TIME SERIES MODELLING AND FORECASTING APPROACH FOR ENERGY PRICE AND CONSUMPTION PREDICTION

A MAJOR PROJECT REPORT

Submitted by

G.V. SAI VARUN	**[RA1811003020102]**
M. SATHVIK	**[RA1811003020101]**
K. YASWANTH KUMAR	**[RA1811003020072]**

Under the guidance of

DR. S. LILLY SHEEBA

(Associate Professor, Department of Computer Science and Engineering)

in fulfillment for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

RAMAPURAM

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM CAMPUS, CHENNAI -600089

MAY 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"TIME SERIES MODELLING AND FORECASTING APPROACH FOR ENERGY PRICE AND CONSUMPTION PREDICTION"** is the bonafide work of **G.V. SAI VARUN [RA1811003020102], M.SATHVIK [RA1811003020101], K.YASWANTH KUMAR[RA1811003020072]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE

DR. S. LILLY SHEEBA
Associate Professor,
Computer Science and Engineering,
SRM Institute of Science and
Technology, Ramapuram Campus,
Chennai.

SIGNATURE

Dr. K.RAJA,M.E., Ph.D.,
Professor and Head,
Computer Science and Engineering,
SRM Institute of Science and
Technology, Ramapuram Campus,
Chennai.

Submitted for the project viva-voce held on........................... at SRM Institute of Science and Technology , Ramapuram Campus, Chennai -600089.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM, CHENNAI – 89

DECLARATION

We hereby declare that the entire work contained in this project report titled "**TIME SERIES MODELLING AND FORECASTING APPROACH FOR ENERGY PRICE AND CONSUMPTION PREDICTION**" has been carried out by **G.V. SAI VARUN [RA1811003020102], M.SATHVIK [RA1811003020101], K.YASWANTH KUMAR[RA1811003020072]** at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **DR. S. LILLY SHEEBA, Associate Professor**, Department of Computer Science and Engineering.

Place: Chennai

Date:

G.V. SAI VARUN
M. SATHVIK
K. YASWANTH KUMAR

Own Work Declaration

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course :B. TECH

Student Name :G.V.Sai Varun, M.Sathvik, K.Yaswanth Kumar

Registration Number :RA1811003020102, RA1811003020101, RA1811003020072

Title of Work :Time Series Modelling And Forcasting Approach For Energy Price And Consumption Prediction

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group

RA1811003020102 RA1811003020101

RA1811003020072

ABSTRACT

Accurate energy charge forecasting is one of the most important aspects of decision-making for power market participants who want to develop cost-effective competitive strategies. In order to save power, it is necessary to predict the charge and use of electricity. Because assist vector regression has a good overall performance in dealing with non-linear statistics regression problems, it is often utilized to forecast power charge and intake in recent years. We conclude that the relationship between light electrical charge and intake, as well as its affecting factors, is non-linear, based on ancient statistics. For power purchasing and selling organizations, forecasting the cost and use of electricity is critical. Because the precision of the forecast translates directly into the profit of the organization, it must be as accurate as possible. One of the most important ventures in electricity gadget operation and planning is the forecasting of electrical masses and electricity charge and intake. However, in a few cases, we wish to tackle this problem in the absence of reliable and acceptable historical data. The help vector regression with radial foundation feature was used to expand the accurate prediction version of lights electricity charge and intake. The forecast effects show that assist vector regression has a higher prediction accuracy than neural networks. The prediction version can estimate the building's hourly power charge and consumption, as well as assess the impact of workplace building electricity management programmes.

ACKNOWLEDGEMENT

We place on record our deep sense of gratitude to our lionized Chairman **Dr. R. SHIVAKUMAR** for providing us with the requisite infrastructure throughout the course.

We take the opportunity to extend our hearty and sincere thanks to our Dean, **Dr. M. MURALI KRISHNA, B.E., M.Tech., Ph.D. MISTE, FIE, C.Engg.,** for maneuvering us into accomplishing the project.

We take the privilege to extend our hearty and sincere gratitude to the Professor and Head of the Department, **Dr. K. RAJA, M.E., PhD.,** for his suggestions, support and encouragement towards the completion of the project with perfection.

We express our hearty and sincere thanks to our guide **Dr. S. LILLY SHEEBA, Associate Professor,** Computer Science and Engineering Department for her encouragement, consecutive criticism and constant guidance throughout this project work.

Our thanks to the teaching and non-teaching staff of the Computer Science and Engineering Department of SRM Institute of Science and Technology, Ramapuram Campus, for providing necessary resources for our project.

G.V. SAI VARUN

M. SATHVIK

K. YASWANTH KUMAR

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

S.NO	ABBREVATIONS	EXPANSION
1	LSTM	Long Short-Term Memory
2	AM	Ambiguity Measure
3	ML	Machine Learning
4	BSA	Backtracking Search Algorithm
5	ANFIS	Adaptive Neuro-Fuzzy Inference System
6	UML	Unified Modeling Language
7	NLP	Natural Language Processing
8	IR	Information Retrieval
9	RNN	Recurrent Neural Network
10	RAD	Rapid Application Development
11	TDD	Test Driven Development
12	OS-ELM	Online Sequential Extreme Learning Machine

CHAPTER 1

INTRODUCTION

1.1 Overview

This paper discusses about consumer energy price forecasting using machine learning with a bi-directional LSTM algorithm. By taking this scenario into consideration, there are many large-scale industries in public and private business unit sectors that are deprived of energy price as these units will have to consider and segregate the expenditure. As we know, electricity rates are non-linear, that is the price may increase or decrease unevenly. By taking this fact into consideration, we need to learn and analyze the data to expect an amount in the future.

Because the amount of data on pricing and consumption is so big, the typical individual will be unable to compute an accurate conclusion. In this instance, machine learning plays a critical role. There are numerous algorithms or approaches for various difficulties in this kind of circumstance. LSTM and data analysis In order to find the most accurate result, LSTM is applied. The rationale for picking this algorithm is because the previous ones had flaws at some point. According to my study and expertise, LSTM is the greatest solution for these kinds of situations.

This system works in a definite process, so there is no need for human intervention. A set of rules helps us to get a precise outcome. Going through the process Firstly, input data is given to the system, collected from various sources in the form of a csv file.

Second, this data is moved on to the next step, which is called data analysis. This step removes unwanted or incomplete relationships from the data. Data preprocessing is a continuing step, including NLP and IR. Feature Extraction includes Ambiguity Measure (AM). The data set is then split into two parts: a training dataset and a testing dataset. Finally, go to the LSTM prediction machine, which is followed by the result.

1.2 Proposed System

Despite the fact that each user's profile is unique, some combinations of profiles appear often in data on electrical load. Using the clients' daily consumption habits and their periodic use patterns, this system is able to provide personalized service.

In order to train an LSTM to find patterns, the price and consumption load data are transformed into a set of predefined profiles. As the learning process progresses, the embedding vectors, which represent the profiles' attributes and connections, are learned. A pattern in the profile sequence is detected and used by the LSTM to predict future consumption based on the embedded vectors that have been learned so far.

Using daily consumption profile sequences and an LSTM network, this system predicts short-term power demand trends. The profile illustrates the wide variety of everyday electricity demands. In order to create a representative profile for each day, the daily load data is divided and the price and consumption patterns for each day are mapped. For the month-ahead profile prediction, the LSTM employs profile sequences instead of minute or hour-based load data. Instead of trying to estimate how much load would be generated at a certain point in time, the proposed approach looks for patterns in the daily load profile sequences.

1.3 Objective

Human decision-making is made easier by machine learning, which is the most extensively utilized technology for predicting the future or classifying data. Machine Learning algorithms are trained on examples or instances, from which they learn from prior experiences and analyze historical data. This ability to recognize patterns and predict the future comes from repetition of the examples, as it trains over and over.

It is vital to estimate the charge and consumption of electricity in order to save energy. Assist vector regression is frequently used to anticipate power charge and intake in recent years because it has an excellent overall performance in dealing with non-linear statistics regression situations. Based on ancient statistics, we determine that the relationship between light electrical charge and intake, as well as its influencing elements, is non-linear.

However, simply creating models is insufficient. You must also properly optimise and tweak the model in order for it to produce correct results. Optimization strategies include fine-tuning hyperparameters to get the best possible outcome.

The accurate prediction version of lights electricity charge and intake was expanded using aid vector regression with radial foundation feature. As a result of the forecast effects, assist vector regression outperforms neural networks in terms of prediction accuracy. The prediction version may calculate the hourly power charge and consumption of the building, as well as evaluate the impact of workplace building electricity management programmes.

1.4 Scope of Project

The main contributions of this project are:

1) Data Visualization
2) Data Preprocessing
3) Training the Classifier
4) Testing the Classifier

1.5 Overview of the Report

The report is organized in the manner as follows

- **Chapter 1**: This chapter tells us about the introduction of the whole project and also about the proposed system that we are going to execute in this project.

- **Chapter 2**: This chapter gives information about the literature survey.

- **Chapter 3**: This chapter tells us about architecture diagram of the project and also the hard ware and software requirements to run the code.

- **Chapter 4**: This chapter says about the methods, modules of the project .
- **Chapter 5**: This chapter shows the sample code and implementation screenshots.
- **Chapter 6**: This chapter gives information about the testing of the project.
- **Chapter 7**: This chapter tells us about the result analysis.
- **Chapter 8**: This chapter serves as a wrap-up and a vision for the project’s future.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

During the last several years, there has been a dramatic change in technology and how it is utilized in daily life. Although deep learning and machine learning are being hailed as the technology behind our most recent technological achievements, such as robotics and search engines, many are left wondering what really divides these two methods.

As a subset of artificial intelligence, machine learning and deep learning use cutting-edge techniques to perform cognitive functions we associate with intuitive learning. Each application is distinct and offers a variety of benefits to the end user, whether it's assisting with speech/facial recognition, speeding up web applications, or solving problems unique to a particular business case. There has been a substantial advancement in both machine learning and deep learning since the 1960s, but the distance between the two has widened.

It is possible for a computer to learn and, in effect, train itself to develop when it is exposed to fresh and continuously changing data via the use of machine learning. Using machine learning, Facebook's news feed tailors each user's feed to their interests. Analytical techniques such as regression analysis and forecasting are essential in traditional machine learning software because they help reveal patterns and hidden insights in previously calculated data without explicitly teaching the programme where to search.

It's amazing how much more capable machine learning has become in sorting through massive amounts of data. Netflix and social media algorithms that alert users to popular subjects leverage ML. There are many similarities between machine learning and deep learning, but there are significant distinctions as well.

Extraction of features a programmer is needed to instruct computer's answer to the question information it should search for in order to make a judgement, which is a time-consuming procedure in machine learning. Human mistake in the programming process reduces the accuracy of machine learning.

2.2 Existing System

When it comes to pricing in the electricity market, pre-dispatch forecasts are critical. When it comes to power system conditions, however, such forecasts are often based on traditional batch-learning algorithms, which are unable to react quickly enough to changes in the local environment. Aside from that, the pre dispatch local pricing projection is regularly impacted by dynamic price changes from nearby sites. To solve these issues, this study employs an online sequential extreme learning machine (OS-ELM) method to give an improved pre dispatch price prediction. The article provides a unique a two-dimensional orthogonal list data structure, as well as two OS-ELM modules to go with it. Rolling day-ahead price prediction and prediction intervals are provided by one module using the daily online training update, while the other uses the 2-h-by-2-h online training update to provide rolling 30-min prediction. In nonlinear patterns, the recommended technique may constantly identify any unexpected occurrences and price variations from nearby areas. Simulation tests According to statistics from the Australian energy sector market are used to verify the recommended strategy, and the simulation outcomes reveal that the proposed technique may assist increase prediction accuracy. This research presents an online-learning ELM algorithm that improves pre-distribution power price forecasts. – A new data structure may be calculated using the suggested approach as follows: To put it more simply, here are some of the numbers you'll need to know to get the most out of this article (cid:26) (cid (cid:27) (cid:27) When the standard error of the measurement is equal to or more than one standard deviation, then the measurement is considered accurate (cid:2) [yt yt]2 1870, where yt is the prediction and yt is the N is the sample size, and t is the time point at which the result is to be compared. The authors would like to appreciate Australia's NEM-Watch software for their contributions for its cooperation with this study, as well as Dr. Ahmed Saber, Manager of ETAP's Smart Power System Optimization Dept. (T&D) for his helpful recommendations.

2.3 Issues in Existing System

- It is not an easy-to-use method
- Maximizes the complexity of the problem
- This system is opportunistic and uncontrollable
- Cannot restrain behavior pattern in complex systems
- Tedious message updating
- Human labor is required for manual annotation.
- Difficult and less commonly used

2.4 Literature Survey

In [1] a forecasting approach is provided in this research to incorporate accurate pricing projections for behind-the-meter storage system operation. This technique employs two distinct forecasting models that combine high-resolution market data with hourly data to capture price spikes to the greatest extent possible. The hourly projections are updated using the proposed intra-hour rolling horizon approach. According to statistical research, the proposed technique improves forecast accuracy by 20% over existing PDPs and has a good potential of detecting price spikes.

Adaptation of systems by end users is impacted by dynamic pricing schemes [2]. Furthermore, dynamic pricing may be considered as opportunities to maximize profit while minimizing expense at each and every rung of the electrical ladder. Energy storage devices operated by utilities boost retail gain equally, and as the degree of integration improves, the share of profits that consumers get increases as well. As utilities interest in large-scale energy storage systems rises, the pricing schemes will become smoother thanks to the advent of location-based marginal pricing schemes. 500 MWh x 1 GWh is 500 MWh × 1 GWh. Reprints of this article can be found in Sha Vehicular Technology 67.7 (2018): 5671-5682. Management of a stochastic model predictive plug-in hybrid electric bus energy management that is time-efficient. An artificial neural network-enhanced energy management solution for plug-in hybrid electric cars has been developed.

Results of utilizing neural networks to estimate electric energy prices, both one-step and n-step ahead in the case of recurrent neural networks, are reported in [3]. In addition, the well-known Lyapunov approach is used to illustrate the stability of the an expanded Kalman filter approach is used to train the suggested artificial neural network. Predictions, based on European power system data are being evaluated for this purpose, where one-step and n-step forward predictions are used.

For the purpose of this research [4] , it is important to establish in the likely rather than a single price for electricity estimate based on the uncertainty in the major factors that influence electricity pricing. It is also possible to get price estimates and an electricity generating portfolio with regard to market participants and main energy sources at the same time. A market variable's impact on electricity prices may be estimated using the model. The created approach is tested using data from the real world. For medium-term forecasting tasks, this work develops an electrical market model that considers dynamics of supply and demand and simulates TEM under particular parameters. The journal Transactions on Power Systems deals with power systems. When predicting hourly prices for annual pricing averages and supply and demand situations are available integrated, price ranges are generated rather than single point forecasts. Three primary components and a sub-model of the energy market model are built to this end: electricity demand, supply, price forecasting models and hydro optimization sub-model. This research has a unique structure since all of the models are employed in the same simulation.

A hybrid model [5] is developed in this study to forecast electricity prices, taking into account both typical price changes and price spikes. While the ARXTV model can adequately estimate regular prices, it cannot effectively track price surges. By combining regression, this restriction was overcome in the ARXTV with kernel hybrid model. The K R-based approach focuses solely on estimating price spikes, allowing for more accurate forecasting. As a result, combining the two models improves overall price forecasting in competitive electricity markets that are very variable. The mutual information strategy is used to identify the significant inputs in both ARXTV and KR based models. It also aids in lowering the total computing burden and enhancing the model's robustness of performance. With a three-level wavelet approach, the ARXTV model's robustness is significantly enhanced.

As a forecasting engine, [6] proposes an improved adaptive neuro-fuzzy inference system (ANFIS) and a two-stage feature selection approach as a hybrid technique for predicting power prices to reliably anticipate energy prices. Using MOBBSA as an effective means of carrying out an evolutionary hunt approach, and the ANFIS method to identify the most influential input variable subsets with maximum relevance and lowest redundancy, this study aims to better understand the relationship between multiple variables. The backtracking search algorithm (BSA) has been included into the ANFIS technique the accuracy of the data has to be improved prediction of the electricity price. Using coupled with one another multiple objective feature selection strategy and a hybrid forecast engine, this research offered an approach for predicting day-ahead power prices in the Ontario electricity market that encompassed both.

A stochastic process is realized as a functional time series in which each observation is a continuous function defined over a certain time period. More and more energy markets are adopting continuous-time marginal pricing models, and these processes are becoming more significant as a result. A continuous function model is needed to predict these time series. With this novel functional forecasting technique for the L Hilbert space, we want to extend the fundamental seasonal ARIMAX time series model [7]. Functional parameters operate on functional variables in the proposed model, which is constructed as a linear regression. Variables might be autoregressive terms, moving average terms (already observed innovations), or exogenous variables. An integral operator with a kernel made up of linear sigmoid functions is used in this method. A Quasi-Newton technique that minimizes the sum of squared errors is used to optimize each sigmoid's parameters. Functional time series models may benefit from this innovative approach to moving average estimation. Using data from the Spanish and German electricity markets, the new model is compared to previous functional reference models.

It is becoming more vital for market players to have accurate power price forecasts as a result of the liberalization of the electric energy industry. management. Although, due because to the high degree of turbulence and intricate electrical power is unequal pricing, it remains a difficult task. A unique a day-ahead hybrid deep learning framework [8] is suggested to address this, and it features preprocessing, deep learning-based point prediction, error correction, and statistical prediction are all included. The least total contraction and choice activator in a rejection forest are helped in the feature preprocessing module to find Anomalies and some linked aspects of the power exchange rate. In this study, a novel integrated underground architecture for all day power estimation is described, Quantization, broad standardized components identification and mistake correction and stochastic classification modules are included. Outliers are detected and linked aspects of electricity price series are identified using the first modules. In the second module, three deep learning models are suggested Long short - term memory and Convolutional neural network were evaluated to obtain tough dynamic behavior. The third segment may assist to lessen the discrepancies among projections and reality prices. Bayesian estimation estimators are used in the develop holistic to highlight the ambiguity problem at various accuracy.

The main goal of [9] is to predict and anticipate time series of utility prices. Determined and random elements of the value current period are segregated to accomplish this, using components estimate approach. Lengthy patterns of behavior, multiple times, also date influences make up the deterministic component, whereas the brief characteristics of the system are accounted for by the random factor.

CHAPTER 3

SYSTEM DESIGN

3.1 Introduction

The Unified Modeling Language (UML) specified as modelling a widely valid lexicon. It is the major purpose of UML to provide a criterion for visualizing a program's architecture. It resembles earlier construction ideas quite a bit.

It is important to remember that language is a core theme, not a piece of software. Flow charts illustrate a kinematic chain. behavior and design. Language is a modelling, tools to promote the creation, innovation process of web applications.

3.2 System Architecture

Activity in network infrastructure aims to design a solution that are plausible and complete related as well as concepts that are both stable and reliable, concepts, and properties. Elements, traits, and attributes of the deployment meet the issue or opportunity stated in a number of hardware necessities to the maximum degree feasible in addition to being technologically realizable.

Vague, cognitive, holistic and fixated here on service's goals, configuration is esoteric. mission and cycle principles. Rising edifice too is addressed in this chapter. Discusses the system of architectural what you care about, concepts, attributes, and its traits and can When applied to a number of different technologies, it may help to create an unified basis, sequence, as well as criterion across several workshops or genera of logically related solutions.

3.2.1 Architecture Diagram

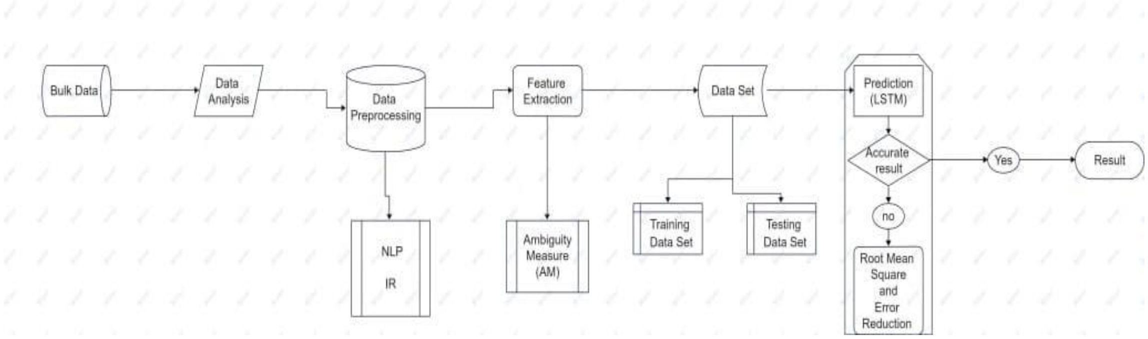


Fig 3.1 System Architecture

3.2.2 Description of Architecture Diagram

Figure 3.1 depicts

- Firstly, input data is given to the system collected from various sources in the form of csv file.
- Secondly, this data is processed to the next step that is Data Analysis which is delete the unwanted or incomplete relation.
- Data pre-processing is continuing step including NLP, IR.
- Feature Extraction includes Ambiguity Measure (AM).
- And then data set is divided into two sets: Training dataset, Testing dataset.
- Finally proceeding into predicting machine that is LSTM followed by result.

3.3 System Requirements

3.3.1 Introduction

Predictors of a venture include well criteria. Agreements are formed of both a client and a provider to work together toward a mutual goal. In addition to lowering fiscal threats plan punctuality, strong, explicit qualifications help. The Biz Dev Wealth of Data trite law as a useful description of its use.Steps like as encoding, commentary, articulation and assurance are all part of the process of developing mandates.

Business Drivers

As a starting point, the customer describes whether he or she is interested in developing the platform. Since the marketing experts, systems planners, and entrepreneurs will use this rationale to make decisions, it is essential. One of the most compelling reasons to explain the program's rationale is whether the client's employees may change throughout the beginning of the project.' In the event that the founding donor decides not to continue with the project, papers outlining the economic case for the program will help keep it alive. Problems (reasons why present systems/processes are insufficient) and opportunities are examples of potential drivers (new business models the system will make available). Motivation for a new system is usually provided by a mix of issues and possibilities.

Business Model

This section defines the system's basic business model for the client. The organizational context, current-state and future-state diagrams, business context, important business functions, and process flow diagrams are among the elements covered. Typically, this component is developed as a result of the functional analysis process.

Functional and System Requirements

This section normally contains a hierarchical arrangement of needs, with business/functional requirements at the top and specific system requirements as child items.

Business and System Use Cases

There may be a use instance chart depicting the elements and their multiple use applications (goals) that they'll have to accomplish in this section. For each utilize, there will be a description of the procedures that must be performed in existence for the company goals to be achieved, and also any pertinent before or following constraints. Use cases for the systems and the business are often found in different parts of an organization's functional needs.

Technical Requirements

All "semi" criteria that efficaciously contain the item's developments and its conceptual restrictions should be included in this section. In order to better understand how support higher demands are decomposed into the more exact feature set, these industry standards are essential.

System Qualities

The "efficiency" of a model is developed using a set of "semi" guidelines. The term "-ilities" refers to the fact that these kind of things end in "ility." An important consideration was whether or not the system could be reliably and easily maintained.

Constraints and Assumptions

Client-imposed limits will be addressed in this section, preventing the researchers from considering certain options in the system's design. Any presumptions by the standards technical department all across process of gathering and analyzing objectives will be included in this section. The network system testing must be s basically if any of the assertions are found to be erroneous, in order to guarantee that the marking are still valid.

Acceptance Criteria

Listed below are the standards that will be used to "log" on the completed system by the customer. Depending on the technique, this may occur at the end of the system of monitoring certification cycle either at the end of the process.

3.3.2 Hardware Requirements

- Processor: 1.4 Giga Hz 64 bit processor is required
- Disk Space: 100GB Free Space
- RAM: Minimum 8GB
- Graphics Device: A display with a resolution of 1024 × 768 or greater

3.3.3 Software Requirements

- Python
- Anaconda
- Jupyter Notebook
- TensorFlow Package
- Plotly Package
- Matplotlib Package

It's common for the criteria to include the need to complete all user acceptance tests and repair any defects/bugs that meet a pre-determined priority or severity level.

3.4 Software Description

3.4.1 Python Language

Python is a freeware piece of software. that is free to use. As a result, all you have to do is install Python once, and you're ready to go. Not to mention the fact that you can help the community by contributing your own code. Python is also a platform-independent language. So, what exactly does this imply? Python may be installed and executed on a variety of operating systems. Python will function on Win, apple, or Linus, to name a few examples of pc operating systems that support this.

Python is also a fantastic tool for visualizing data. It includes libraries for creating attractive visuals, such as Matplotlib, seaborn, and bokeh. Physical lines and logical lines or statements are both included in the Python coding language. Instead of the semicolon, the end of a physical line in a Python programme ends the line sequence. This is different from other programming languages, such as C and C++ that utilize semicolons to signify the end of statements. There are physical lines that compose a logical line, on the other hand. Python does not need the use of a semi-colon, although it is not disallowed. The NEWLINE token signifies the logical line's end.

Blank lines are logical lines that include just spaces, comments, or tabs and are disregarded by the interpreter.

Fig 3.2 Python Language

A new line merely signals that a new sentence has begun, as we saw in Python. Fig 3.2 Python does, however, allow you to break a statement over many lines or unite multiple statements into a single logical line. This might assist to make the sentence easier to understand. The two methods for splitting a line into two or more lines are as follows:

- Explicit Line Joining

  A reverse slash is used to break a sentence into several lines in explicit line joining.

- Implicit Line Joining

  Without employing a back slash, statements contained inside [],, or () parentheses may be split down into two or more physical lines.

- Multiple Statements on a Single Line

  In Python, a semi-colon may be used to group numerous statements on the same line; However, most programmers don't think this is a good practice since it makes the code more difficult to understand.

- Whitespaces and Indentation

  Python, unlike most of the other programming languages, uses indentation to denote a code block. The Python coding style guideline, or PEP8, recommends a four-indentation size.

Indentation is provided by most programming languages for better code layout, although it is not required. However, it is required in Python. This is why, in Python, indentation is so important.

- Comments

  Comments are used to improve the readability of code in any programming language. Similarly, as a programme becomes more involved in Python, using Python comments is one of the greatest methods to keep the code readable. If you're working on a project with a large number of other programmers, it's important that everyone can understand the code, therefore include documentation and comments in the python syntax is considered best practice.

The following are examples of several types of comments that may be used in our Python programme:

- Single Line Comments

  The # character is used to indicate single-line Python comments. All characters after the # character are considered part of the remark, which ends at the end of the physical line.

- Docstring Comments

  Documentation strings (or docstrings) are a feature of Python that is generally the initial statement in methods and modules.

  Docstrings may be accessible at run time via the dot operator, rather than being disregarded by the Python Interpreter like standard comments.

  It allows programmers to quickly add comments to each Python module, function, class, and method. This capability is made possible by the use of triple quotes at the beginning and end of the documentation string or comment. Docstrings may be either one-liners or multi-liners.

- Multiline Comments

  Unlike certain programming languages that enable multiline comments, such as C, Java, and others, Python does not have a dedicated functionality for them. However, this does not rule out the possibility of creating multiline comments in Python. In our Python code, there are two methods to incorporate comments that span many lines.

  Python Block Comments: For an entire block, we may utilise numerous single line comments. This remark is frequently used to describe the code block that comes after the Block comment. The only method to write a true remark that spans many lines in Python is to use a block comment. Because Block comments are disregarded by the Python interpreter or parser, it is accepted and favored by Python's PEP8 style standard.

Understanding how data is stored and managed in a programming language is one of the most important aspects of learning any programming language. Python is popular among programmers because of its simplicity of use and wide range of capabilities. Dynamic typing is one of these characteristics.

Python doesn't really need the record of a field to be made clear, unlike strongly typed programs like C or Java. In reactive programming languages such as Python, the interpreter. anticipates based on the figure sent into the Python Parameter, determines the data data types.

3.4.2 Anaconda Software

Anaconda is the data science platform of the future, designed for data scientists, IT professionals, and business executives. It's a Python, R, and other programming language distribution. It becomes one of the greatest platforms for any project with over 300 data science tools.

The use of Anaconda simplifies the process of managing and distributing packages. Machine learning and artificial intelligence algorithms may be applied to a wide variety of data sources using Anaconda's built-in toolsets. It helps to create an environment that can be simply controlled and deploy any project with a single click.

Anaconda makes package distribution and administration much easier. Furthermore, it includes a number of technologies that use artificial intelligence and machine learning techniques to assist in data collection.

Using Fig 3.3 Anaconda Navigator instead of the command line, Conda packages, environments, and channel may be run and managed. The desktop interface for Anaconda is called Anaconda Navigator. It can look for packages in an Anaconda repository on your computer or on Anaconda Cloud. You no longer need to input commands in a terminal since Navigator allows you to deal with packages and environments with a single click.

Fig 3.3 Anaconda Software

3.4.3 Jupyter Notebook

There are a slew of new online tools for working with notebooks, code, and data, the latest of which is called Jupyter Lab. Modularity encourages the development of new features that expand and increase functionality. When it comes to creating and distributing computational documents online, Jupyter Notebook is the gold standard. I like the fact that it's easy to use and focuses on documents.

A notebook is a document that includes a variety of different types of media, such as pictures, narrative text, mathematical computations, and more, all in one place. This is an easy-to-use tool that allows you to run code on a single page while simultaneously displaying the results and adding details such as formulas and charts to make your work easier to understand and share.

At firms all across the world, using Notebooks is now an important element of the data science process. If you want to work with data, a Notebook can streamline your process and make it simpler to discuss and share your findings.

Fig 3.4 The Jupyter Notebook App is a server-client application that lets you modify and execute your notebooks from a web browser. For those who don't have an Internet connection available, the software may be run on a local computer and accessible via the Internet.

Kernels and a dashboard are the two primary components.

A kernel is a software that executes and examines the code of the user. There is a kernel for Python code in the Jupyter Notebook App, but there are other kernels for other programming languages.

The application's dashboard not only displays the notebook pages you've created and can reopen, but it can also be used to control kernels, allowing you to see which ones are operating and shut them down if required.

.

Fig 3.4 Jupyter Notebook

The Jupyter Notebook features include

- Pluggable authentication

  PAM, OAuth, or integration with your own directory service system may be used to manage users and authentication.

- Centralized deployment

  Thousands of users in your business can utilise the Jupyter Notebook thanks to centralized infrastructure on- or off-site.

- Container friendly

  To scale your deployment, segregate user processes, and simplify software installation, utilise Docker and Kubernetes.

- Live coding environments

  Changes to the code may be made in real time, and feedback is supplied straight in the browser.

- Code meets data

  To enable unified software administration and data access inside your business, deploy the Notebook alongside your data.

3.4.4. TensorFlow

Deep learning is a machine learning discipline that consists of a collection of algorithms inspired by the structure and function of the brain. Machine learning includes deep learning as a subset. Machine learning is divided into specialisations, which is why it is referred to as deep learning. Deep learning, for example, makes use of neural networks, which are similar to human brain simulations. Unlike classical machine learning, which normally employs structured data, deep learning also entails evaluating enormous volumes of unstructured data.

Images, video, audio, text, and other forms of unstructured data might be supplied into the system.

A second machine learning model created by Google is TensorFlow, which is used to design, create, and train deep learning models. Even while numerical calculation using the TensorFlow framework may not appear notable in and of itself, data flow graphs are employed to carry out these computations. Mathematical operations are represented by the vertices in these graphs, while the data that is passed between them is often represented by the edges.

TensorFlow is named after the processes that neural networks perform on multiple data arrays, or tensors! It's a tensor flow in its purest form. Using tf.keras, you can build, fit, evaluate, and apply deep neural networks to generate predictions in only a few short lines of code. Deep learning activities like regression and classification predictive modelling may be performed using this software.

Another essential feature is TensorFlow's scalability. For training purposes, you may write your code and have it run on a CPU, GPU, or a cluster of these devices. In general, a considerable portion of the computation is spent on training the model. In addition, the training procedure is repeated many times to address any difficulties that may develop. As a result of the increased power usage, you'll need to use distributed computing. TensorFlow makes it simple to handle massive quantities of data by executing the code in a distributed way.

Graphic processing units, or GPUs, have grown quite common. Nvidia is a market leader in this area. It excels at mathematical operations like matrix multiplication and plays an important part in deep learning. TensorFlow also features C++ and Python API integration, making development considerably quicker.

A tensor is a mathematical entity that is represented by higher-dimensional arrays. The neural network is supplied with these arrays of data of various sizes and rankings. The tensors are what you're looking for.

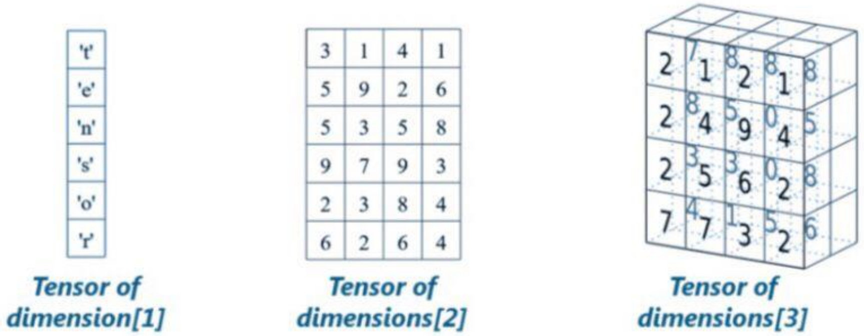


Fig 3.5 Tensor Flow

Fig 3.5 One-dimensional arrays or vectors, as well as two-dimensional matrices, are available. Tensors, on the other hand, may have more than three, four, or five dimensions. As a result, it aids in maintaining all of the data in one place and then completing all of the analysis around it.

CHAPTER 4

MODULE DESCRIPTION

4.1 Module-1 Data Analysis

To uncover the underlying law in exploratory analysis, the distribution of data is employed. EDA is a sort of data analysis that makes use of visual tools to discover the structure of the data. This is owing to the fact that human eyes and brains have a high structural capacity to detect, which is why visual data analysis techniques have been employed for many centuries in a broad variety of applications. Visual analysis, on the other hand, is the application of a wide range of human models to the processing capability of a particular presentation style for facts. Deciding on the structure or stochastic quantity of the data, analysts always do exploratory data analysis to reveal unforeseen deviations that traditional theories cannot. Exploratory Data Analysis' main feature is the flexibility with which it applies to the data structure as well as the flexibility with which it reacts to the disclosed mode of the subsequent analysis step. During exploratory data analysis, the observed mode or effect-evoke is evaluated. The verification stage includes gathering and analysing additional data to confirm the results, as well as combining other closed related information.

4.2 Module-2 Pre-Processing

Information retrieval, natural language processing (NLP), and text mining have such a deal do to with, pre-processing is an important work and crucial stage (IR). Data pre-processing is a technique for extracting valuable and non-trivial information from unstructured text data in the field of Text Mining. Information retrieval (IR) is the process of determining which documents in a collection should be retrieved in order to meet a user's information needs. A query or profile represents the user's demand for information, and it contains one or more search phrases, as well as some other information like word weight. As a result, the retrieval choice is determined by comparing the query's terms to the document's index terms (significant words or phrases). The decision could be binary (retrieve/reject) or entail judging the document's degree of significance. Unfortunately, many structural variants exist among the terms that appear in documents and queries. So, before retrieving information from documents, data preparation techniques are used to lower the size of the target data collection, increasing the effectiveness of the IR System. Because multiple inflected forms of the same lemma may occur seldom (or not at all) during training, the goal of lemmatization is to diminish sparsity. This, however, may come at the expense of overlooking key syntactic details. These multiword phrases' meanings are frequently difficult to deduce from their constituent tokens. As a result, treating multiwords as individual units may result in improved model training. As a result, word embedding toolkits like Word2vec present statistical ways for extracting multiwords or include multiwords alongside single words in their pretrained embedding spaces.

4.3 Module-3 Feature Extraction

Picking just those features that are most essential to the project's scope is known as feature selection. Wrappers, filters, and embedding techniques are the three types of feature selection methods. When a set of features is collected via the feature selection process, a new collection of features is obtained through the feature extraction process instead. The features that occur consistently in only one category will be given a higher score by the Ambiguity Measure (AM) feature selection approach. For each feature, an AM score is calculated. A score near to 1 is assigned if the feature is unambiguous; otherwise, a score close to 0 is assigned. A single threshold is set, and features with an AM score below that threshold are filtered, while features with an AM score above that level are used in the learning phase. The difference in entropy between whether or not a feature exists in the text is used to calculate the information gain of a feature. The greater the information gain, the more the features contribute to the text. As a feature, characteristics with a significant information gain will be chosen.

4.4 Module-4 Prediction

The LSTM network is a recurrent neural network that uses Backpropagation Through Time training to overcome the vanishing gradient issue. Consequently, enormous recurrent networks may be constructed using it, and these networks can be utilised to address complex machine learning sequence problems and achieve cutting-edge results. Instead of neurons, LSTM networks employ memory blocks that are linked by layers. With the ability to remember recent sequences, a block is more intelligent than a typical neuron. As the state and output of a block change, so do the gates that control them. When activated, each gate in a block, controls whether or not the information going through the block changes state or adds new information. The Recurrent Neural Network is a kind of neural network with a large short-term memory (RNN). RNNs use information from prior time events to help them predict future events. For example, in order to categorise what type of event is occurring in a film, the model must consider past events. RNNs operate well when the problem just requires recent information to complete the task at hand. RNN might struggle to model a situation that demands long-term dependencies. The LSTM was created with the goal of learning long-term dependencies. It retains the information for a long time.

4.5 Waterfall Model

It is referred to as a conventional method. Waterfall is a linear (sequence) strategy for developing any software programme. The development of an application is divided into a series of pre-defined stages. This is easily depicted in the Fig. 4.1. It includes

1. **Requirement gathering and documentation**

   It's correct to conclude that you have a firm grasp of the scale of the issue at hand. For this information, you may use several approaches, such as interviews, surveys, and brainstorming. It is imperative that the task requirements and a requirements file be allocated to your team before the finish of this section.

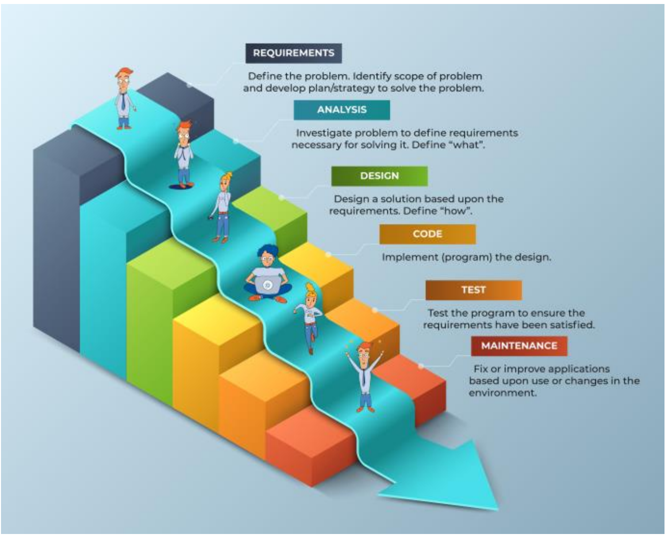

REQUIREMENTS
Define the problem. Identify scope of problem and develop plan/strategy to solve the problem.
ANALYSIS
Investigate problem to define requirements necessary for solving it. Define "what".
DESIGN
Design a solution based upon the requirements. Define "how".
CODE
Implement (program) the design.
TEST
Test the program to ensure the requirements have been satisfied.
MAINTENANCE
Fix or improve applications based upon use or changes in the environment.

Fig 4.1 Water fall

2. **Requirement gathering and documentation**

It's correct to conclude that you have a firm grasp of the scale of the issue at hand. For this information, you may use several approaches, such as interviews, surveys, and brainstorming. It is imperative that the task requirements and a requirements file be allocated to your team before the finish of this section.

3. **System Design**

Your team creates solutions based on well-known criteria. There will be no development during this time. However, the project team begins by defining needs such as programming language or hardware.

4. **Implementation**

Coding will take place at this phase of programme development. After gathering the necessary data, web application programmers use it to create a functional product. Code for web applications is written in smaller bits and assembled at various stages of development.

5. **Testing**

It is possible to begin testing the product after all the code has been written. In order to find and identify any problems, testers do extensive testing. Phase one reevaluation may be necessary if severe issues arise.

6. **Delivery/Deployment**

   As a result, your project team is ready to go on with deploying or releasing the final deliverables.

7. **Maintenance**

   The final solution has been built and is in use by the customer. As issues develop, Patches and updates to resolve these issues may be required by the project team. Major difficulties may need a trip back to the beginning.

4.6 RAD model

Rapid application development (RAD) as in Fig. 4.2 is an agile software development approach that prioritises current software projects and user feedback above following a rigid plan. Fast prototyping takes precedence over meticulous preparation, as a consequence.

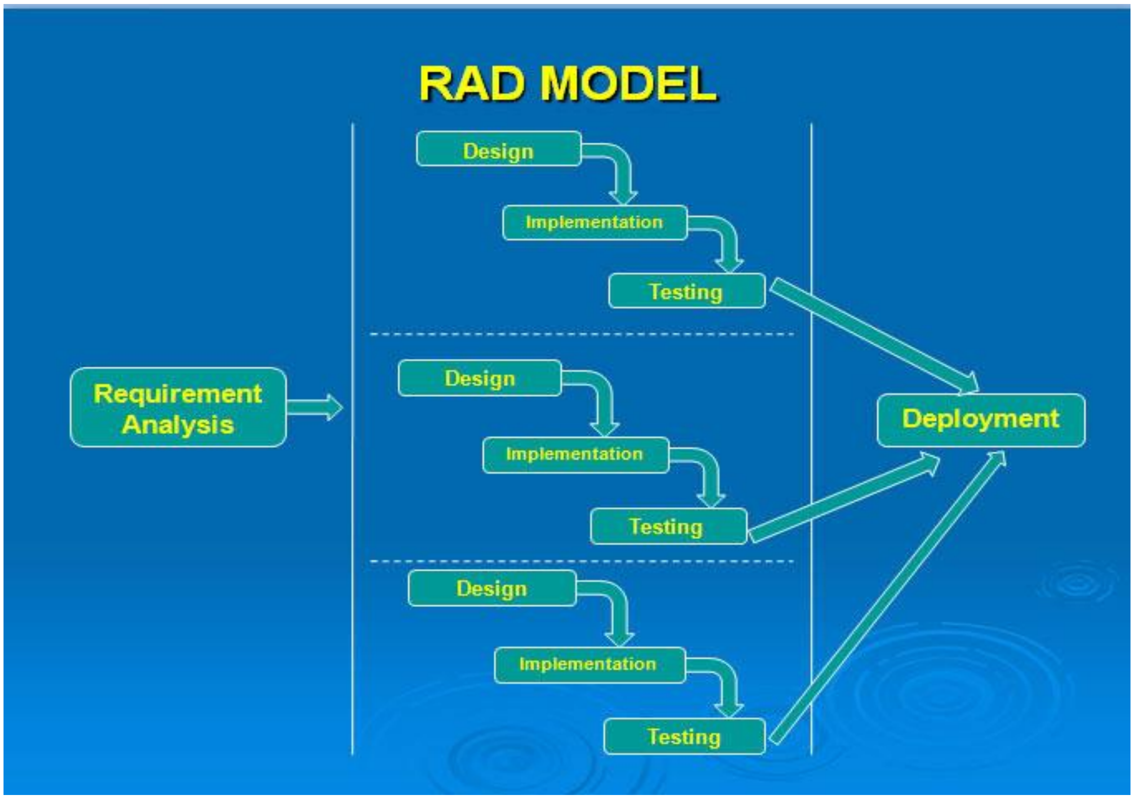


Fig 4.2 RAD Modal

1. **Define Requirements**

Rather of requiring you to spend months defining specifications with users, RAD begins by creating a loose set of criteria. It refers to the ability to modify requirements at any point throughout the development cycle, which is one of the basic concepts of fast application development. Basically, coders gather the essentials. As a team, the customer and developers work together to create a product that fulfils the client's needs. It is necessary to have a meeting to determine the scope of the project. Although the planning phase is short compared to other project management systems, it is critical to the project's success.

Programmers, customers (software users), and other members of the team work together at this stage to establish the project's goals and expectations, while also addressing any current or potential challenges that may arise during construction.

A basic breakdown of this stage involves: It's vital that everyone involved has an opportunity to weigh in on the project's goals and expectations and provide input. Consent from all important stakeholders, including web application coders, may help teams avoid misunderstandings and subsequent costly change orders.

2. **Prototype**

During this rapid application development phase, the developer's goal is to produce something that the consumer can see. All or some of the requirements may be met by this prototype.

To get to a workable condition, this prototype may have to cut corners, which is OK. In most rapid application development (RAD) methods, developers pay off the technical debt accrued by early prototypes during finalization. Every hiccup and glitch is smoothed out in an ongoing process. Both sides meet to discuss what worked and what didn't after the web application developer produces a prototype for testing.

Using this strategy, web application programmers are able to experiment with the model until they discover a design they like. The expertise gained by both the software web application programmers and the consumers ensures that nothing gets overlooked.

3. **Absorb Feedback**

   Customers and end-users may see RAD developers' efforts in action by examining a current prototype. Product specifications may be scrutinized here, as well as feedback on anything from the user interface to functioning.

   Clients' views may shift or they may come to realise that what seemed to be accurate on paper is incorrect in reality. After all, clients are just people. After collecting input, developers go back to the drawing board and continue prototyping. Developers may proceed to step 4 if all feedback is good and the customer is pleased with the prototype.

   Since most of the concerns and modifications were dealt with during the rigorous iterative design process, web application programmers may be able to develop the final operating model quicker than they would using a traditional project management technique.

   Software developers, coders, testers, and web application programmers work together to guarantee that the final product fulfils the client's expectations and objectives at this stage.. Customer input might come at any stage in the process, thus this third phase is critical. In addition, they may provide recommendations for new ways to deal with problems as they arise.

4. **Finalize Product**

   Stability and maintainability may be improved by re-engineering or optimising the system now. If they need to integrate the backend with production data, offer thorough documentation and do any other maintenance tasks, they may do so within this period.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Implementation Details

5.1.1 Sample coding

```
import os
import random
import numpy as np
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.tsa.seasonal import seasonal_decompose
energy_df = pd.read_csv("Dataset/Energy.csv", parse_dates=['time'])
weather_df = pd.read_csv("Dataset/Weather.csv", parse_dates=['dt_iso'])
energy_df.head()
energy_df.info()
weather_df.head()
weather_df["city_name"].unique()
weather_df.query("city_name == 'Madrid' ").head()
weather_df.info()
city_counts = weather_df[“city_name"].value_counts()
def display_bar_chart(data, xvalues, yvalues, graph_title, xtitle, ytitle):
fig = px.bar(data, x=xvalues, y=yvalues, color=xvalues)
fig.update_layout(title=graph_title, xaxis_title=xtitle, yaxis_title=ytitle)
fig.show()
display_bar_chart(city_counts, city_counts.index, city_counts.values, "City wise count", "City
Name", "Count")
energy_df["year"]=energy_df["time"].apply(lambda x: x.year)
energy_df['time']=pd.to_datetime(energy_df['time'], infer_datetime_format=True, utc=True)
energy_df = energy_df.set_index(‘time')
energy_year_counts = energy_df[“year”].value_counts()
display_bar_chart(energy_year_counts, energy_year_counts.index,
energy_year_counts.values, "Energy Year wise count", "Year", “Count")
weather_df["year"]=weather_df.dt_iso.apply(lambda x: x.year)
weather_year_counts = weather_df[“year"].value_counts()
display_bar_chart(weather_year_counts, weather_year_counts.index,
weather_year_counts.values, "Weather Year wise count", "Year", "Count")
energy_df.index.min(),energy_df.index.max()
energy_df1 = energy_df.copy()
len(energy_df)
energy_df.isnull().sum()
energy_df1 = energy_df.copy()
energy_df1.index = pd.to_datetime(energy_df1.index, utc=True)
load_series = energy_df1["total load actual"]
load_series.sample(15)
forecast_series = energy_df1["total load forecast"]
forecast_series.sample(15)
energy_demand_df = pd.concat([load_series.dropna(), forecast_series.dropna()], axis=1)
energy_demand_df.columns = ['Energy_Load', 'Energy_Forecast']
energy_demand_df
energy_demand_df.describe()
fig = go.Figure()
fig.add_trace(go.Histogram(x=energy_demand_df["Energy_Load"],name='Energy
Load',cumulative_enabled=False))
fig.add_trace(go.Histogram(x=energy_demand_df["Energy_Forecast"],name='Energy
Forecast',cumulative_enabled=False))
fig.update_traces(opacity=1)
fig.update_layout(barmode='stack',
    title_text='Distribution of energy demanded and TSO 1 day forecasts',
xaxis_title_text='MW Power',
    yaxis_title_text='Energy', bargap=0.1,  bargroupgap=0 )

energy_groups = energy_demand_df['Energy_Load'].groupby(pd.Grouper(freq='A'))
fig = go.Figure()
fig = make_subplots(rows=len(energy_groups), cols=1)
row = 1
for name, group in energy_groups:
pd.Series(group.values)
fig.add_trace(go.Scatter(y=group.values, name=name.year), row=row, col=1)
row = row + 1
fig.update_layout(height=1200, width=800, title_text="Yearly Consumption")
fig.show()
energy_demand_df
energy_df.columns=[x.replace('/','_') for x in energy_df.columns]
energy_df.columns=[x.replace('-','_') for x in energy_df.columns]
energy_df.columns=[x.replace(' ','_') for x in energy_df.columns]
energy_df.apply(lambda x: len(x.unique()))
energy_df = energy_df.drop(['generation_fossil_coal_derived_gas','generation_fossil_oil_sha
'generation_fossil_peat', 'generation_geothermal',
'generation_hydro_pumped_storage_aggregated',
'generation_marine', 'generation_wind_offshore', ‘
forecast_wind_offshore_eday_ahead', 'forecast_solar_day_ahead',
'forecast_wind_onshore_day_ahead'], axis=1)
def show_raw_visualization(data, nrows, width, height):
    #time_data1 = data.index
    fig, axes = plt.subplots(
        nrows=nrows, ncols=2, figsize=(width, height), dpi=80, facecolor="w", edgecolor="k"
    )
    for i in range(len(feature_keys)):
        key = feature_keys[i]
        c = colors[i % (len(colors))]
        t_data = data[key]
        t_data.index = data.index
        t_data.head()
        ax = t_data.plot(
            ax=axes[i // 2, i % 2],
            color=c,
            title="{} - {}".format(key, titles[i]),
            rot=25,
        )
        ax.legend([titles[i]])
    plt.tight_layout()
 colors = ["blue", "orange", "green", "red", "purple", "brown", "pink", "gray", "olive", "cyan"]
titles = [ 'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'MWh',
          'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'MWh', 'Euro', 'Euro']
feature_keys = ['generation_biomass', 'generation_fossil_brown_coal_lignite',
'generation_fossil_gas', 'generation_fossil_hard_coal']
show_raw_visualization(energy_df[:24*7*4],nrows=2,width=12,height=4)
energy_df.describe().transpose()
def show_hist_boxplot(cols):
    cols = cols
    for col in cols:
        plt.figure(figsize=(12,2))
        plt.subplot(1,2,1)
        sns.histplot(data=energy_df, x=col, bins=30, kde=True)
        plt.subplot(1,2,2)
        sns.boxplot(data=energy_df, x=col)
        plt.show()
energy_df.loc[energy_df.generation_fossil_brown_coal_lignite ==0,
'generation_fossil_brown_coal_lignite'] = np.nan
energy_df.loc[energy_df.generation_biomass <110, 'generation_biomass'] = np.nan
energy_df.loc[energy_df.generation_fossil_gas ==0, 'generation_fossil_gas'] = np.nan
energy_df.loc[energy_df.generation_fossil_oil <50, 'generation_fossil_oil'] = np.nan
energy_df.loc[energy_df.generation_nuclear <2000, 'generation_nuclear'] = np.nan
energy_df.loc[energy_df.generation_other_renewable <20, 'generation_other_renewable'] =
np.nan
energy_df.loc[energy_df.generation_waste <50, 'generation_waste'] = np.nan
def show_heat_map(df,cols, width=5, height=5):
    width=width
    height=height
    df=df
    cols=cols
    plt.figure(figsize=(width,height))
    sns.heatmap(df[cols].corr(), annot=True)
    plt.show()
cols =['generation_hydro_pumped_storage_consumption', 'generation_fossil_gas',
'generation_fossil_oil',
  'generation_hydro_water_reservoir', 'price_actual', 'generation_fossil_hard_coal',
'generation_fossil_brown_coal_lignite', 'total_load_actual']
show_heat_map(energy_df,cols,10,2)
cols = ['pressure', 'humidity', 'wind_speed', 'wind_deg', 'clouds_all', 'weather_id']
for col in cols:
    weather_df[col]=weather_df[col].values.astype(np.float64)
weather_df['time']=pd.to_datetime(weather_df['dt_iso'], infer_datetime_format=True,
utc=True)
weather_df = weather_df.set_index('time')
weather_df = weather_df.drop(['dt_iso'], axis=1)
weather_df.apply(lambda x: len(x.unique()))
```

```
weather_df = weather_df.drop(['rain_1h','weather_main', 'weather_description','weather_icon', 'weather_id'], axis=1)
feature_keys = ['temp', 'pressure', 'humidity','wind_speed']
show_raw_visualization(weather_df[:24*7*4],nrows=2,width=12,height=4)
weather_df.describe().transpose()
print(weather_df.duplicated().sum())
features = ['pressure','wind_speed', 'wind_deg', 'rain_3h']
def show_hist_boxplot2(features):
    features = features
    for feature in features:
        plt.figure(figsize=(12,3))
        plt.subplot(1,2,1)
        sns.histplot(data=weather_df, x=feature, bins=30, kde=True, hue='city_name')
        plt.subplot(1,2,2)
        sns.boxplot(data=weather_df, x=feature, y='city_name')
weather_df.loc[weather_df.pressure >1030, 'pressure'] = np.nan
weather_df.loc[weather_df.pressure <800, 'pressure'] = np.nan
weather_df.loc[weather_df.wind_speed >100, 'wind_speed'] = np.nan
weather_df.loc[weather_df.rain_3h >2, 'rain_3h'] = np.nan
weather_df.interpolate(method='linear', limit_direction='forward', inplace=True, axis=0)
wv = weather_df['wind_speed']
wd_rad = weather_df['wind_deg']*np.pi / 180
weather_df['Wx'] = wv*np.cos(wd_rad)
weather_df['Wy'] = wv*np.sin(wd_rad)
df_bcn, df_bil, df_mdr, df_sev, df_val = [x for _, x in weather_df.groupby('city_name')]
dfs = [df_bcn, df_bil, df_mdr, df_sev, df_val]
data = energy_df
for df in dfs:
    city = df['city_name'].unique()
    city_str = str(city).replace("'", "").replace('[', '').replace(']', '').replace(' ', '')
    df = df.add_suffix('_{}'.format(city_str))
    data = data.merge(df, on=['time'], how='outer')
    data = data.drop('city_name_{}'.format(city_str), axis=1)
cols =['humidity_Valencia','humidity_Barcelona', 'humidity_Seville','humidity_Bilbao',
       'temp_min_Valencia','temp_Valencia','temp_max_Valencia', 'temp_Seville',
       'humidity_Madrid','temp_Madrid','wind_speed_Valencia', 'total_load_actual']
show_heat_map(data, cols, 10, 2)
import missingno as msno
def process_missing_data(data):
    missing_df=pd.DataFrame()
    missing_df["missing_row_count"]=data.isnull().sum()
    missing_df["missing_row_data"]=data.isnull().sum()/len(data)

missing_df=missing_df.loc[missing_df["missing_row_data"]>0].sort_values(by="missing_row_data", ascending=False)
    missing_df["top"]=missing_df["missing_row_data"].map(lambda x:1 if x>=0.05 else 0)
    return missing_df
process_missing_data(energy_df)
energy_df.eq(0)
energy_df.eq(0).sum().to_frame(name="Missing Count")
msno.bar(energy_df,figsize=(12,5),
         sort="descending",
        fontsize=12);
msno.dendrogram(df=energy_df,figsize=(12,5),fontsize=12)
msno.heatmap(df=energy_df)
energy_df.isnull().sum()
energy_df.dropna(inplace=True)
energy_df.isnull().sum()
def show_decompose(result):
    result=result
    s1 = result.observed
    s2 = result.seasonal
```

```
s3 = result.trend
s4 = result.resid
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(16, 8), sharex=True)
fig.suptitle('Descomposition of multiplicative time series\n jan 2015')
ax1.plot(t, s2, 'k')
ax1.set_ylabel('Seasonal')
ax2.plot(t, s3, 'k')
ax2.set_ylabel('Trend')
ax3.plot(t, s4, 'k')
ax3.set_xlabel('time (H)')
ax3.set_ylabel('Resid')
plt.show()
```

5.2 Implementation Images

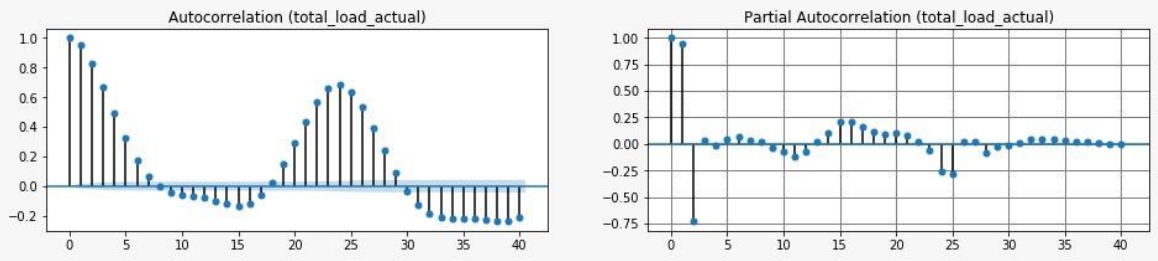


Fig 5.1 Auto Correlation

Fig.5.1 Randomness in the time series may be detected using an autocorrelation test. Assuming that the data produced is random is a common assumption in statistical operations. We need to look at the autocorrelation of lag 1 to ensure randomization. There are several ways to examine the relationship between past and future time series data.

Autocorrelation functions (PACFs) are used in data sets to calculate a stagnant time series's partial correlations with its own lagged values, regressed at all reduced lags. The autocorrelation function, on the other hand, does not regulate other delays.

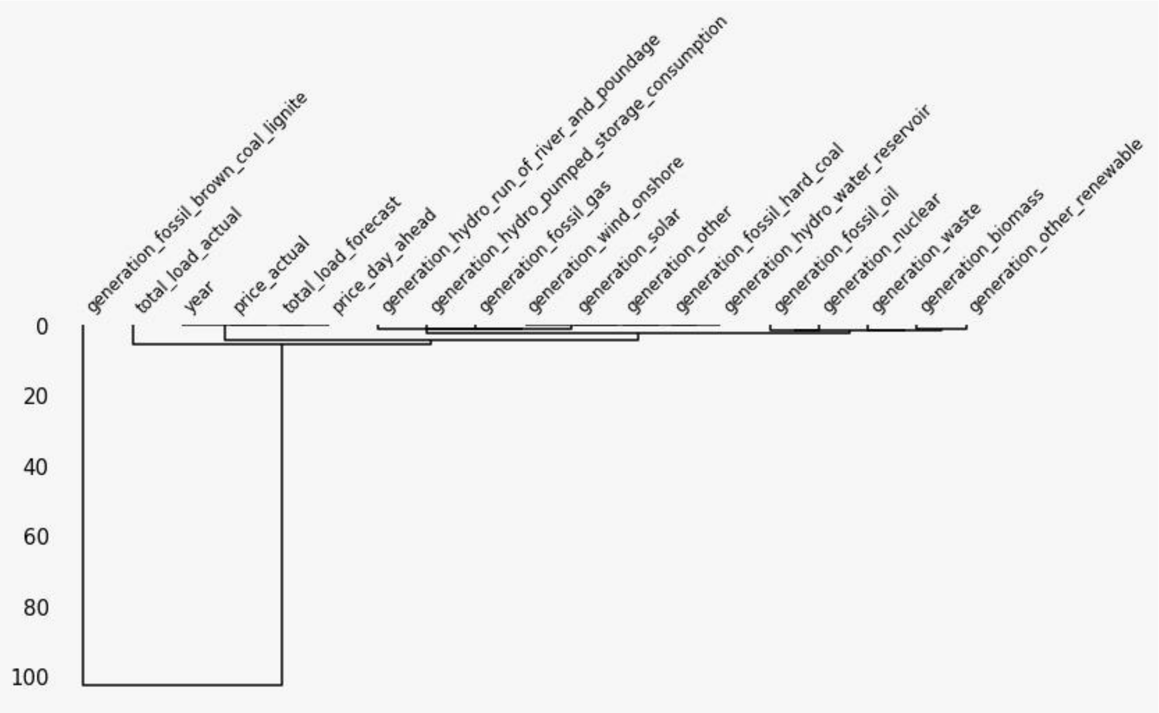


Fig 5.2 Plot line diagram of contents

Fig. 5.2 A dendrogram is a diagram that depicts a tree's structure. When data is clustered hierarchically, the figure factory creates a dendrogram, which generates a tree representation. Distances between clusters are represented by values on the tree depth axis. Computational biology often uses dendrogram plots in order to demonstrate the grouping of genes or data, like the distance between fossil_brown_coal_lignite and total_load_forecast is 100 units as follows.

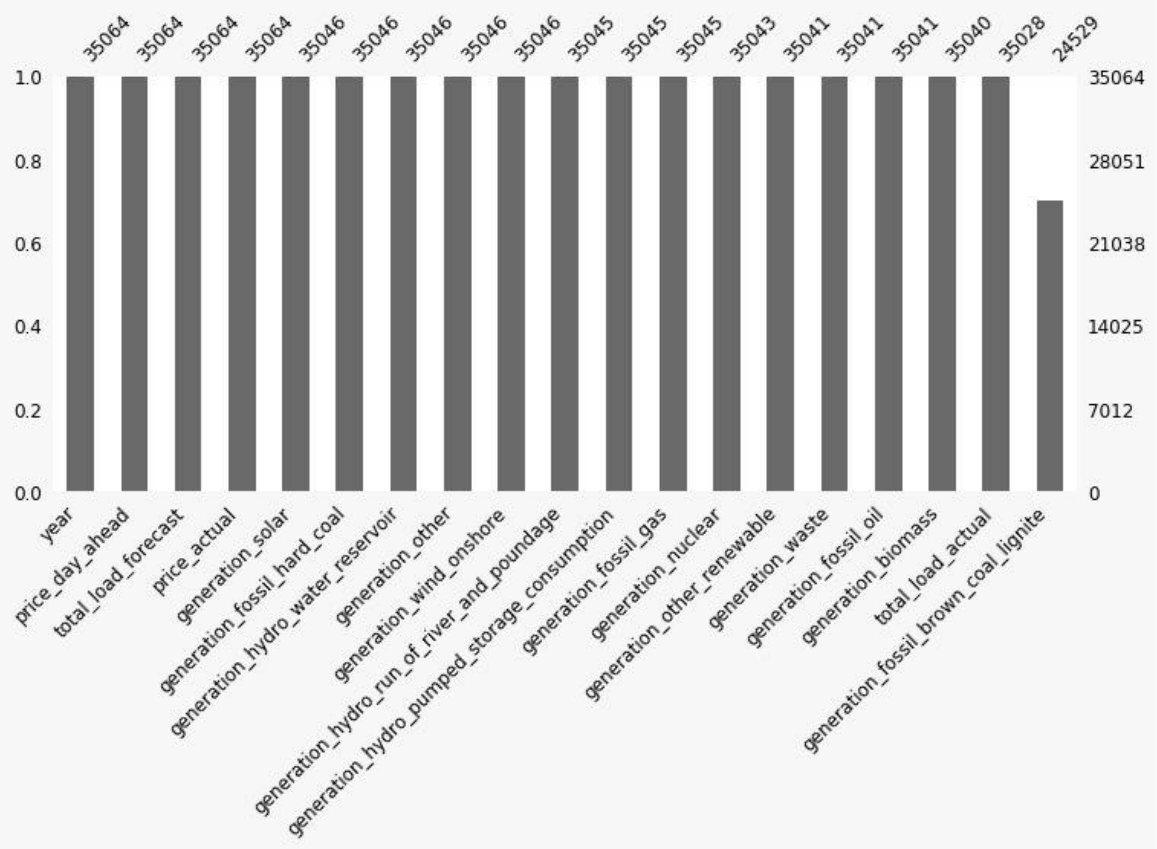


Fig 5.3 Contents vs Mean Graph

Figure 5.3 A horizontal bar graph with four values two x-axes with the names of the contents and the number of values, two y-axes with scaled values in the 0.2 unit range, and a right y-axis with a 7012 range The bar is plotted according to these four values and the axis.

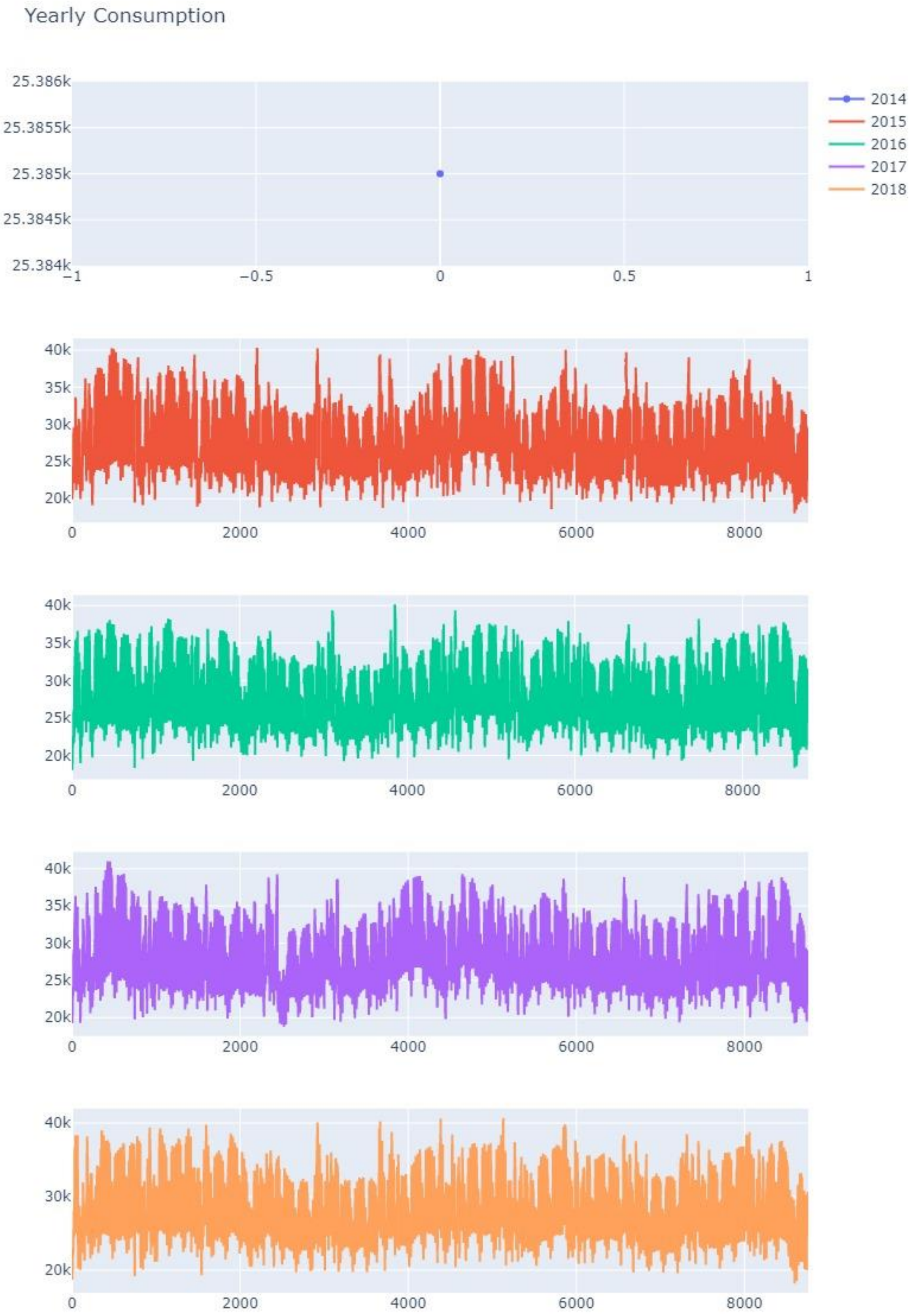


Fig 5.4 Matplot graph of year wise changes

Fig: 5.4 Four graphs are plotted with year wise from 2014 to 2018.In 2014 the highest recorded is 25.385k and in 2015 represented as red color with max height of 40k and range of 8000.Where as in 2016 in green color with max height of 41k and range of 8000.And as follows

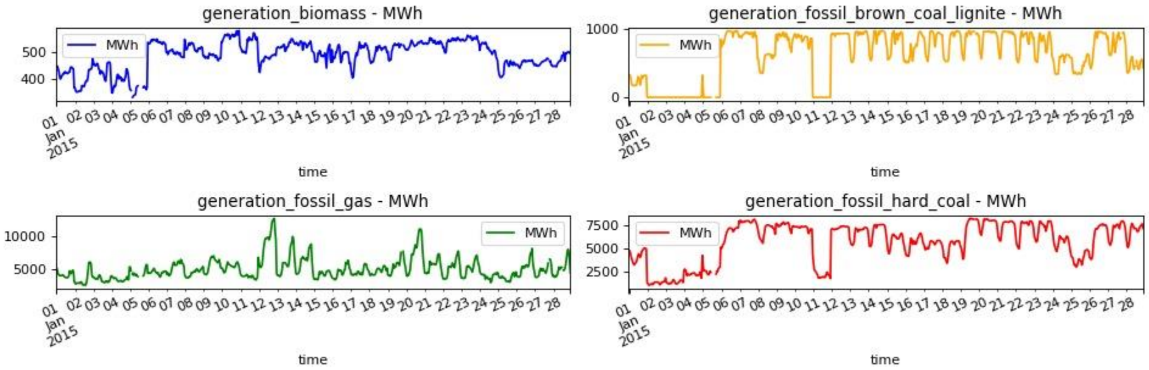


Fig 5.5 Sources of Energy

Fig 5.5 Each energy source is presented by a graph with a MWh range of 28 and height and peaked at 1000,7500,10000 in time. There are different variations for each and every source.

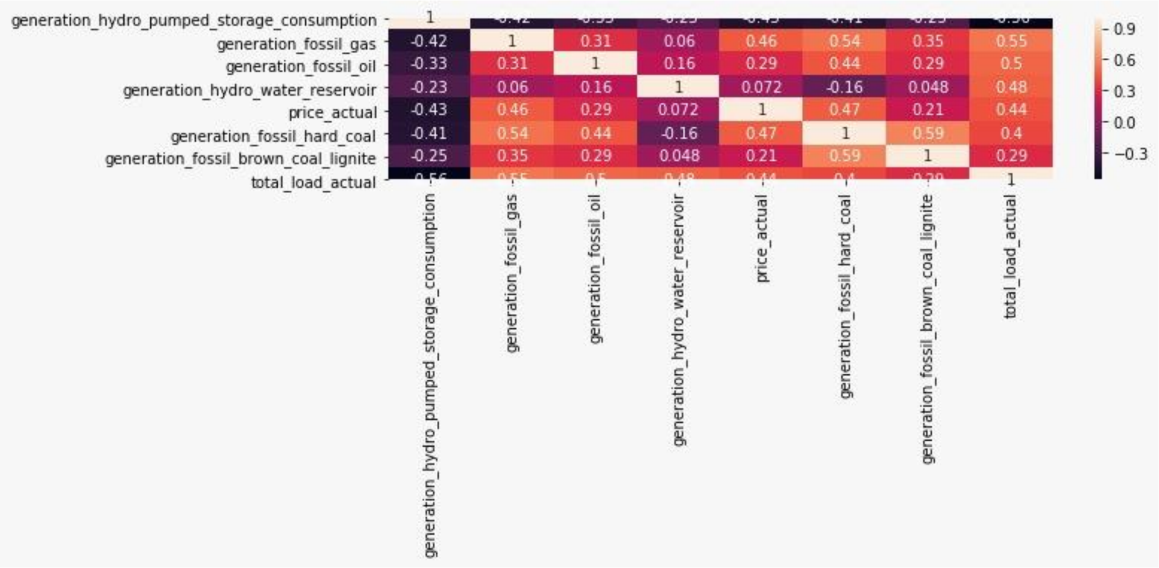


Fig 5.6 Heat map of Sources

Fig 5.6 Data analysis is done using a heat map. As we can see, the diagonal path is the same throughout the image. So, we can finalise that the contents are valid and good to go for further processing.

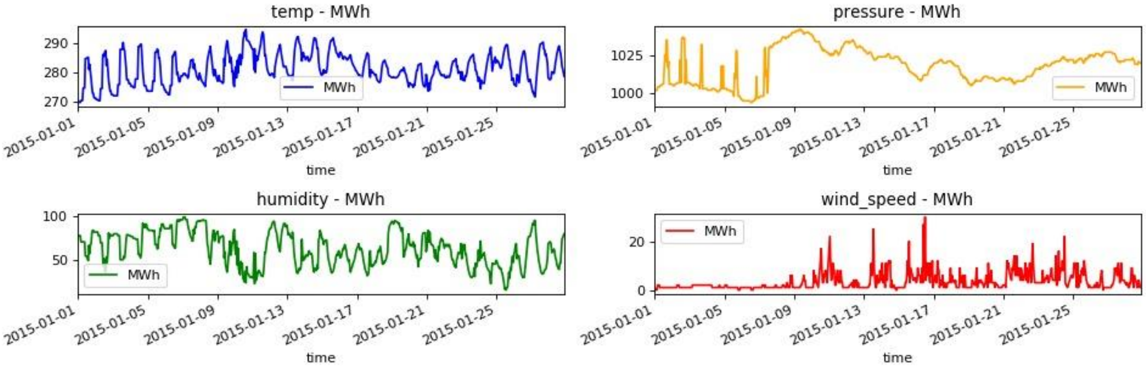


Fig 5.7 Year wise changes in sources

Fig 5.7 Data visualisation aids in the comprehension, observation, and analysis of data by providing a well-organized graphical depiction of it. Python will be used to create data visualisations in this session. Furthermore, if the data is in its raw form, it may be more difficult to look for specific trends or patterns. This is where data visualisation comes in.

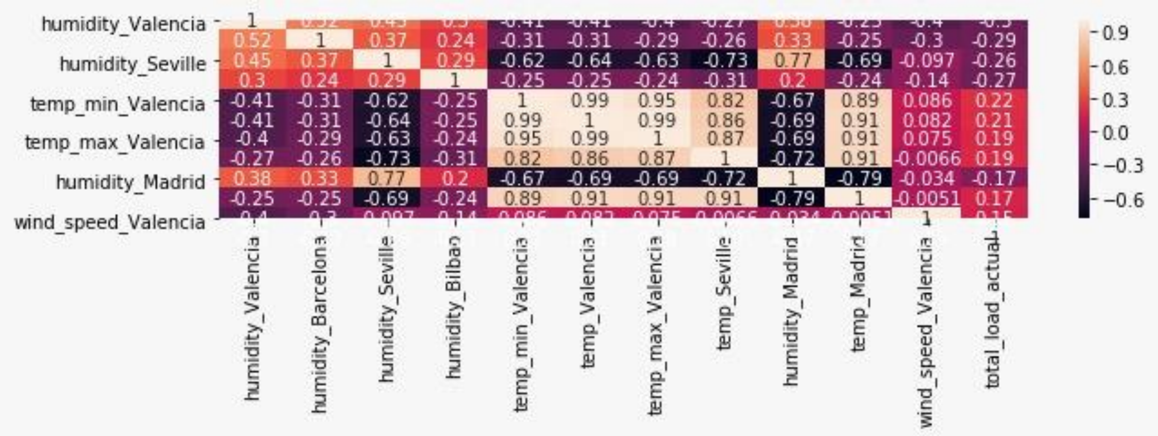


Fig 5.8 Heat map of features

Fig 5.8 The term "heatmap" refers to a visual depiction of a matrix's value using colour. In this case, brighter reddish hues are used to symbolise more common values or greater activity levels, whereas darker colours are used to represent less frequent or activity values. The name of the shading matrix is also used to describe a heatmap. Use the seaborn.heatmap() method to create heatmaps

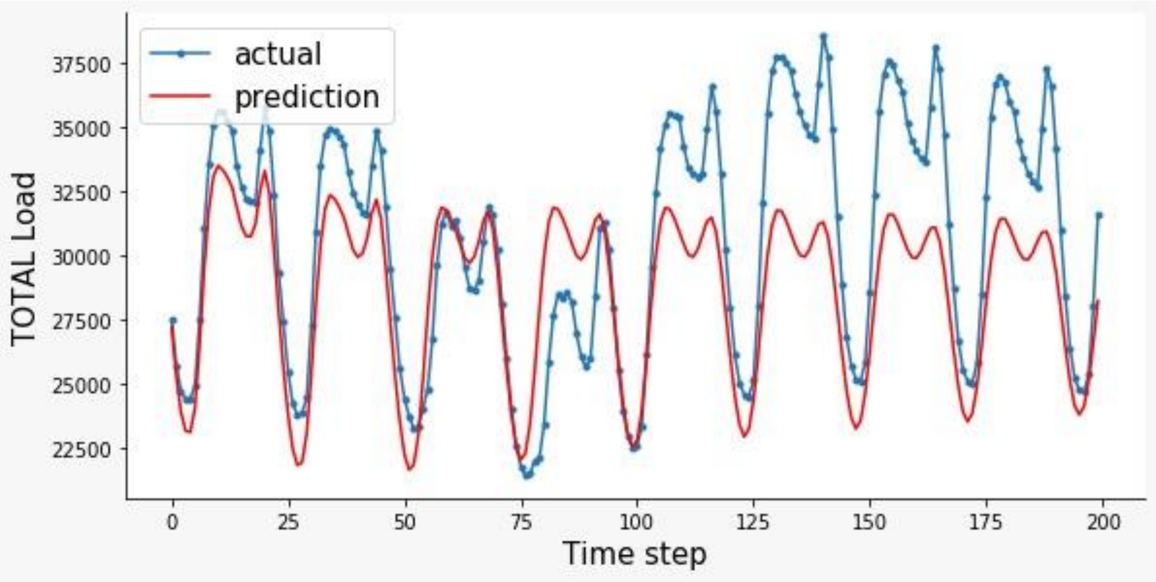


Fig 5.9 Prediction 1 Failed

Fig 5.9 This graph represents the actual and predicted data points with total load and time step at the x and y axis. As we can see this plotted graph is not coinciding with not a good sign. So, we should try again with different data points until the graphs best fit the line.

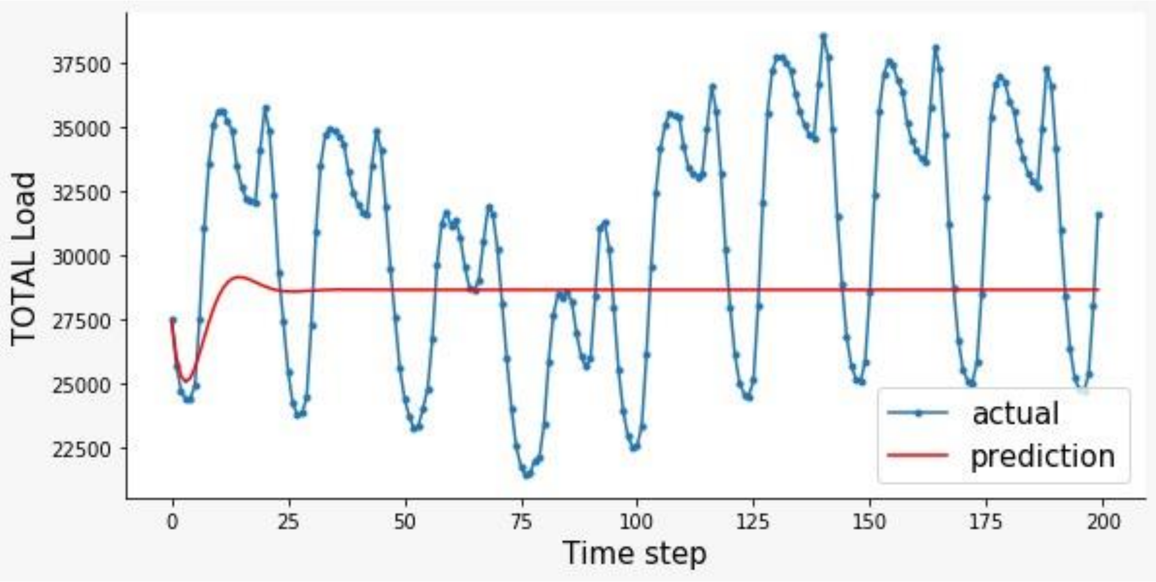


Fig 5.10 Prediction 2 Failed

Fig 5.10 This is the second graph and the prediction data points and line are not merging with the actual line even after the change in data is not working. So, we need to try other methods for following process.

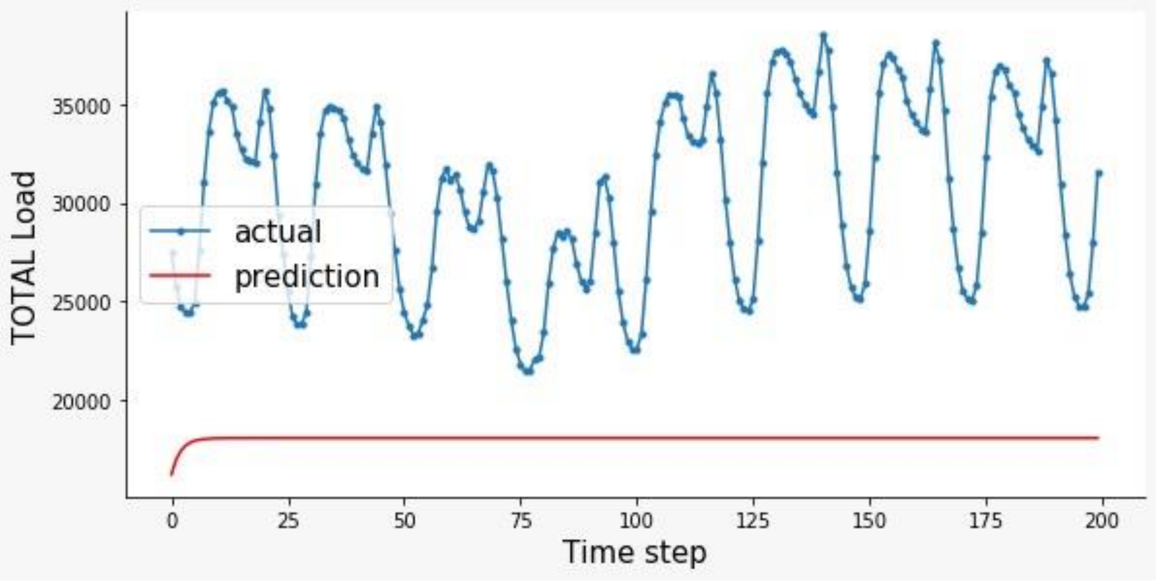


Fig 5.11 Prediction 3 Failed

Fig 5.11 This is the third graph with wrong prediction visualization after applying RMS that is Root Mean Square method will use a formula to calculate between the data points. The verdict is not successful. let’s try one more time.

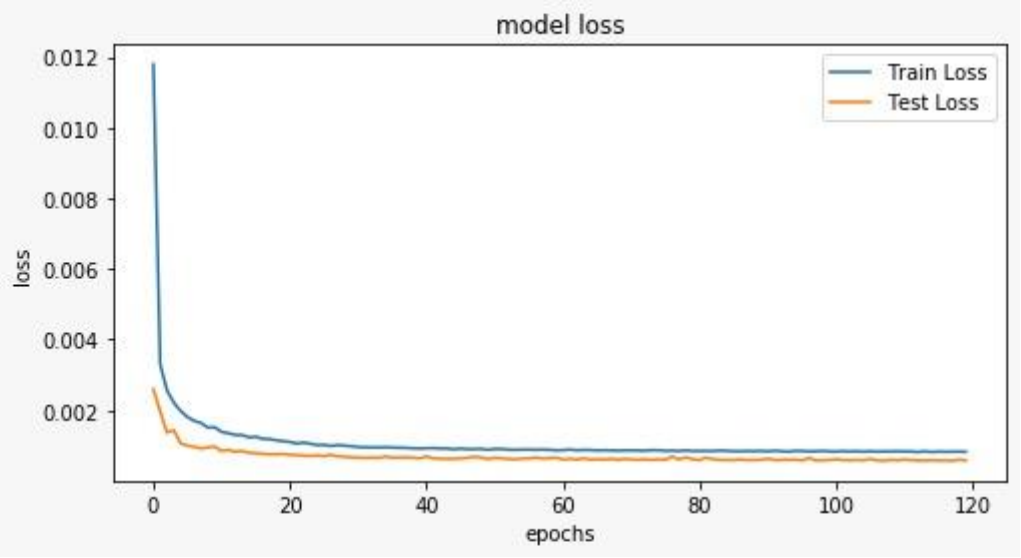


Fig 5.12 Model loss graph

Fig 5.12 This graph represents the train and test loss in the model after splitting the data sets into train set and test sets with 60% and 40% respectively. Initially, there is a significant increase, but as time passes, there is a less straight line with less than 0.02 units of loss for each unit for epochs, suggesting that there may be a difference between prediction and actual graph.

CHAPTER 6

TESTING

6.1 Introduction

Unit testing frameworks are generally used to make it easier to develop unit tests. The inherent compiler in most programming languages does not provide unit testing. Unit testing made much more enjoyable by using third-party commercial and open-source software.

For various programming languages, tools for unit testing are listed below:

- Java framework Tool for unit testing in unit
- PHP framework Tool - PHPUnit
- C++ frameworks Tool - Unit Test++ and Google C++
- .NET framework Tool - Nunit
- Python framework Tool - py.test

6.2 Test Driven Development

TDD is a programming technique in which a prior to developing any actual production code, a programmer creates a test, and then produces the code to pass that test. The notion is that with just a smidgeon of confidence from that initial test, the programmer can feel free to rework and refactor some more until they have the cleanest code they can write. The concept is simple, but implementation, like most simple things, is difficult. TDD necessitates an entirely different mindset than most people are accustomed to, as well as the perseverance to deal with a steep learning curve. The following are some instances of functional testing:

6.3 Unit Testing

Testing of individual software components or units is known as "Unit Testing." A primary aim is to make sure that each piece of software functions as it was designed to. A unit is the smallest piece of software that can be tested. In most cases, it just has one input and one output. Any one of a number of different types of objects may serve as a component in procedural programming. The base unit in object-oriented computing is a method, which may correspond to a base/super, abstract, or derived/child class. (An application's module is seen by some as a single entity. This is a bad idea since that module is likely to include a large number of separate components. Unit testing can be made easier with the usage of frameworks, drivers, stubs, and mock/fake objects.

There are many different types of units, including lines of code, procedures and classes. In general, smaller is preferred. To get a clearer understanding of your code's performance, you should run smaller tests. Practically speaking, you can run thousands of tests in a second with extremely tiny units.

Unit testing is unimportant to black box testers. Their major purpose is to check the application for compliance with the requirements without getting into the implementation specifics.

There is nothing new about the idea of doing unit tests inside a larger application. As far back as there has been computer programming, it has been around. In order to ensure that the code used to fulfil functional requirements is of the highest possible quality, unit tests are often created by developers and, on rare occasions, by outside White box testers.

6.4 Blackbox Testing

It is always a good idea to start with some background information. The waterfall was the dominating impact of software development thought when I first learnt about integration testing in the 1980s. A design phase would be included in a bigger project to establish the interface and behavior of the various system elements. After then, developers would be assigned to programme the modules. It was not uncommon for a single programmer to be in charge of a single module, but this one would be large enough that its development may take months. All of this work was done in isolation, and once the programmer was satisfied that it was complete, it was handed over to QA for testing.

6.5 Integration Testing

A kind of software testing known as Integration Testing integrates and tests various modules together. In order to find problems with the way the various components work together, this degree of testing is used. With the use of drivers and test stubs, integration testing may be done more quickly and efficiently.

An integration test looks for problems in the way linked systems and components interact and communicate with one another. The words "integration testing of components" and "system integration testing" might be interchangeable.

Verification of component integrity in order to discover interface and component interaction issues, testing was carried out. Testing the system's integrity Integration of system and modules is being tested, as well as external interfaces (e.g. Electronic Data Interchange, Internet).

Integration testing examines the interactions between components, such as the operating system, file system, and hardware, as well as the interfaces between systems. Software testing includes integration testing

Integration tests are used to ensure that separate software parts work together properly. Even by the amorphous standards of the software business, the term has grown muddled, so I've been hesitant to use it in my work. Many people believe that integration tests must be extensive in scope, but in fact they can be done more successfully with a smaller scope. Integration testing examines the interactions between components, such as the operating system, file system, and hardware, as well as the interfaces between systems. Software testing includes integration testing.

6.6 System Testing

A subset of software testing known as "system testing" focuses on the complete software architecture as a whole. In order to evaluate whether the systems are up to snuff, this test is conducted. The whole system must be tested during system testing. All of the packages of the system are joined together in order to verify that it operates as planned.

Next, Once Integration Testing is complete, System Testing begins. This is essential if you want to make something of the highest quality. Testing a completely integrated software product or system using pre-determined standards and functional criteria is known as system testing. It answers the question, "Does the whole system perform in line with its pre-defined requirements?”

This kind of testing is known as "black box" testing, which evaluates just the software's outward working features. It does not necessitate any internal knowledge of coding, programming, design, or the like, and is entirely user-centered.

System testing, which is a sort of black box testing, is the first technique for thoroughly testing a software product. As part of this system testing, the integrated system is examined to see whether it meets the client's requirements.

Systems are rigorously tested in this phase to make sure they fulfil all of the client's requirements, as described in their functional specification or system specification paperwork, before being delivered. Integration testing is often used as a complement to this kind of testing since it covers the whole system's routine. In order to do this kind of testing, a specialised Test Plan and additional documentation generated from the system requirement specification are required, which should include both software and hardware requirements. We use this test to discover the mistakes. It ensures that the whole system works as expected. System performance and functionality are tested to guarantee a high-quality product is delivered to the customer. A "system test" is a word used to describe the process of evaluating the complete system. This testing examines the whole client experience from start to finish. System testing also performs non-functional and functional tests. To guarantee that the system is bug- and defect-free, the development team does all they possibly can. Hardware and software requirements are also tested during system testing. There are many other types of system testing, however this one is more specialized.

6.7 Sanity Testing

When a QA doesn't have the time to execute all of the test cases, whether it's Functional Testing, UI, OS, or Browser Testing, sanity testing is used. Sanity testing includes regression testing. Sanity testing is achieved after getting the software build to confirm that the code modifications are working as planned. The purpose of this testing is to determine whether or not the build's testing can proceed. It is the primary objective of this testing to make sure that the product is safe to use the suggested improvements or functionality works as planned. The build is rejected by the testing team if the sanity test fails, saving time and money. It's only done once the build has passed the smoke test and been approved for additional testing by the Quality Assurance team. During this testing procedure, the team's primary focus is on validating the application's functionality rather than performing extensive testing.

In order to establish whether or not the build we got from the developers can be tested, we employ Smoke Testing to do so. The A Day check is another name for this kind of payment. In terms of the construction, it's done.

It's better to avoid testing the whole programme if the most important features do not function or the most critical issues haven't been resolved. Focus here is on the basic and fundamental proposed enhancement.

We do not use test cases when performing smoke examining. We just select the required test cases from a list of test cases that have already been written. As previously stated, the core application work flow will be the primary focus of our Smoke Testing efforts. As a result, for each application we choose a subset of our whole testing library of test cases. We select the smallest amount of test cases that can be completed in under half an hour.

Sanity testing's major goal is to ensure that the planned functionality works as expected. Sanity testing is carried out instead of full regression testing

If the build fails, sanity tests help to prevent wasting time and money on testing. When a build fails, the tester should reject it. After regression testing is completed, sanity testing is performed to ensure that the defect fixes and changes made to the software programme do not break the software's core functionality. This is usually done near the end of the SDLC, i.e. when the software is released. Sanity testing might be considered a subcategory of acceptance testing. Sanity testing is also known as Tester Acceptance Testing.

6.8 Regression Testing

Regression testing is a kind of software testing done to assure change has no effect on the product's current functionality. This is to guarantee the quality of the final product is compatible includes the addition of any new features, the release of any bug fixes, or the improvement of any already existing functions. Re-running previously completed test cases is used to confirm the effect of the modification.

When the original functionality is significantly altered, performing this test on a fresh build is feasible, even if the fix only addresses a single problem issue. With the right approach, regression testing may help guarantee that your software is free of bugs and errors by combining it with other types of testing, such as front-end UI automation and focused unit testing, all of which can be prioritized according to the importance of the risks involved. Workflow approaches like XP, Scrum, RUP, or a combination of these have made regression testing an essential part of dynamic, iterative deployment plans. Your company may use any method for software development and quality assurance, but taking the time to devise an effective testing strategy that includes automated regression tests at its core can help you avoid overspending, keeping your team on track to meet deadlines, and protecting your goods and company's bottom line against unanticipated problems, whatever approach you choose.

CHAPTER 7

RESULT ANALYSIS

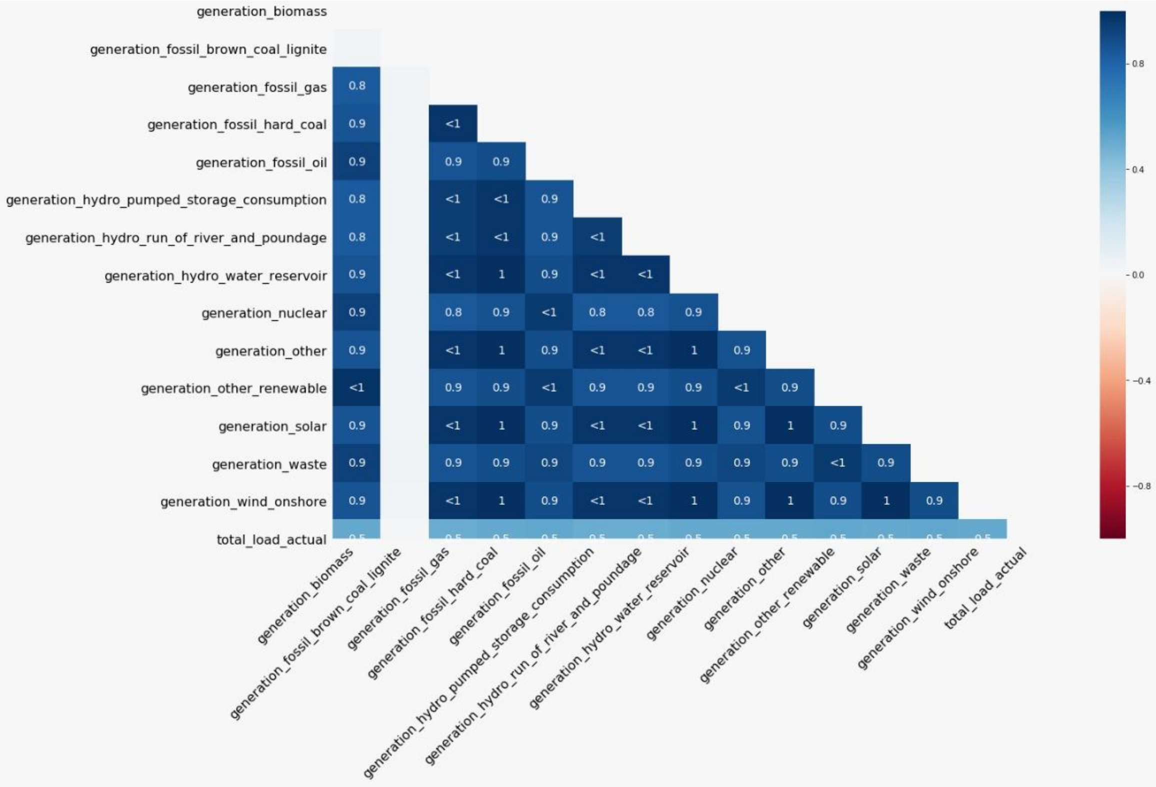


Fig 7.1 Heat map of contents

Fig 7.1 Two variables are displayed on two axes in a heatmap to demonstrate the connection between them. There may be patterns in the values of one or both variables by monitoring how cell colours vary throughout each axis marked down as in a histogram. A correlogram is a kind of heat map in which each of the variables on the two axes of a correlogram is replaced by one of the numeric variables in the dataset. Using a simple significant association, each cell represents the connection between the crossing variables. As an alternative to straightforward correlations, scatter plots may be used to depict relationships.

In the above heat map figure, different sources are mentioned for energy generation namely,generation_biomass has a very light shade and a value of 0 at the start. Secondly, generation_fossil_brown_coal_lignite is a source of power with an average value and a slightly dark blue grade. And the same is followed by the remaining contents in the heat map. The figure's final verdict is to find a correlation between each component and verify it.

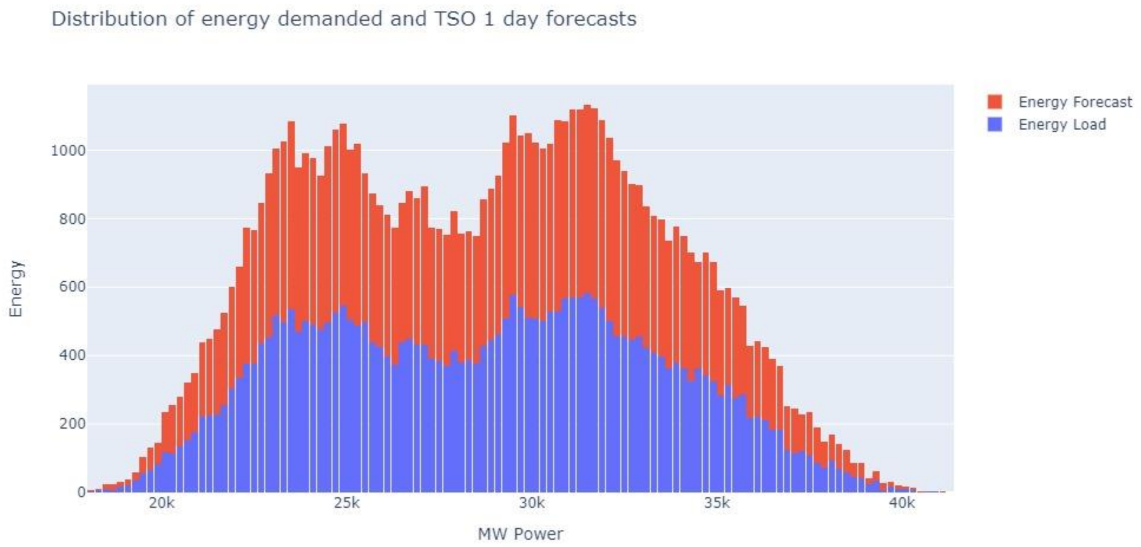


Fig 7.2 Graph between Input and Output

Fig 7.2 Using a typical bar chart to compare numerical values across different levels of a category variable is the primary goal of this kind of charting. The length of each bar indicates the numerical value of the category variable. A stacked bar chart does the same thing, but it also has a secondary aim in mind.

The relative breakdown of each main bar according to the levels of a second category variable necessitates the use of a stacked bar chart. A secondary bearing is represented by a series of sub-bars, one for each of the several levels of the main bar. Each stacked bar has the same total length.

In the above graph, there are two axes represented by energy and MW power. There are two bar graphs merged for the same constants known as stacked bar graphs, which are formed by plotlab and matpolt libraries, energy forecast and energy load, which are input and output data. We can see that forecast data points are higher than load data points, which is nearly a double difference.

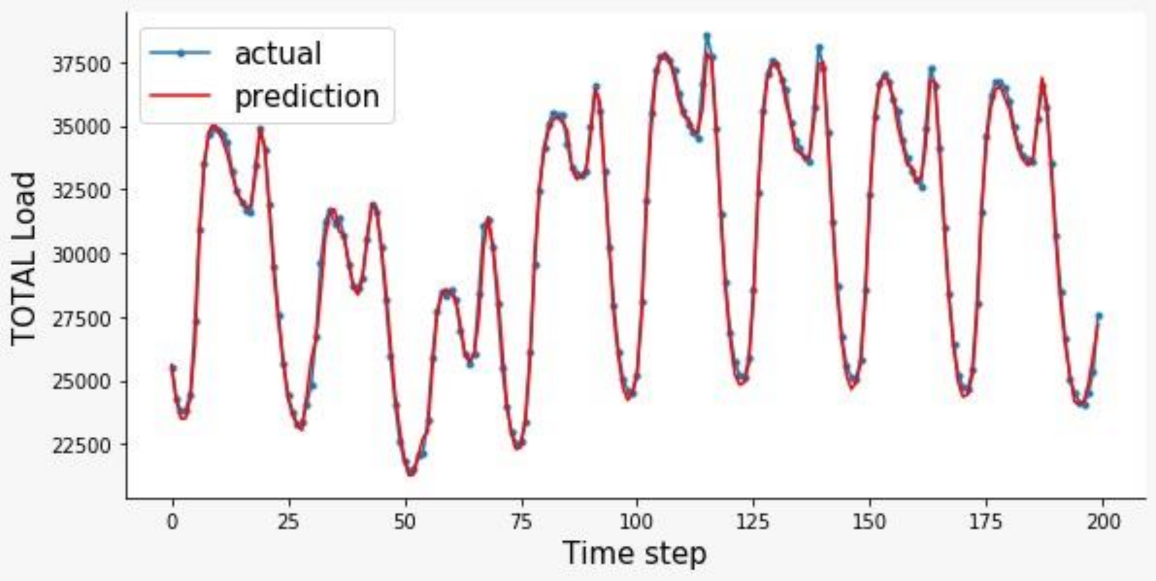


Fig 7.3 Prediction 5 Passed with 98% of accuracy

Fig 7.3 This is Matplotlib line graph with coinciding graph lines and data points between Total Load and Time step with two types one is blue and another is red which is actual and prediction respectively. After 4 failed attempts of plotting the result graph deleting unwanted piece of information helped a lot but using Root Mean Square method has major asset in solving and getting the accurate result as shown in the above graph which is around 98%.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

One-hour-resolution energy usage data for many kinds of buildings in the City was used to train and test the proposed LSTM-based system. MLP, RF, and SVM are also built and evaluated analysing the same set of data to see how well the proposed system performs in comparison to others. The suggested LSTM-based model in nine out of twelve months, it was more accurate than the comparison models in predicting building electricity use, according to the week-ahead forecasting results. A LSTM network-based short-term electrical load forecast approach is suggested. Uses daily load profiles, temperature and humidity data to construct embedding vector that describe the profiles' fundamental properties and correlations.

8.2 Future Enhancements

Furthermore, in future work, we will attempt to perform the model utilising a variety of datasets. For each sort of customer, a distinct form of data analysis is required (ex: home, public facility, etc.). In the same situation, we'll concentrate on improving the model's performance in order to produce better experimental results in the future.

REFERENCES

[1] P. Mandal, A. U. Haque, J. Meng, A. K. Srivastava, and R. Martinez, "A novel hybrid approach using wavelet, rey algorithm, and fuzzy artmap for day-ahead electricity price forecasting," IEEE Transaction on Power System, vol. 28, no. 2, pp. 10411051, May 2013.

[2] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," International Journal of Forecasting, vol. 32, no. 3, pp. 896913, July 2016.

[3] H. C. Wu, S. C. Chan, K. M. Tsui, and Y. Hou, "A new recursive dynamic factor analysis for point and interval forecast of electricity price," IEEE Transaction on Power System, vol. 28, no. 3, pp. 2352 2365, August 2013.

[4] T. Christensen, A. Hurn, and K. Lindsay, "Forecasting spikes in elec- tricity prices," International Journal of Forecasting, vol. 28, no. 2, pp. 400 411, June 2012.

[5] X. Lu, Z. Y. Dong, and X. Li, "Electricity market price spike forecast with data mining techniques," Electric Power Systems Research, vol. 73, no. 1, pp. 1929, January 2005.

[6] S. K. Aggarwal, L. M. Saini, and A. Kumar, "Electricity price fore- casting in deregulated markets: A review and evaluation," International Journal of Electrical Power & Energy Systems, vol. 31, no. 1, pp. 1322, January 2009

.

[7] H. Zareipour, C. Canizares, and K. Bhattacharya, "Economic impact of electricity market price forecasting errors: A demand-side analysis," February 2010.

[8] E. Nasrolahpour, S. J. Kazempour, H. Zareipour, and W. D. Rosehart, "Strategic sizing of energy storage facilities in electricity markets," October 2016.

[9] S. Shaee, P. Zamani-Dehkordi, H. Zareipour, and A. N. knight, "Eco- nomic assessment of a price-maker energy storage facility in the Alberta electricity market," Energy, vol. 111, pp. 537547, September 2016.

[10]J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2006.

[11] D. H. Vu, K. M. Muttaqi, and A. P . Agalgaonkar, "Combinatorial approach using wavelet analysis and artificial neural network fo r short- term l o ad forecasting," i n Power Engineering Conference (AUPEC), 2014 Australasian Universities, 2014, pp. 1-6.

[12] C. E. Shannon, "A Mathematical Theory of Communication," The Bell System Technical Journal, vol. 27, pp. 379423, 623656, 1948.

[13] Z. Dawy, B. Goebel, J. Hagenauer, C. Andreoli, T. Meitinger, and J. C. Mueller, "Gene mapping and marker clustering using Shannons mutual information," IEEE/ACM Transactions o n Computational Biology and Bioinformatics, vol. 3, no. 1, pp. 47-56, 2006.

[14] Z. Yang, L. Ce, and L. Lian, Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods, Appl. Energy, vol. 190, pp. 291305, Mar. 2017.

[15] S. K. Aggarwal, L. M. Saini, and A. Kumar, Electricity price forecasting in deregulated markets: A review and evaluation, Int. J. Electr. Power Energy Syst., vol. 31, pp. 1322, Jan. 2009.

[16] M. Cerjan, I. Krelj, M. Vidak, and M. Delimar, A literature review with statistical analysis of electricity price forecasting methods, in Proc. IEEE EUROCON, Jul. 2013, pp. 756763

Format - 1

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u / s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION / PROJECT REPORT FOR UG / PG PROGRAMMES
(To be attached in the dissertation / project report)

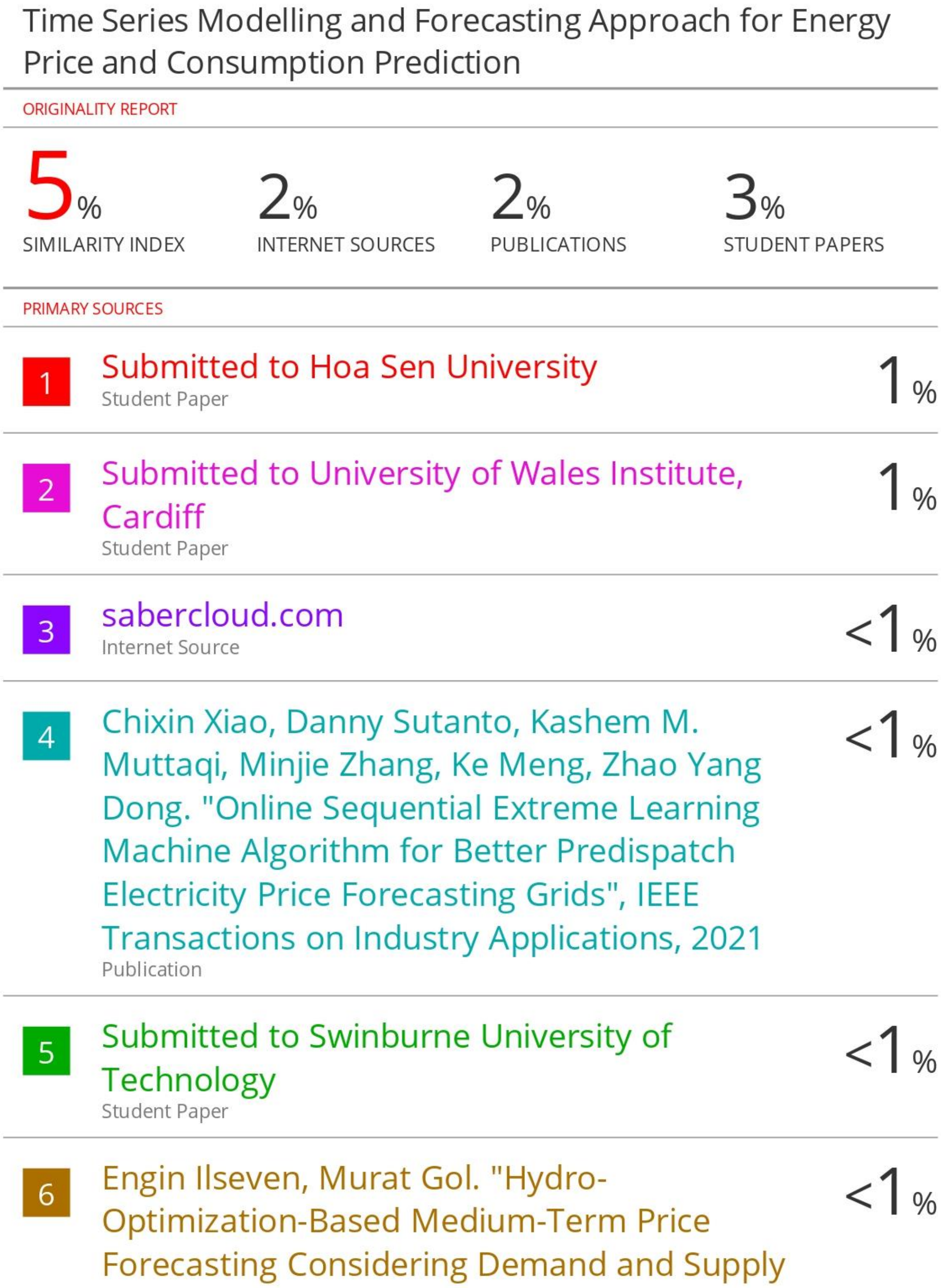
1	Name of the Candidate **(IN BLOCK LETTERS)**	G.V.SAI VARUN, M. SATHVIK, K.YASWANTH KUMAR

2	Address of Candidate	9, Hi-Tech Road Ayodhya Nagar,Hyderabad 500001;
		3-145-6-15, Sapthagiri Nagar,Madanapalli, A.P, 517325;
		3/84 A Venkataramana Suppliers, Guntur, A.P, 522235
		Mobile Number: 7382035797, 8019993363, 7539996467

3	Registration Number	RA1811003020102, RA1811003020101, RA1811003020072
4	Date of Birth	27/03/2001, 17/05/2001, 09/08/1999
5	Department	Computer Science and Engineering
6	Faculty	Faculty of Engineering and Technology
7	Title of the Dissertation / Project	Time Series Modelling And Forecasting Approach For Energy Price And Consumption Prediction
8	Whether the above project / dissertation is done by	Individual or group : Group (Strike whichever is not applicable) a) If the project / dissertation is done in group, then how many students together completed the project : #03 b) Mention the Name & Register number of other candidates : G.V.SAI VARUN [RA1811003020102] M.SATHVIK [RA1811003020101] K.YASWANTH KUMAR [RA1811003020072]
9	Name and address of the Supervisor / Guide	Dr. S. LILLY SHEEBA, Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai 89 **Mail ID: lillyshs@srmist.edu.in** **Mobile Number : 9486774991**

10	Name and address of the Co-Supervisor / Guide	NA **Mail ID:** NA **Mobile Number:** NA		
11	Software Used	TURNITIN SOFTWARE		
12	Date of Verification	06/05/2022		
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Report**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	Time Series Modelling And Forecasting Approach For Energy Price And Consumption Prediction	NA	NA	5%
Appendices		NA	NA	NA
I / We declare that the above information have been verified and found true to the best of my / our knowledge.				
Signature of the Candidate		Dr. S. LILLY SHEEBA **Name & Signature of the Staff** **(Who uses the plagiarism check software)**		
Dr. S. LILLY SHEEBA **Name & Signature of the Supervisor / Guide**		NA **Name & Signature of the Co-Supervisor / Co-Guide**		
Dr. K. Raja **Name & Signature of the HOD**				

DOCUMENT PLAGARISM REPORT



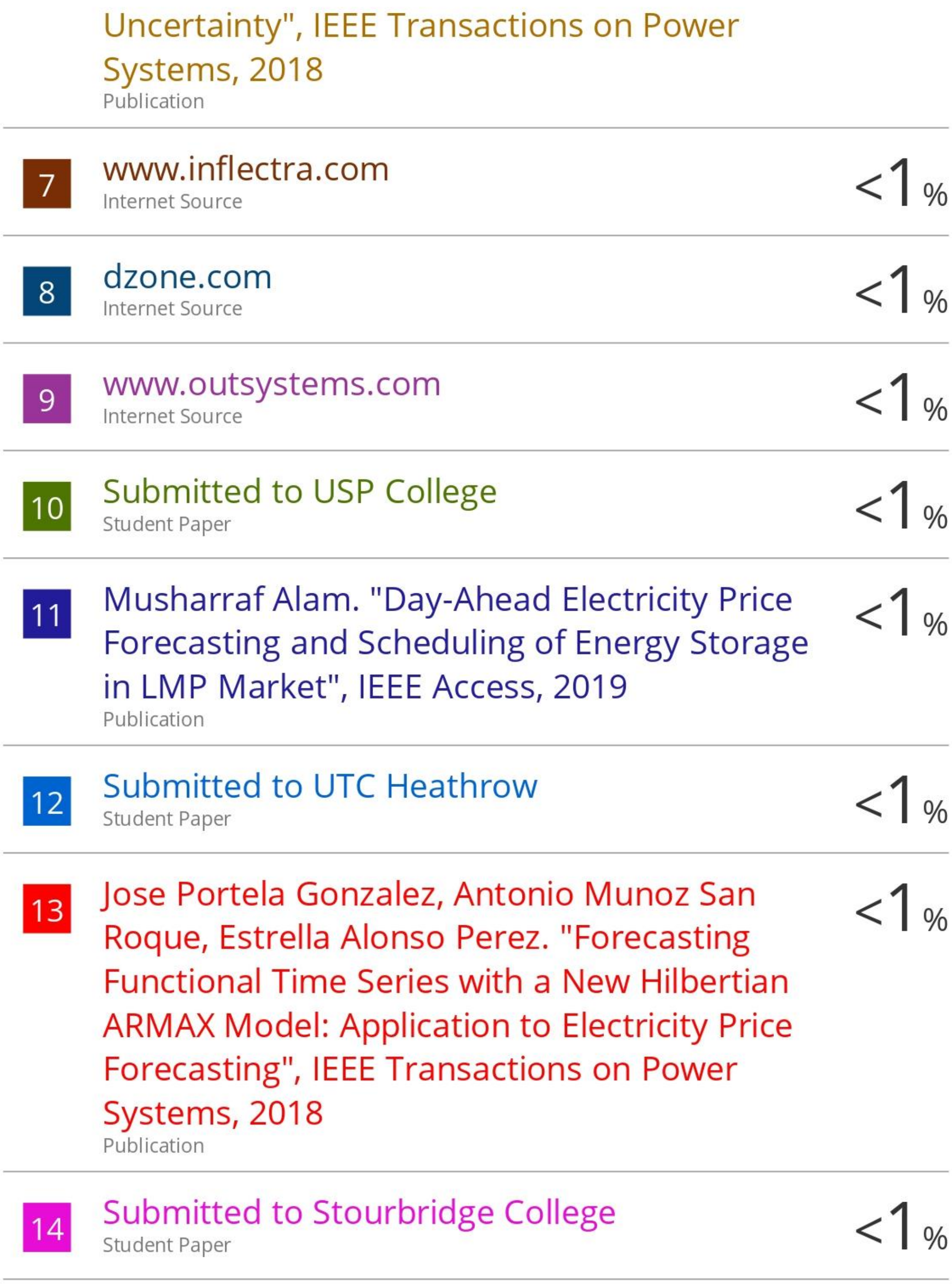
Time Series Modelling and Forecasting Approach for Energy Price and Consumption Prediction

ORIGINALITY REPORT

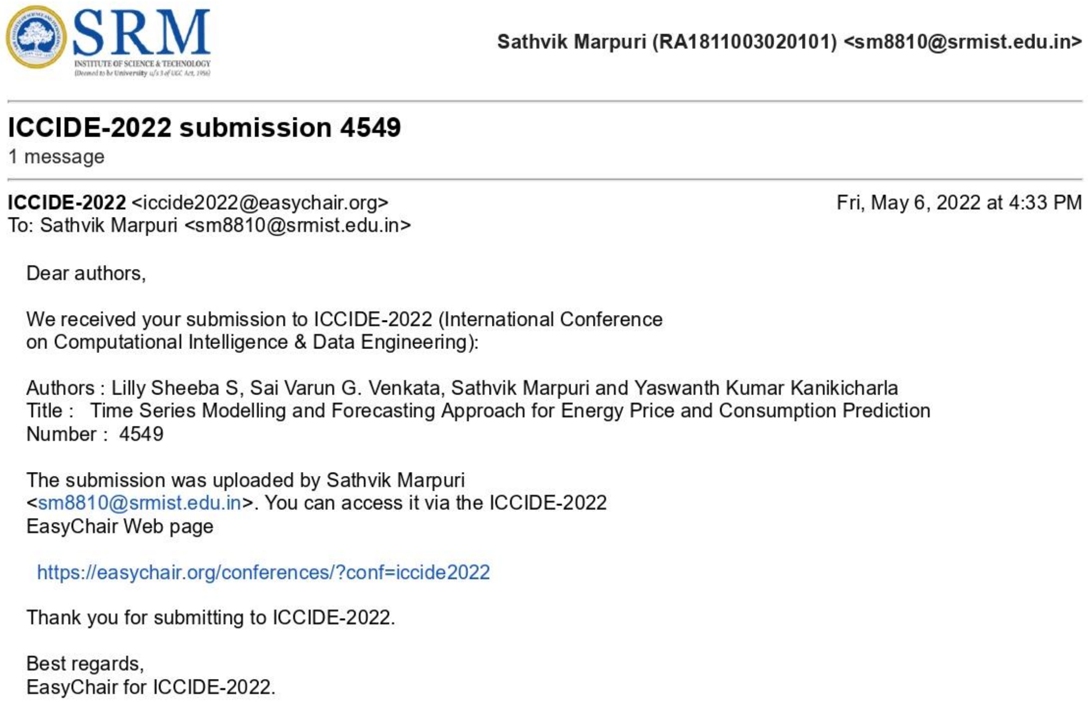
5%	2%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Hoa Sen University Student Paper	1%
2	Submitted to University of Wales Institute, Cardiff Student Paper	1%
3	sabercloud.com Internet Source	<1%
4	Chixin Xiao, Danny Sutanto, Kashem M. Muttaqi, Minjie Zhang, Ke Meng, Zhao Yang Dong. "Online Sequential Extreme Learning Machine Algorithm for Better Predispatch Electricity Price Forecasting Grids", IEEE Transactions on Industry Applications, 2021 Publication	<1%
5	Submitted to Swinburne University of Technology Student Paper	<1%
6	Engin Ilseven, Murat Gol. "Hydro-Optimization-Based Medium-Term Price Forecasting Considering Demand and Supply	<1%



	Uncertainty", IEEE Transactions on Power Systems, 2018 Publication	
7	www.inflectra.com Internet Source	<1 %
8	dzone.com Internet Source	<1 %
9	www.outsystems.com Internet Source	<1 %
10	Submitted to USP College Student Paper	<1 %
11	Musharraf Alam. "Day-Ahead Electricity Price Forecasting and Scheduling of Energy Storage in LMP Market", IEEE Access, 2019 Publication	<1 %
12	Submitted to UTC Heathrow Student Paper	<1 %
13	Jose Portela Gonzalez, Antonio Munoz San Roque, Estrella Alonso Perez. "Forecasting Functional Time Series with a New Hilbertian ARMAX Model: Application to Electricity Price Forecasting", IEEE Transactions on Power Systems, 2018 Publication	<1 %
14	Submitted to Stourbridge College Student Paper	<1 %



SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Sathvik Marpuri (RA1811003020101) <sm8810@srmist.edu.in>

ICCIDE-2022 submission 4549

1 message

ICCIDE-2022 <iccide2022@easychair.org> Fri, May 6, 2022 at 4:33 PM
To: Sathvik Marpuri <sm8810@srmist.edu.in>

Dear authors,

We received your submission to ICCIDE-2022 (International Conference
on Computational Intelligence & Data Engineering):

Authors : Lilly Sheeba S, Sai Varun G. Venkata, Sathvik Marpuri and Yaswanth Kumar Kanikicharla
Title : Time Series Modelling and Forecasting Approach for Energy Price and Consumption Prediction
Number : 4549

The submission was uploaded by Sathvik Marpuri
<sm8810@srmist.edu.in>. You can access it via the ICCIDE-2022
EasyChair Web page

https://easychair.org/conferences/?conf=iccide2022

Thank you for submitting to ICCIDE-2022.

Best regards,
EasyChair for ICCIDE-2022.