = cities[index]

    verif = data.truncate(before='2020-01-01')

    axes[index].plot(verif.index, verif.y,
color=default_colors[1], linestyle='--', lw=2,
label='Ground truth')
    train = verif[verif['train']]
    axes[index].plot(train.index, train.yhat,
color=default_colors[0], lw=2, label='Prediction before
Lockdown')
    test = verif[verif['train'] == False]
    axes[index].plot(test.index, test.yhat,
color=default_colors[2], lw=2, label='Prediction after
Lockdown')
    axes[index].axvline(verif[verif['train']].index[-1],
color='black', alpha=0.5, label = 'Lockdown')

    axes[index].set_ylabel(f'{city}\nDaily Electricity
Use [GWh]')
    axes[index].grid(ls=':', lw=0.5)

train.to_csv(f'C:/Users/Compu50store/Desktop/{city}_train
.csv')
```

```
test.to_csv(f'C:/Users/Compu50store/Desktop/{city}_test.csv')
    # summary of monthly reduction
    verif['month'] = verif.index.month
    monthly = verif.groupby('month').sum()
    monthly['percent'] = monthly['y']/monthly['yhat']
    monthly_all[region] = monthly.percent.values



axes[2].set_xlabel('Time')

#plt.savefig(generate_fig_path('Figure 18'))
```

In [ ]:

```
test.to_csv(f'C:/Users/Compu50store/Desktop/F1.csv')
```

In [ ]:

```
import matplotlib.ticker as mtick
```

In [ ]:

```
monthly_all_df = pd.DataFrame(monthly_all)-1
monthly_all_df.columns = cities
monthly_covid = monthly_all_df.iloc[2:,:]





ax.set_ylabel('Electricity Use Change \nSince COVID-19 Lockdown')
#plt.savefig(generate_fig_path('Figure 19'))
```

In [ ]:

# RESUME

## Personal Information

Surname, name : Saif Mohammed Salamn Al-azzawi

Nationality : Iraqi

## Education

| Degree | Education Unit | Graduation Date |
|---|---|---|
| Master | Electrical – Electronic Engineering | 2023 |
| Bachelor | Power & Electrieal Machines | 2006 |
| High School | Alshareef Alrady Secondary School | 2002 |

## Work Experience

| Year | Place | Title |
|---|---|---|
| 2008 Till now | Presidency of Diyala University | Maintenance and development department |

## Foreing Language:

Arabic and English

## Publications:

Hyperparameter Optimization of regression model for Electrical Load Forecasting during the COVID-19 Pandemic lockdown period

## Hobbies:

Growing plants, reading and fishing