AN OPTIMIZED VARIANT OF MACHINE LEARNING ALGORITHM FOR DATA-DRIVEN ELECTRICAL ENERGY EFFICIENCY MANAGEMENT (D2EEM)

SHAMIM AKHTAR

DOCTOR OF PHILOSOPHY

UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : SHAMIM AKHTAR

Date of Birth : 01 APRIL 1974

Title : AN OPTIMIZED VARIANT OF MACHINE LEARNING ALGORITHM FOR DATA-DRIVEN ELECTRICAL ENERGY EFFICIENCY MANAGEMENT (D2EEM)

Academic Session : Semester II 2023/24

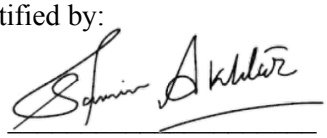
I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

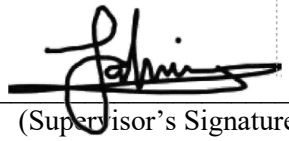
I acknowledge that Universiti Malaysia Pahang Al-Sultan Abdullah reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang Al-Sultan Abdullah
2. The Library of Universiti Malaysia Pahang Al-Sultan Abdullah has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)



(Supervisor's Signature)

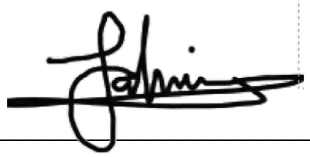
Passport Number
Date: 26/06/2024

Name of Supervisor
Date: 26/06/2024

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

We hereby declare that We have checked this thesis, and, in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy.



(Supervisor's Signature)

Full Name : Assoc. Prof. Ir.Dr Muhamad Zahim bin Sajod

Position : Assoc.Prof.

Date : 26 June 2024

(Co-supervisor's Signature)

Full Name : Dr. Sayed Sajjad Hussain Rizwi

Position : Associate Professor

Date : 26 June 2024

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang Al-Sultan Abdullah or any other institution.

(Student's Signature)

Full Name : SHAMIM AKHTAR

ID Number : PES19002

Date : 26 June 2024

AN OPTIMIZED VARIANT OF MACHINE LEARNING ALGORITHM FOR DATA-DRIVEN ELECTRICAL ENERGY EFFICIENCY MANAGEMENT (D2EEM)

SHAMIM AKHTAR

Thesis submitted in fulfillment of the requirements

for the award of the degree of

Doctor of Philosophy

Faculty of Electrical and Electronics Engineering Technology

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

JUNE 2024

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis guide Dr. Muhamad Zahim Bin Sujod for their guidance, advice, and constant support throughout my research work. I would like to thank her for being my advisor here at Universiti Malaysia Pahang. I consider it my good fortune to have got an opportunity to work with such a great researcher and mentor.

I am heartily grateful to our department head "Prof. Ir. Ts. Dr. Hamdan Bin Daniyal ", for his moral support and all the facilities I provided for doing my research work under such a great team. In addition to my official supervisors, I have also had a field supervisor, the most important one being Dr. Syed Sajjad Hussain Rizvi thank you for allowing me to participate in research and always taking on time to discuss knowledge and life in usual with me. Your enthusiasm for research is extremely inspiring, and you were the first person to enthuse me sufficiently to be a decent researcher.

I am especially obligated to my parents, wife and children because of their love and support I can get my PhD.

Finally, I would also like to thank my friend Irshad Ahamad. He spent lots of time, suggestions, inspiration, and cooperation developing this thesis.

ABSTRAK

Elektrik adalah sumber tenaga yang paling diminta di seluruh dunia. Pada saat yang sama, ia terhad secara kritikal untuk memenuhi permintaan. Terdapat hanya dua penyelesaian logik untuk memenuhi permintaan ini. Pertama, meningkatkan kapasiti pengeluaran kuasa, meningkatkan teknologi transmisi, meningkatkan kecekapan pengeluaran kuasa. Kedua, mengurus penggunaan tenaga di premis. Kajian ini terutamanya memberi tumpuan kepada pengurusan kecekapan tenaga elektrik berdasarkan data menggunakan kecerdasan buatan. Khususnya, kampus universiti dipilih sebagai kajian kes dalam penyelidikan ini. Ia merupakan fakta yang sudah mapan bahawa pembelajaran mesin lebih unggul dari segi ramalan dan klasifikasi. Oleh itu, dalam kajian ini, satu variasi teroptimum baru daripada algoritma pembelajaran mesin dikemukakan. Dalam kajian ini, satu set data rujukan tentang penggunaan tenaga di kampus universiti IIT, India (disediakan oleh Smart Energy Informatics Lab, SEIL) dipilih untuk latihan dan pengujian variasi algoritma pembelajaran mesin yang dicadangkan. Selain itu, prestasi yang sama juga disahkan di kampus universiti lain dengan budaya yang seangkatan. Dalam kaitannya ini, set data yang disediakan oleh Energy Informatics Group Department of Computer Science, SBASSE Lahore University of Management Sciences, Pakistan dipilih. Skop kajian ini adalah tiga kali lipat. Pertama, satu kajian bandingan yang menyeluruh dan parametrik pada pelbagai jenis algoritma pembelajaran mesin dikemukakan untuk menilai prestasi algoritma pembelajaran mesin dalam ramalan beban tenaga. Hasil daripada fasa ini adalah pemilihan calon terbaik bagi algoritma pembelajaran mesin untuk ramalan beban tenaga kampus universiti. Kedua, adalah pengoptimuman algoritma pembelajaran mesin terbaik yang dipilih untuk meningkatkan lagi kecekapan dan keberkesanan ramalan. Akhirnya, algoritma-algoritma yang dicadangkan juga disahkan pada set data lain dari kampus universiti di rantau yang berbeza. Kajian ini mengesyorkan kompromi pemilihan sebagai fungsi keberkesanan dan kecekapan ramalan algoritma. Khususnya, Bagged Trees yang dioptimumkan adalah algoritma yang paling berkesan untuk aplikasi ramalan permintaan tenaga, manakala Medium Trees yang dioptimumkan adalah algoritma yang paling cekap untuk sistem masa nyata. Begitu juga, Fine Trees yang dioptimumkan mempunyai kompromi optimum antara keberkesanan dan kecekapan.

ABSTRACT

The electricity at the most demanded energy source around the globe. At the same time, it is critically limited to meet the demand. There are only two logical solutions to meet this demand. First, to increase the power generation capacity, enhance transmission technology, and improve power generation efficiency. The second, is to manage the energy utilization in the premises. Since the electrical energy consumption is different in each application and management of energy utilization in large scale is complex, therefore this study proposed (Data-driven electrical energy efficiency management) D2EEM using optimized ML. This research is mainly focused on data-driven electrical energy efficiency management using artificial intelligence. Particularly, a university campus is selected as a case study in this research. It is a well-established fact that machine learning is outperforming in terms of prediction and classification. Therefore, in this study a new optimized variant of machine learning algorithms is presented. In this study, a benchmark dataset of energy consumption in a university campus of IIT, India (provided by the Smart Energy Informatics Lab, SEIL) was selected for training and testing the proposed variants of machine learning algorithms. The scope of this study is tri folded, First, an exhaustive and parametric comparative study on a wide variety of machine learning algorithms is presented to evaluate the performance of machine learning algorithms in energy load prediction. The deliverable of this phase is the selection of the best candidate of machine learning algorithm for university campus energy load prediction. The second is the optimization of the best selected machine learning algorithms to further improve the efficiency and efficacy of the prediction. Finally, the proposed algorithms were also validated on another dataset of a university campus in a different region. This study recommends a selection trade-off as the function of prediction efficiency and efficacy of the algorithm. Particularly, the proposed optimized Bagged Trees are the most effective algorithm for energy demand prediction applications, and the proposed optimized Medium Trees are the most efficient algorithm for real-time systems. Likewise, optimized Fine Trees have the optimum trade-off between efficacy and efficiency.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
AI	Artificial Intelligence
NN	Neural Networks
SVM	Support Vector Machines
RF	Random Forest
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Networks
DNN	Deep Neural Networks
NLP	Natural Language Processing
PCA	Principal Component Analysis
KNN	K-Nearest Neighbors
DT	Decision Trees
SGD	Stochastic Gradient Descent
EM	Expectation-Maximization

CHAPTER 1

INTRODUCTION

1.1 Electrical Energy Management System

In the current trend, the electric power management system has received considerable attention. The path towards a sustainable energy for society is at the forefront of public interest and is a high priority of policymakers worldwide. The human ability to harness of electrical energy has resulted in the continuous transformation of civilization. An expanding portion of the global population is now able to heat and light their homes, fertilize, and irrigate their crops, communicate with one another, and travel around the globe (Kabeyi and Olanrewaju 2022). All this development is made possible by the ever-improving dexterity of human beings in locating, obtaining, and using electrical energy. Scientific research assists in advancing a sustainable future based on clean electrical energy generation, transmission & distribution, electrical & chemical energy storage, electrical energy efficiency, and improved electrical energy management systems. Electrical Energy Management can be broadly understood as the proactive, planned, and systematic control of electrical energy use in a facility or organization to meet environmental and economic needs (Infield and Freris 2020). In simple words, Electrical Energy Management is the practice of maximizing energy use for the optimum results while also taking action to conserve it. Many robust and commercial solutions have dealt with the scarcity of electrical energy. It includes, but is not limited to, efficient generation of electrical energy (Beér 2007), alternate energy sources ((Stathis) Michaelides 2012), and Energy Management System (EMS) (C. Chen et al. 2011). As per the identification of the researchers, EMS is the optimum candidate among others because it is cost-effective, robust, flexible and easy to manage compared to alternative energy generation (Shakir et al. 2014). Therefore, the goal of this procedure is to attain complete environmental sustainability and financial savings. Energy management system is becoming increasingly popular among businesses of all sizes to cut operational expenses.

In such a world where electrical energy costs are set to rise with the growing demand and shrinking supply of non-renewable national resources like coal, saving energy makes good business sense. The fundamental guidelines that are typically followed for electrical energy management include, but are not limited to, gathering data on electrical energy usage, and measuring it, looking for ways to save electrical energy, putting those ideas into practice, and keeping track of progress and ongoing improvements. The EMS can be deployed on both small- and large-scale levels.

However, keeping in view the fact that the electrical energy consumption profile and consumption patterns differ for each application. Therefore, the intensive level of customization, is a pressing need of the time (Mohajeryami et al. 2016). For this purpose, the researchers soon identified artificial intelligence to be customised. This thesis is presenting an intelligent data-driven approach for electrical energy load management using machine learning algorithms. This study facilitates the researchers and industry experts in the field of computing and engineering sciences and many other firms related to electrical energy management.

1.2 Benefits of the electrical energy management system (EEMS)

From educational institutions to industrial buildings, reducing facility operational costs has become a big challenge in today's world. One cannot imagine daily life without electricity, but since consumption increases, so do the prices. This is where Electrical Energy Management System comes in. An EMS system tracks, regulate, and improves electrical energy transmission and use. Ultimately, EMS is the key to essential energy and cost saving. Electrical Energy Management solutions are typically much more cost-effective for factories and businesses to operate than those that do not use them. The company's entire operation is examined by EMS, which then optimizes it to use less electrical energy.

The bottom line is immediately impacted by the savings produced by the adoption of electrical energy management technologies. The following list includes some of the main benefits of electrical energy management systems:

- Reduction in Electrical Power Usage: Reducing power use and utilizing electrical energy management systems will lead to more ecologically friendly procedures.
- Reduction in Electrical Energy Consumption: Reducing electrical energy consumption through process optimization and efficient electrical energy load planning increases the overall productivity of industrial operations and allows businesses to catch up with their competitors through continual process improvement.
- Decrease in Carbon Emissions: Using energy management techniques results in a considerable decrease in carbon emissions and consumption.
- Increase in Property Value: Owning the energy management system increases the property value.
- Reduction in Electricity Bills: Electrical energy management systems (EEMS) are one of the most widely advocated solutions for reconciling electricity demand with limited electricity resources. Furthermore, these systems contribute to a significant reduction in electricity consumption bills.

1.3 Different strategies for energy management

The electrical energy demand has skyrocketed with ongoing population and economic growth. The U.S. Energy Information Administration (EIA) has presented a study in which they forecast a 48% increase in global electrical energy demand between 2012 and 2040.

The study reported that if current policy and technology trends continue, global electrical energy consumption and energy-related carbon dioxide emissions will increase through 2050 due to the increasing population and economic growth (Mostafaeipour et al. 2022) as shown in Figure 1.1.

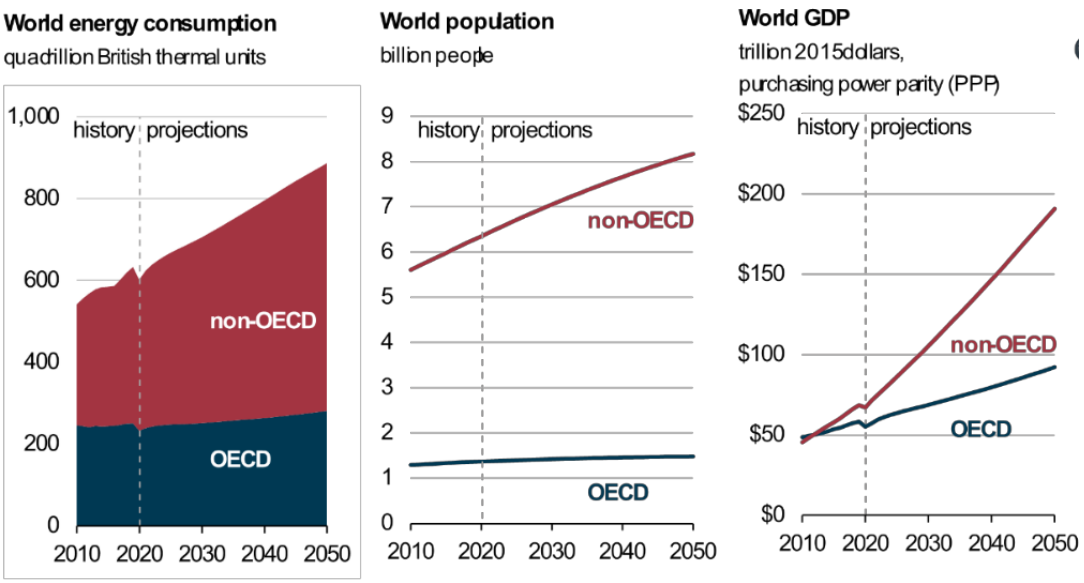


Figure 1.1 Expected Global Energy Usage by 2050
Source : International Energy Outlook 2019

It added that renewables would be the primary source of new electricity generation. Still, natural gas, coal, and, batteries will be used to help meet load and support grid reliability (Nalley and Larose 2021).

There are various strategies and approaches for effective energy management. Some commonly employed strategies include`:

Energy Audits: Conducting thorough assessments of energy usage to identify areas of inefficiency and potential improvements. Energy audits involve analyzing consumption patterns, identifying energy-intensive processes, and recommending energy conservation measures.

Energy Efficiency Measures: Implementing energy-efficient technologies and practices, such as efficient lighting systems, insulation, optimized HVAC systems, and equipment operation.

Demand Response: Participating in demand response programs that allow organizations to adjust their energy consumption during peak demand periods. This strategy involves reducing energy usage or shifting it to off-peak hours in response to grid conditions or utility signals.

Renewable Energy Integration: Incorporating renewable energy sources, like solar panels or wind turbines, to generate clean and sustainable electricity on-site. This reduces dependence on conventional energy sources and can lead to long-term cost savings.

Energy Monitoring and Control: Utilizing advanced energy monitoring systems and smart meters to track real-time energy consumption. This data helps identify waste, detect anomalies, and decide for energy optimization.

Behavioural Changes: Promoting energy-saving behaviours among occupants through awareness campaigns, training, and incentives. Simple practices like turning off lights and equipment when not in use can contribute significantly to energy conservation.

Energy Management Systems (EMS): Implementing comprehensive EMS software that enables centralized control, monitoring, and optimization of energy-consuming systems and devices. EMS can automate energy-saving measures, analyse data, and provide actionable insights for further efficiency improvements.

Energy Procurement Strategies: Exploring alternative energy procurement options, such as power purchase agreements (PPAs) or energy aggregation, to secure energy from renewable sources or at favourable rates.

Continuous Improvement and Monitoring: Regularly evaluating energy management practices, conducting periodic energy audits, and setting targets for energy reduction to ensure ongoing improvement and optimization.

1.3.1 Electrical energy demand forecasting

Electricity today, is regarded as a valuable commodity and the most efficient secondary energy. In recent decades, research on electrical energy consumption issues has grown in importance (Larcher and Tarascon 2015). For society's safety and well-being, electrical energy issues are crucial. According to economic theories, electrical energy is one of the most crucial resources for industrial production, and macro-planning for the industry and electrical energy sectors includes projecting energy use.

The modern world's businesses and civilization rely largely on this resource. Along with other necessary commodities. Electricity serves as a primary source of

survival for human society. Electricity demand forecasting is critical in the electric energy sector since it serves as the foundation for making decisions in electrical power system planning and operation (Soysal and Soysal 2020). Electrical Energy providers use various techniques to forecast electricity consumption. These are used in short-term, medium-term, or long-term forecasting. However, the intricate connections between socioeconomic and meteorological elements lead to electricity consumption. Standard forecasting approaches are inadequate in such a dynamic setting, necessitating more advanced methodologies (Klyuev et al. 2022).

The goal is to sort out all the elements contributing to the demand for change and identify the fundamental causes. Electrical energy demand forecasting is an essential and integral part of the EEMS. It aims to manage, monitor, optimize, and analyse the day-to-day electricity demand of a specific area. (R. Wang, Wang, and Xu 2019). The world's reliance on electrical energy is growing daily, and it can be seen in many (if not all) aspects of human life. It is critical to distribute energy with the least cost and waste. Forecasting consumption load is an important aspect of economic and safety planning electrical power distribution system. Forecasting, estimating, and predicting are marketing terms for having an expected value for demand.

Accurate, robust, adaptive, and efficient electrical energy forecasting promises efficiency in the electrical energy management system. An efficient electrical energy forecasting system complements other energy management policies, optimising energy consumption (Li et al. 2019). This eventually turns into a competitive advantage and sustainable development in general. Recently, researchers have strongly advocated for a data-driven approach to robust, adaptive, and efficient energy forecasting systems (Ahmad et al. 2018).

1.3.2 Motivation

Most people have heard the term "Energy Management" in their lives, especially in recent years, when energy conservation has become increasingly important for the future of companies worldwide. Because of rising fuel costs, increasingly aggressive environmental targets, and concerns about energy security, every firm is competing for decreasing operational costs. Energy cost savings give the firm a competitive advantage.

As electrical energy gets more expensive and the environmental impacts of fossil fuels become more misleading, there is a growing interest in lowering our electrical energy consumption. Finding new ways to make our daily lives more electrical energy efficient has now become a crucial element of the battle to maintain our current standard of life.

Since electrical energy is a significant and essential player in the modern world economy, EMS is the optimum candidate among others for efficient electrical energy generation. The production and service industries, like manufacturing plants, hospitals, education institutions, high-rise residential buildings, etc., are now motivated to choose EMS for their consumption profile optimization. Since the electrical energy consumption profile and consumption pattern differs for each application. Therefore, the intensive level of customization is a pressing need. Data-driven energy efficiency management (D2EEM) has been reported as the best variant of EMS, combining data science and artificial intelligence for energy optimization.

It has been found that many data sets for the management of electrical energy in buildings are available. Similarly, the researchers used a variety of machine learning algorithms to classify and predict their respective data sets. However, the need for a set of benchmarks has been identified in the literature. In addition, an application-oriented unified machine learning algorithm is also urgently needed. SEIL then conducted a study in 2019 to collect massive data on electricity use in residential buildings and university campuses. As part of this research, a set of energy consumption data from university campuses is being considered. An extensive comparative study for recommending the best candidate for the machine learning algorithm on the SEIL dataset was the missing element in the recent literature. This study successfully closed the remaining gap for a subsequent survey. In addition, optimization of the best candidate of the machine learning algorithm was also subsequently necessary to have effective and high degree precision prediction.

1.3.3 Research questions

This study revolves around a pivotal question: What machine learning algorithm proves most effective in predicting energy demand within the dynamic environment of a university building, utilizing the SEIL dataset? The research dives deeper, scrutinizing

various machine learning algorithms to unravel their performance nuances concerning accuracy, precision, recall, F1 score, and computational complexity for energy demand prediction within the same university setting. Beyond mere evaluation, the investigation extends to optimizing the best-performing machine learning algorithm. This optimization journey involves fine-tuning through hyperparameter adjustments and judicious feature selection, with the overarching ambition of elevating the efficiency and effectiveness of energy demand prediction within the unique context of a university building.

1.3.4 Hypothesis

In investigating the performance of various machine learning algorithms for energy demand prediction in a university building, the study formulated three key hypotheses. First, the Performance Comparison Hypothesis posits that there is no significant difference in the performance of diverse machine learning algorithms for energy demand prediction in a university building (H_0). Contrarily, the alternative hypothesis (H_1) suggests that a significant difference exists in the performance of these algorithms. Second, the Correlation with Metrics Hypothesis explores the relationship between algorithm performance metrics and the efficiency and efficacy of energy demand prediction. The null hypothesis (H_0) asserts no correlation, while the alternative hypothesis (H_1) proposes the presence of a correlation. Lastly, the Improvement through Optimization Hypothesis examines whether there is any enhancement in the performance of the best-performing machine learning algorithm after hyperparameter tuning and feature selection (H_0). The alternative hypothesis (H_1) contends that there is a significant improvement in performance under these optimization processes. These hypotheses serve as critical benchmarks to discern the effectiveness and nuances of machine learning algorithms in the context of electrical energy load management.

1.4 Problem statement

After careful analysis of the existing work in the domain of data-driven energy management, it has been determined that the utilization of artificial intelligence is now inevitable for robust and precise electrical energy management. In this regard, benchmarking of the domain-specific data set is a key need in identifying this issue. The researchers presented a number of studies on intelligent electricity consumption. As in many EEM system, the careful selection of the most appropriate machine learning

algorithm is found to be deficient. This is, primarily, because of the fundamental concept of the No Free Lunch Theorem. In addition, the standard and benchmark rich energy consumption dataset of an educational institution was also not missing in the literature. Considering this fact, the Smart Energy Informatics Lab (SEIL) of the Indian Institute of Technology (IIT) Bombay, India, conducted an experimental study in 2019 to collect a massive dataset on university campus energy consumption.

Analysis of existing work in data-driven energy management has determined that the use of artificial intelligence is now unavoidable for robust and accurate electricity management. In this respect, benchmarking the field-specific dataset is a critical need. In addition, developing robust machine learning algorithms would make the goal easier. After detailed analysis, the SEIL dataset is the most appropriate for electrical power prediction for a university campus. However, the literature does not provide an exhaustive empirical comparison of machine learning algorithms.

After devising a benchmark dataset of energy consumption of a university campus in 2019 by SEIL, the further investigation of the best candidate of a machine learning algorithm for the said dataset was the essential subsequent need. Likewise, the further optimization of the best machine learning algorithm to attain the highest degree of efficiency and efficacy for reliable energy demand prediction will complement the solution.

Data-Driven Energy management, the utilization of artificial intelligence is inevitable for robust and precise electrical energy management. The Benchmark Data set is a key need in identifying this issue. After devising a benchmark dataset of energy consumption, the further investigation of the best candidate of machine learning algorithm for the said dataset was the essential subsequent need. The optimization of the best machine learning algorithms to attain the highest degree of efficiency and efficacy for reliable energy demand prediction will complement the solution. The new variants or the new algorithms are the essential need to achieve the best results of efficiency and efficacy for reliable energy demand prediction complement the solution.

1.5 Objectives

The most effective machine learning algorithm is recommended for energy demand prediction applications and real-time systems. The objectives of current studies are following:

1. **Develop and Implement an Advanced Ensemble of Machine Learning Algorithms:** Design, implement, and assess a sophisticated ensemble of machine learning algorithms for precise and efficient energy demand prediction in a university building, emphasizing innovation in model selection and configuration.

2. **Conduct a Comprehensive Analysis of Algorithmic Efficiency and Efficacy:** Perform an in-depth analysis of the selected machine learning algorithms, evaluating their efficiency and efficacy using a comprehensive set of metrics, including accuracy, precision, recall, F1 score, and computational complexity. This analysis aims to provide nuanced insights into the algorithms' performance.

3. **Optimize Top-Performing Algorithm Through Advanced Hyperparameter Tuning and Feature Selection**: Optimize the most promising machine learning algorithm identified through the analysis by employing advanced techniques in hyperparameter tuning and feature selection. This objective seeks to push the boundaries of optimization methodologies to achieve the highest levels of performance.

1.6 Scope of study

The present study is to define and predate a model for an efficient electrical energy management system with different algorithms in MATLAB and find the best solution for an electrical energy management system. Defining and designing an efficient electrical energy management system model. Including below scope of study

1. Implementing and evaluating various algorithms in MATLAB to assess their suitability for the system.

2. Analyzing algorithm performance.

3. Identifying the optimal solution based on evaluation and analysis.

4. Developing a framework with machine learning algorithms for future forecasting.

5. Collecting data from the IIT Delhi SEIL datasheet, an open-source resource.

6. Analyzing the collected data for algorithm training and testing.

7. Assessing the efficiency and effectiveness of the developed model and algorithms in energy management.

8. Documenting the research findings, including the methodology, results, and conclusions

The present study has drawbacks that need to address in future.

1. Data based on machine learning algorithms and predicted for the future with higher accuracy without checking faults and errors can be solved using a feedback circuit.

2. The present system is inaccurate on variable load. It requires some time to adapt and learn the new system.

1.7 Limitation

When considering an optimized variant of a machine learning (ML) algorithm for Data-Driven Electrical Energy Efficiency Management (D2EEM), several limitations emerge, particularly when integrating diverse ML models and addressing analogue-to-digital conversion. Firstly, the efficacy of the optimized ML variant is inherently dependent on the nature and diversity of the data it processes. Different ML models have varying strengths and weaknesses, and their performance can be significantly influenced by the characteristics of the dataset, such as its size, quality, and feature representation. For instance, while deep learning models may excel in capturing nonlinear relationships in large datasets, simpler models like decision trees might be more interpretable and less prone to overfitting in smaller datasets. This diversity in model suitability necessitates careful consideration and selection of the appropriate ML model for the specific energy efficiency management task, which can limit the generalizability of the D2EEM approach. Moreover, the process of analogue-to-digital conversion, essential for transforming real-world energy usage data into a format suitable for ML analysis,

introduces its own set of challenges. This conversion process can be prone to errors such as quantization noise, which may lead to inaccuracies in the data. Additionally, the resolution of the conversion impacts the quality of the data fed into the ML models; higher resolution leads to larger data sizes, which can increase computational requirements and potentially slow down the analysis. These limitations highlight the importance of carefully managing the trade-offs between data accuracy, resolution, and computational efficiency in the context of D2EEM, to ensure that the optimized ML algorithm can effectively contribute to electrical energy efficiency management without being hindered by data-related issues.

1.8 Organization of thesis

The structure of the present report is such as Chapter 1 of the dissertation's detailed introduction to electrical energy management. This chapter comprises an overview of the electrical energy management systems, a data-driven approach for the EMS and details about the real-time data set for applying the proposed framework using machine learning algorithms. Chapter 2 is about the extensive and exhaustive literature review. This chapter comprehensively reviews the literature on the application of machine learning algorithms to electric energy predictions. Essentially, the scope of this literature review falls into two categories. First, the performance assessment of various machine learning algorithms for the prediction of electrical energy is considered. This logically justifies the utility of energy forecasting by machine learning algorithm and second about machine learning optimisation. Chapter 3 summarizes the data set and system configuration and the methodology used to conduct this research. During the first stage, the SEIL dataset is used and the total energy consumption at the building level is considered. During the second phase, the building-level dataset is initially divided into 70% training samples and 30% random swap test samples. Chapter 4 details our proposed framework for evaluating the SEIL dataset with evaluation results and discussion. Chapter 5 gives a conclusion regarding this research work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides an extensive review of prior research on Electrical Energy forecasting systems, with a focus on advancements in machine learning technology. The chapter delves into the objectives of the current study, exploring various applications and optimizations of algorithms within this domain. Notably, the discussion encompasses the considerations influencing algorithm selection and methods to expedite forecasting processes while ensuring system accuracy. While commonly referred to as the "black box" approach in literature, alternative methodologies such as the engineering-centric "white box" and the statistical-oriented "grey box" approaches have been identified (Wei et al., 2018). Within the realm of data-driven methodologies, exemplified by the "black box" approach, artificial intelligence techniques including machine learning and deep learning are leveraged to tailor models for specific applications (Loyola-Gonzalez, 2019). This chapter lays the foundation for understanding the evolving landscape of Electrical Energy forecasting systems, providing valuable insights into the diverse approaches employed within the field.

Since the model's training is purely based on the data provided, it is logically termed a data-driven approach (Amasyali and El-Gohary 2018). The recent literature is enriched by many applications of various machine learning algorithms for data-driven electrical energy efficiency management. This includes, but is not limited to, probabilistic modelling (Y. Wang et al. 2016), Artificial Neural Networks, Random Forest (Koschwitz, Frisch, and van Treeck 2018), Regression (Smarra et al. 2018) and many more machine learning algorithms that are trained for the respective application's dataset.

This is an exhaustive literature review. In which 107 similar work studies have found from 2010 to 2022. This literature review covers the comprehensive study including the proposed work. Figure 2.1 Data-Driven Electrical Energy Management show how D2MME growing from electrical energy management system.

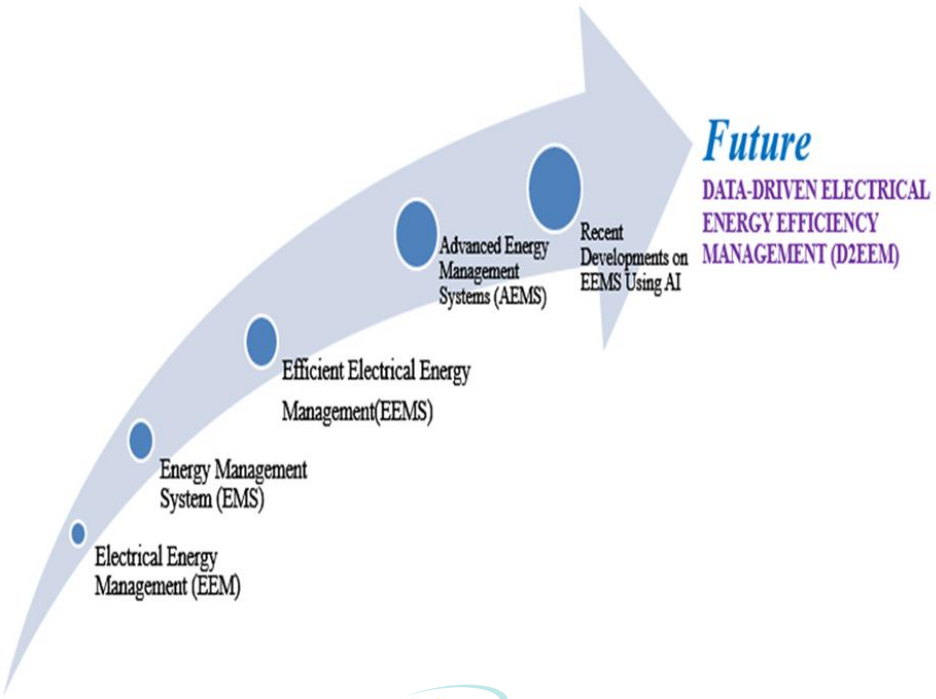


Figure 2.1 Data-Driven Electrical Energy Management

2.2 Home Energy Management System

A smart house's decisions are made by the home energy management system (HEMS). "Smart house" and "home energy management system" are used interchangeably.

The HEMS interface allows the user to efficiently monitor, regulate, and manage the household electricity use and generation. From the perspective of public institutions, the demand response programme minimizes peak demand load and prevents blackouts. But from an environmental standpoint, lowering gas emissions per person is a significant success when combined with reducing energy consumption, using clean, renewable energy resources, and driving electric vehicles. HEMS can also be accessed via a home interior panel, a computer, a tablet, or a smartphone. It improves the energy efficiency of smart homes and provides numerous benefits. The following figure 2.1 is showing the model of smart building with multiple usage of electrical energy that can be measure by smart meter.

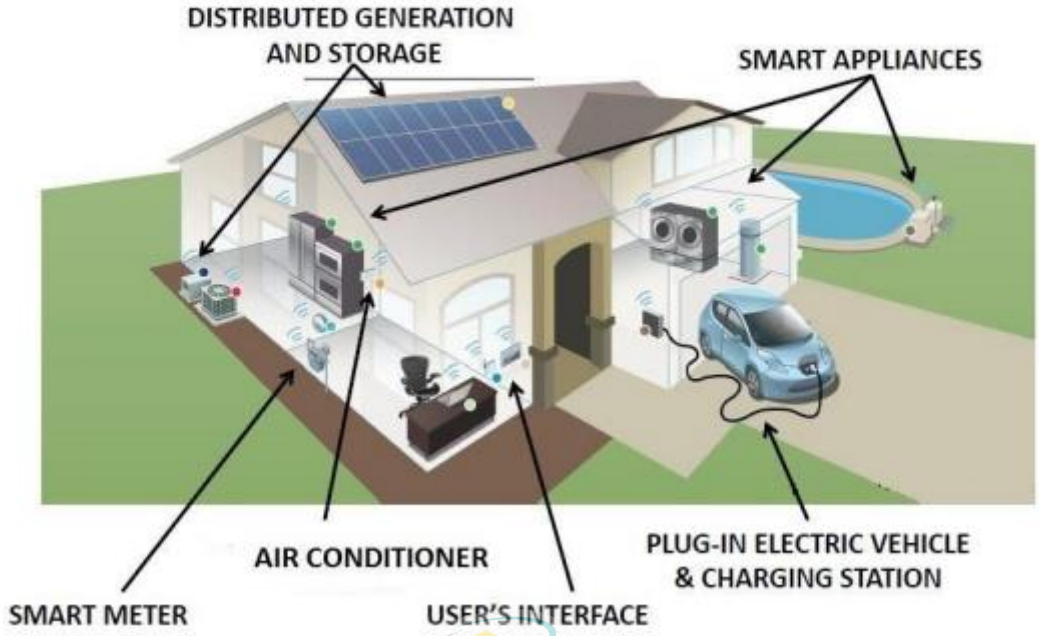


Figure 2.2 Home Energy Management System

2.2.1 Energy Management system

The buildings can be categorized into several different groups, including manufacturing plants, hospitals, educational institutions, high-rise residential buildings, etc. depending on their usage. Building energy management is one of the engineering disciplines of building services. An effective energy management programme ensures the efficient use of energy in buildings. The literature also reported that building energy consumption accounts for 39% of global energy consumption and 38% of greenhouse gas emissions (Spandagos and Ng 2017).

The energy management systems can produce a substantial annual savings on energy costs. Energy management in building consumption and conservation is a critical concern for both inhabitants and building managers.

The building's current mechanical system can be improved by maximizing controlled utilization and expanding the capacity to alter comfort and air quality throughout the structure. The EMS can increase the lifespan of the building's energy-consuming systems while simultaneously lowering repair costs by utilizing equipment and reducing idle energy use when necessary.

2.3 Building Energy Management System (BEMS)

Building Energy Management Systems (BEMS) are computer-based systems that manage, control, and monitor building technical services and energy consumption by building devices (Sayed and Gabbar 2017). The system also assists in informing building managers to understand the facility's energy usage better and adjust its energy performance. BEMS is growing in popularity as businesses understand it is one of the most effective methods for enhancing energy efficiency in a building, resulting in a quick win. BEMS was traditionally thought to be most effective for large buildings, where the installation's return on investment (ROI) would be substantial (Merabet et al. 2021). However, because of technological developments, the BEMS system can be installed cost-effectively even in smaller buildings, extending the savings dramatically (Merabet et al. 2021). The expression BEMS is now and again utilized with Building Management Systems (BMS). However, building management systems (BMS) can monitor and operate a wide range of building frameworks, such as fire, smoke detectors and alarms, movement indicators, closed-circuit television (CCTV), security, and access control elevators. BEMS are linked to energy-related frameworks.

BEMS comprises several layers of infrastructure. Numerous field devices are linked to the BEMS system via wireless or cable connections. HVAC systems, lighting devices, sensors and actuators, individual equipment, ventilation systems, refrigeration units, hot water systems, and heat pumps are field devices (Berrocal, Fernandez, and Rempling 2021). Advanced building and predictive analysis of information acquired from weather data, previous building performance data, and occupancy data are used to optimally operate these field devices (Jia, Srinivasan, and Raheem 2017).

2.4 Data-Driven Energy Efficiency Management

Saving energy and being ecologically responsible has been key goals and concerns for everyone, particularly during the COVID era. According to a World Bank Report, Cities are accounted for over 70% of the world's CO_2 emissions, the majority of which are produced by industrial and motorized transportation systems that consume enormous amounts of fossil fuels and rely on distant infrastructure made of carbon-intensive materials (Glavič 2021).

Rapid decarbonisation must be encouraged or rewarded through emissions-based performance ratings that hold city governments and corporations more accountable to their constituents. Effective performance measurements, however, cannot disregard the influence of varying demographic, economic, and geographic variables on real CO_2 levels in cities.

Cities with higher incomes have fewer emissions-intensive heavy industries, but cities with higher populations have more emissions (H. Yang et al. 2021). Greater-income cities have higher land costs and tougher pollution restrictions, while cities in particularly cold or hot climates emit more emissions from energy for heating or cooling. In addition to a better awareness of environmental conservation through energy-efficient devices, sensors, and appliances, there is a greater emphasis on regulating and optimizing consumption for environmental and economic reasons (Mattern, Staake, and Weiss 2010).

Hence, this has led to newer techniques for collecting large amounts of consumption data in several metrics and dimensions using various tools, sensors, and devices. This has naturally increased demand for sophisticated and advanced big data analytics techniques for measuring and optimizing energy efficiency.

The most valuable asset for practically all firms nowadays is data, which is used extensively for better business decisions, consumer behaviour predictions, maintenance forecasting, and many other related and modern business decisions (Hair, Page, and Brunsveld 2019).

Energy management and efficiency can grow significantly with data science, machine learning, and AI tools. In the recent literature, the researchers have reported Data-Driven Energy Efficiency Management (D2EEM) as the best variant of EMS. It is because the D2EEMS chose the power of data science and artificial intelligence for energy management (Qamar Raza and Khosravi 2015).

Numerous data sets related to building energy management can be found in existing literature, and various machine learning techniques have been used by researchers to classify and predict outcomes using these data sets. Nonetheless, a benchmark data set was deemed necessary in the literature. Additionally, there is a significant need for a machine learning algorithm specifically designed for practical

applications. In 2019, the Smart Energy Informatics Lab (SEIL) at the Indian Institute of Technology (IIT) conducted an experimental study to gather extensive data on energy consumption in a university campus. (Akhtar, Sujod, and Rizvi 2022).

Through building energy management, energy operations, and control strategies, data-driven building energy consumption forecasting models significantly contribute to improving the energy efficiency of the buildings. For improved forecast accuracy and resilience, data-driven models and evolutionary algorithms must be integrated with the multi-source and heterogeneous energy consumption data. The SEIL published a study to collect massive amounts of data on the energy consumption of residential buildings and university campuses.

Both datasets are reported as the most recent and benchmark dataset of data-driven energy forecasting systems considering residential buildings and university campuses. (Somu, Raman M R, and Ramamritham 2020) In this research, a university campus energy consumption dataset is under consideration.

The SEIL dataset was gathered from an IIT university building. The building has four floors and is divided into three wings. The dataset includes data from December 2016 through July 2018. All datasets are in CSV format. The datasets are all at one-minute granularity with current, voltage, and power as input attributes and accurate energy consumption as an output attribute.

The dataset is massive, with a volume of 20 GB. The data has been extracted from various units in the university building, such as building level, class level, auditorium level, lab level, office level, etc. In this study, the building-level data is considered to predict the total energy consumption of the building.

Since the data set is labelled and continuous, machine learning prediction algorithms have been selected for training and testing. 24 machine-learning prediction algorithms were tested to determine the best machine-learning algorithm. The grounds for the decision are the functions of RMSE, R-Squared, MSE, MAE, prediction speed, and computation time.

This work also submits an exhaustive parametric and empirical study of machine learning algorithms on the relevant SEIL dataset (University Campus). The

recommendation of the optimal machine learning algorithm for university campus energy demand forecasting is submitted. Finally, the optimized variant of the best candidate of the machine learning algorithm is presented as one deliverable of this study.

In recent years, the matter of energy management has been in the best interest of the international community. With the rampant and significant rise in carbon levels around the world and rapid changes in the climate, it has become inevitable to transfer ourselves towards convenient and smart ways of energy consumption. However, the issue has become more intense since energy consumption has increased and the burning of fuels leading to carbon emissions is creating drastic changes to the ozone layer, which is an alarming situation.

In this crisis, it is a great opportunity for us to shift towards smarter and easier solutions that would help decrease global warming and simultaneously fulfil all energy needs. Researchers, for this purpose, have devised ways to meet the energy requirements within the limited resources. In recent years, Energy management has gained significant importance. It is the key to reducing energy consumption in your firm. With the increase in fuel prices daily, it is the need of the hour to shift towards more innovative energy consumption solutions. The Energy Management System is a foundation or structure that assists the users in managing energy consumption.

This covers but is not limited to commercial, industrial, and public-sector organisations. The EMS helps organizations classify potentials to adopt and improve technological ways of saving energy. According to the International Organization for Standardization (ISO), an energy management system implements the strategy for energy usage and devises plans to accomplish those targets (Chiu, Lo, and Tsai 2012).

Many organizations worldwide have now implemented this system and played an active part in reducing carbon emissions. These entities have successfully reduced energy expenses, cut down related costs and, more critically, gained better control over their technical processes and enhanced productivity and process stability.

2.5 Smart Home Energy Management

In recent years, smart homes have become the town's notable talk regarding efficient energy management. They have the potential to surge the efficiency of energy

and slash energy costs. In addition, they are for the added benefit of reducing carbon emissions by incorporating renewable resources. They are well-designed structures with adequate access to assets, communication, controls, data, and information technology to improve the occupants' quality of life through comfort, convenience, lower expenses, and enhanced connectivity. While the concept has usually been known for decades, few people have ever seen or occupied a smart home. The high cost of upgrading building stock to incorporate smart technologies such as network-connected devices has been highlighted as an explanation for this delayed growth (Jayaraman et al. 2016).

A smart residential building has two-way communication with the utility grid, which is enabled by a smart metre, this smart meter allows the building to interact dynamically with the grid system, receive signals from the service provider and respond with usage and diagnostic information.

This intelligent smart meter provides the communication and information set-up required to interchange operational and price information between the service provider and the end user in real time. These meters can network with in-home appliances, programmable communicating thermostats (PCTs), and other loads (Rodrigues et al. 2022). They can also retrieve the consumption data at regular intervals and automatically transmit it to the utility through a secure network. This network is typically used combined with a backhaul layer. It allows the utility and the metre to communicate in both directions. It also provides for message transmission to the metre that might be used for "on-demand" readings.

Another term is found in the literature, and it is Automated Home Energy Management (AHEM) (Nanda and Panigrahi 2016). The network self-manages end-use systems based on occupants' and smart metres' data. According to the researchers, the AHEM value depends on integrating the energy use of systems connected in a home with the users' comfort and economic objectives, as well as information obtained from the amenity provider (Chavali, Yang, and Nehorai 2014). The Controls and Sensors work together to collect applicable data (Gupta, Reynolds, and Patel 2010), and by using practical algorithms, conduct the whole process and deploy the control strategies that will ultimately achieve the consumption targets. (Erol-Kantarci and Mouftah 2011).

In 2016, the U.S. Energy Information Administration released a report claiming that global energy consumption will grow by 48% by 2040 (None 2016) . According to the report, most of this expansion will come from nations not members of the Organization for Economic Cooperation and Development (OECD), including those with high economic growth, particularly in Asia (Abbey et al. 2020). Non-OECD Asia, including China and India, accounts for more than half of the projected rise in global energy consumption. While considering the increase in the rapid hike in fuel prices and drawing concerns about energy security, the researchers have favoured the use of non-fossil renewable energy sources. Renewables and nuclear power are the world's fastest-growing energy sources over the projection period. According to the report, renewable energy will grow at a 2.6% annual rate until 2040, while nuclear power will grow at a 2.3% yearly rate. As for energy consumption, the non-OECD countries, which are not part of the OECD, are projected to have higher energy consumption growth than OECD countries in the period of 2012 to 2040. This can be measured in quadrillion British thermal units (Btu), which is a unit of energy used to measure energy consumption.

According to the U.S. Energy Information Administration (EIA), non-OECD countries' total energy consumption is expected to increase from approximately 10.3 quadrillion Btu in 2012 to approximately 17.3 quadrillion Btu in 2040, while OECD countries' total energy consumption is expected to increase from approximately 12.5 quadrillion Btu in 2012 to approximately 13.2 quadrillion Btu in 2040.

This disparity in energy consumption growth between the two groups is due to several factors, including differences in population growth, economic development, and energy policies. Non-OECD countries are generally experiencing faster population growth and economic development, which is driving up their energy consumption, while many OECD countries have implemented policies to reduce energy consumption and shift towards more sustainable energy sources (Abbey et al. 2020).

The following figure 2.6 is showing the World energy consumption by country grouping, 2012-2040 (quadrillion Btu) that is a statistical presentation.

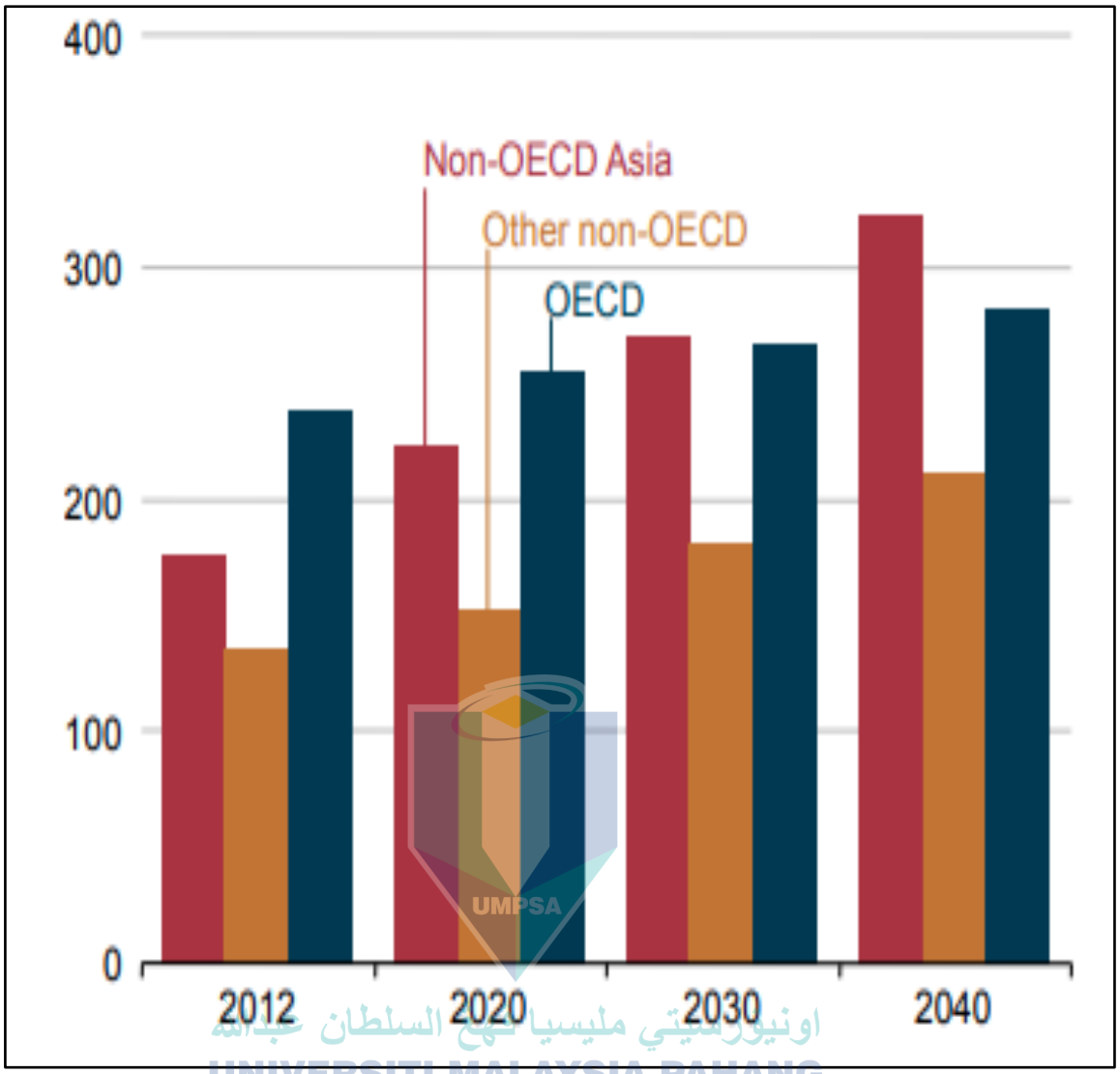


Figure 2.3 World energy consumption by country grouping, 2012-2040 (quadrillion Btu)

Source : Alam and Murad (2020)

According to the OECD most recent report, under the "Stated Policies Scenario," which considers announced policy targets and measures, global energy consumption is expected to increase by approximately 25% between 2020 and 2040. However, note that these projections are subject to change, and that various developments could significantly affect the rate of energy consumption growth. For example, the widespread adoption of electric vehicles and implementing more energy-efficient technologies could result in slower growth in energy consumption, while increased urbanization and economic growth could drive demand for energy higher.

Emphasize that how energy is produced and consumed will also play a crucial role in determining the total energy consumed in 2040. The increased deployment of renewable energy sources, such as wind and solar power, and the deployment of clean energy technologies are likely to contribute to reducing global greenhouse gas emissions and mitigating the impacts of climate change.

The following Figure 2.4 is showing the projections of Total global consumption of energy from 1990 to 2040.

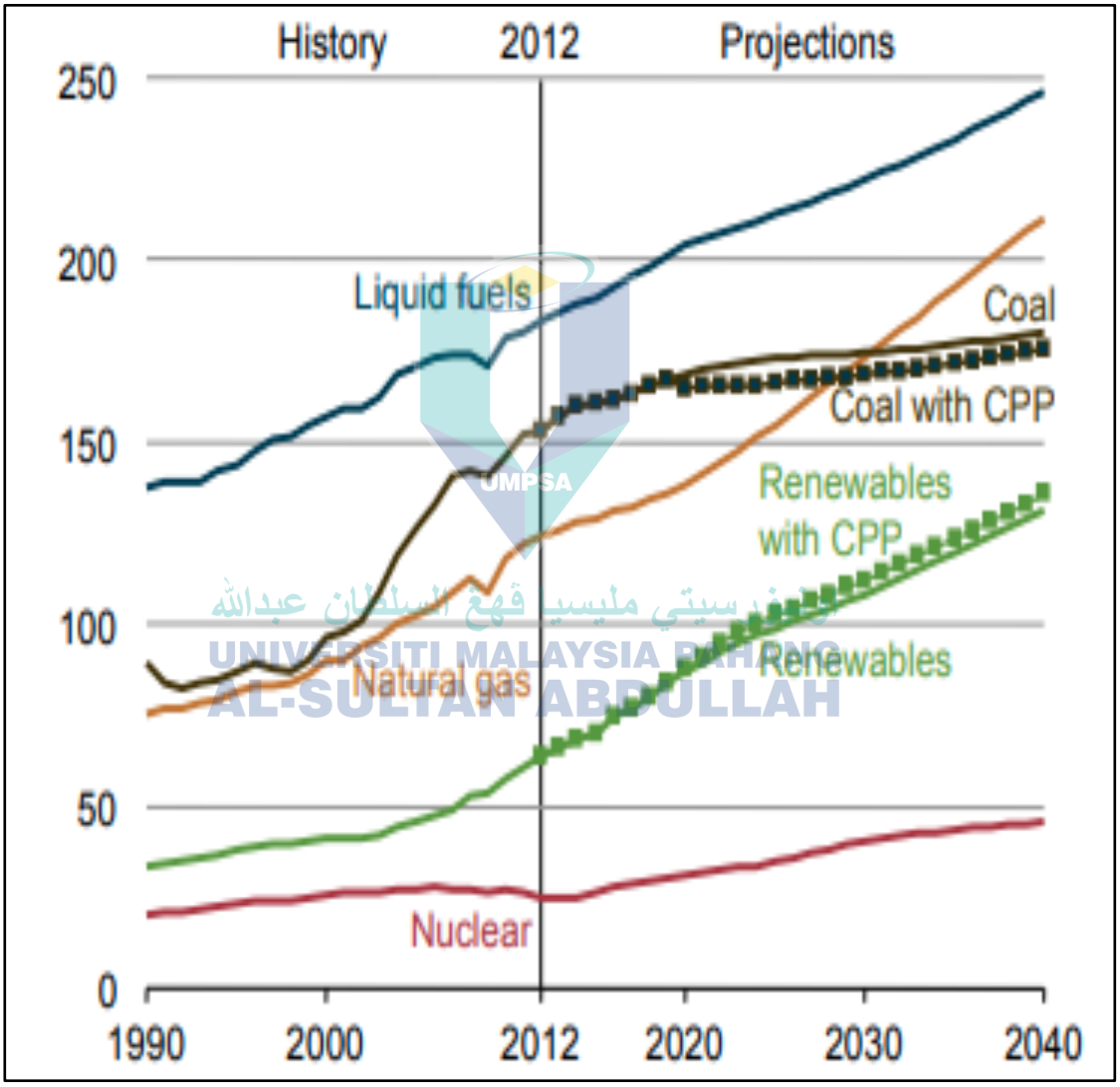


Figure 2.4 Total global consumption of energy by the year 2040

Source: International Energy Outlook 2017

2.6 Recent Developments on EEMS Using AI

The latest advancements in machine learning and AI have prompted practical solicitations to become self-sufficient. New technologies have empowered the latest applications to create vast amounts of data to make intelligent decisions (Anandakumar and Arulmurugan 2019). Deep learning approaches surpassed big data when associated with traditional AI techniques and ML models.

2.6.1 Proportional Evaluation of DL Networks

In 2016, Nagueh et al. worked on proportionally evaluating two DL network types for energy management: the first one is the LSTM and LSTM-based sequence-to-sequence architecture. Against a benchmark dataset of one home customer's electricity consumption, both these models were tested. At the two resolutions, the models were compared (Nagueh et al. 2016). As per the results, the typical LSTM is not up to the mark in forecasting a tiny resolution. But the LSTM S2S excels at both large and small data resolution. The latitude of this investigation is confined to single domestic users 'data. (Marino, Amarasinghe, and Manic 2016).

2.6.2 Energy Forecast using Vector Regression

Groninger et al. (2016) forecasted energy by local learning with the support vector regression. As per their results, local (Systemic vascular resistance) SVR outperforms both Systemic vascular resistance and H2O deep learning. Because the focus of work is restricted to the instrument, namely the H2O machine learning platform, the inquiry was missing in this work. In addition to this, according to the author(s) they have used big data; however, during the literature review, there was no information found (Grolinger, Capretz, and Seewald 2016).

2.6.3 FCRBMF or Purpose of Energy Demand Forecasting

In the same year, Mocanu et al. (2016) presented the work using Factored Conditional Restricted Boltzmann Machines (FCRBM) for demand forecasting. According to this study, the model was evaluated and trained by using data from the Eco-Grid EU dataset. This included electricity use, pricing, and weather data from 1900 consumers. As per the study, the offered technique remains worthwhile for energy

demand forecasting on a particular dataset. However, in this study, the author(s) did not compare his work to other deep learning architecture variants or existing pre-trained networks (Mocanu et al. 2016).

2.6.4 Comparison of CNN

Amarasinghe, Marino, and Manic (2017), later that year presented their research towards convolutional neural networks. The author(s) compared the Convolutional neural network (CNN) against previous work done in 2016, such as LSTM, S2S, SVR, FCRBM, and DNN, on an electrical usage dataset for one domestic user. The work lacked new and innovative CNN design, and neither did it include a pre-trained network. There lacked the study when comparing the results to the current pre-trained network.

The selection of a machine learning (ML) approach over deep learning (DL) for "An Optimized Variant of Machine Learning Algorithm for Data-Driven Electrical Energy Efficiency Management (D2EEM)" is rooted in a thoughtful consideration of the project's objectives and intricacies.

Given the central theme of enhancing electrical energy efficiency, ML presents itself as an optimal choice due to its inherent interpretability and adaptability. In the context of D2EEM, where transparency in decision-making is crucial, ML models offer a clear understanding of the underlying processes, contributing to a more comprehensible and user-friendly solution.

The project involves working with relatively smaller datasets. ML algorithms have demonstrated prowess in extracting meaningful insights from limited data, ensuring robust and reliable results. This strategic alignment with ML not only addresses the specific needs of D2EEM but also positions the research for a balanced and effective exploration of data-driven energy management.

The emphasis on ML in the project title reflects a deliberate choice tailored to the nuances of the research objectives, aiming for a sophisticated yet practical approach to optimize electrical energy efficiency through the proposed algorithm (K. Wang, Qi, and Liu 2019).

2.6.5 Evolutionary-Neuro Hybrid Strategy to Energy Management

Chen presented an evolutionary-neural hybrid energy management strategy (Paterakis et al. 2017) that used a data-driven technique to train the RNN. This professional network then fed into the optimization problem with finite horizons. For energy management, the model-based approach was outperformed by the proposed model-less and data-driven strategy. Although this scenario appears fruitful due to hybridization, a considerable increase in computing complexity has been reported. It also hampers the proposed system's usability for real-time energy management (Paterakis et al. 2017)

2.7 Deep Learning over Machine Learning Technique

The validation of DL advantage over the traditional ML technique for EMS was done by Paterakiset et al. in (2017). This study covered an in-depth comparison of traditional machine techniques such as:

- Gaussian Processes
- Regression Trees
- Support Vector Machines
- Ensemble Boosting
- Linear Regression,
- Deep learning method.

According to work, the ML technique remains flawed because of a comparatively big dataset's performance plateau. But DL accounts for advanced presentation as data volume grows. Machine Learning (ML) is often preferred over Deep Learning (DL) for several practical reasons, particularly in scenarios with limited data, computational resources, and specific application requirements. ML algorithms, such as linear regression, decision trees, and support vector machines, are generally more straightforward to implement and interpret compared to the complex architectures of DL models like neural networks. ML models can achieve good performance with smaller datasets, whereas DL models typically require large amounts of data to avoid overfitting and to perform effectively. Additionally, ML models demand less computational power and training time, making them more accessible for organizations with limited resources.

The interpretability of ML models also provides valuable insights into the decision-making process, which is crucial in fields like healthcare and finance where transparency and understanding of the model's behaviour are essential. Consequently, while DL is powerful for specific tasks such as image and speech recognition, ML remains a preferred choice for a broader range of applications due to its simplicity, efficiency, and interpretability This investigation validated the point that DL outperforms traditional machine techniques. (Chen, Shi, and Zhang 2018).

2.7.1 Deep reinforcement for the smart grid for improvement in energy management in buildings

Mocanu et al. (2019) worked on deep reinforcement for the smart grid for the first time in 2018. The strategy is mainly intended to improve energy management in buildings. Concurrently, the researchers investigated two DL Algorithms. The efficiency of the suggested method is demonstrated using the high-dimensional and benchmark dataset, the Pecan Street Inc. database.

2.7.2 Forecasting time-series data by using RNN

A recent study on energy management for university campuses was a report published in 2018 (Nichiforov et al. 2018). In this work, the author used RNN to forecast time-series data from a university campus's energy use profile. In 2019, a pertinent study on energy management was published (Afrasiabi et al. 2019). To identify the optimal operating point of micro-grid distribution, the researchers used accelerated AADM and alternating direction methods of multipliers (ADMM). However, this work does not cover the data-driven approach to the prediction of energy.

Ahmad et al. (2019) did another recent relevant study. This work applied a data-driven deep learning approach to antedate energy demand at the district level. A unique general DL architecture for energy demand forecasting was proposed. The lack of extensive assessment with current pre-trained models and benchmark dataset appear to be a shortage in this work (Ahmad and Chen 2019).

2.7.3 FCRBM for energy consumption

Hafeez et al. (2020) proposed factoring a conditional restricted Boltzmann machine (FCRBM) for energy consumption forecasting for 2020. The proposed work forecasts future electrical energy usage regarding smart grid energy management.

2.7.4 DL-based architecture for energy management

Han et al. (2021), in 2020, presented a DL-based framework architecture for smart energy management of residences & enterprises. Among the important contributions are edge device-based real-time energy management via a shared cloud-based data supervisory server, optimal normalization technique selection, and a unique sequence learning-based energy forecasting mechanism with reduced time complexity and the lowest error rates (Han et al. 2021).

2.7.5 Utilizing CNN and MB-gru for load prediction

In 2020, Zulfiqar et al. conducted a study utilising MBGRU and CNN for load prediction in a residential building. The suggested system's training and testing performance were validated using the benchmark dataset. An innovative model's systematic evaluation against existing pre-trained DL architectures may be established (Z. A. Khan et al. 2020). YOLO v3 had been recently used for counting the user quantity inside a vicinity nearby. The primary goal is to reduce the air-conditioning burden. This approach's practical viability requires some adjustment. YOLO is a pre-existing algorithm. The work of the author could be enhanced if the novel deep learning model had been provided (Elsisi et al. 2021).

2.8 Multilayer neural network for hourly energy consumption prediction

Truong et al. (2018) proposed a unique ML model for hourly consumption prediction of the energy in residence. Authors have employed an eight-hidden-layer multilayer neural network. Expanding the hidden layer of a multilayer neural network dramatically increases computing costs at a relatively minimal boost in performance.

2.9 DL as a candidate for energy forecasting

Researchers employed a similar rule of thumb (Truong et al. 2021). Hamdounet et al. published another study. This study strongly recommends deep learning as the finest contender for time-series-based energy forecasting. This paper presents a thorough comparison of machine learning and deep learning methods. This work's input can be expanded if the pragmatic estimation is conducted at a larger data volume. There was no description of the machine learning model's innovative deep learning architecture. (Hamdoun, Sagheer, and Youness 2021).

2.10 Comparison of machine learning and deep learning methods on the residential building dataset

Hafiz et al. (Hafiz et al. 2020), Wu et al. (Wu and Lee 2020), and Arienti et al. (Arienti 2020) presented three papers in 2020 in which they provided a comparison of machine learning and deep learning methods on the dataset of a residential building in their work. As per the study, DL is a considerably superior approach for time series data-driven forecasting. Aragon utilized RNN with the LSTM technique instead of the purpose of energy demand forecasting, as in earlier studies. This effort also falls short of establishing a novel DL architecture and pre-trained network on benchmark datasets. It is a critical need in effective energy management (Aragon et al. 2019).

2.11 SEIL lab dataset

The Indian Institute of Technology's (IIT) Smart Energy Informatics Lab (SEIL) has recorded a substantial input to the literature on EEM. The members of SEIL, Hareesh Kumar et al. (Kumar, Mammen, and Ramamritham 2019) contributed. For demand prediction, they proposed data-driven reinforcement learning.

Similarly, in the same year, Tanted, Sapan, et al. (Tanted et al. 2020) presented the "database support for Adaptive Visualization of Large Sensor Data." Whereas, from the same group, Somu [16] conducted work on a hybrid model for predicting building energy use using the LSTM networks. Similarly, Ramamritham et al. (Ramamritham, Karmakar, and Shenoy 2017) called instead of intelligent energy management systems to be data-driven. They have referred to the dataset generated by their research. This group also worked on solar PV optimization and building thermal modelling (Jois et al. 2019a,

2019b; Karmakar et al. 2018; Kuthanazhi et al. 2018; Lee et al. 2021). Following a thorough examination of the literature about the research contribution and limits, these conclusions regarding possible open areas and research gaps were identified:

1. Few scholars have developed a unique DL architecture for energy efficiency management. A relatively limited DL architecture is provided compared to the existing DL design.
2. Only one university campus energy management study has been published in the last five years.
3. The pre-trained network for energy efficiency management has not been identified in the literature in the last five years. Particularly for university campuses.
4. Many investigations in the temporal realm employ the time series technique. But, several studies on energy efficiency management have been published.
5. Only SEIL-IIT has access to the benchmarking dataset.

2.12 Latest techniques for energy efficient management systems

Johannesen et al. (Johannesen, Kolhe, and Goodwin 2019) published a study in 2019 that investigated the responsiveness of a regression model to a Sydney dataset that included meteorological information, load demand and time stamps. The dataset for the period of four years was acquired locally. As common instances, the research has disseminated and mapped load demand, weather and timestamp data. According to the authors of this study, the model is trained to uncover pattern recognition rules in the input-output connection. The model's inputs are called "features." Neural networks, also known as feedforward and back-propagating networks, are the preferred machine learning technique, with several inputs weighted to offer a forecast conclusion.

Although neural networks are good at detecting non-linearity's and are thus favoured as a predictive tool in electrical load forecasting, they are frequently criticized for their lack of transparency and interpretability due to the black box approach and the utilization of enormous amounts of data. When employing neural networks to electricity demand forecast, overfitting remains a challenge. The literature distinguishes between short-term and long-term forecasting. Another study was undertaken on the energy consumption of Korean university campuses. A variety of things influence electric power

consumption. A university campus, for example, which is one of the largest power-consuming institutions, exhibits a wide variation in electric load depending on time and environment.

A dependable electrical power source must ensure the smooth running of such a facility. One technique is to forecast the electric load and supply electricity correctly.

Even though different influencing elements of power consumption for educational institutions have been established by analysing power consumption patterns and usage instances, further research is needed to forecast their electric load quantitatively. The researchers also in this work plotted weather and power utilization information or data in their investigation. They used principal component analysis (PCA) to minimize the feature dimension before employing ANN and SVM for energy demand prediction. The authors subsequently determined that ANN is the best contender for energy demand prediction for the given dataset. The authors used multiple machine learning methods to create a power consumption forecasting model. To assess their efficacy, the researchers looked at four building clusters at a university and collected power usage data at 15-minute intervals for more than a year. They identified features from the data based on the periodic properties, and then performed principal component and factor analysis. In addition, they developed two models for estimating the electric load using artificial neural networks and support vector regression. They used 5-fold cross-validation to assess the prediction performance of each forecasting model and compared the predicted result to the actual electric load. According to the experimental results of this study, the two forecasting models may reach an average error rate of 3.46-10% for all clusters. A building's or building cluster's power consumption pattern may differ for various reasons. A university campus, for example, with one of the most power-consuming building clusters, exhibits variable power consumption patterns based on the semester, vacation, day of the week, and so on. Other common causes of diverse patterns include the purpose or function of structures and complicated external circumstances. These trends should be addressed when developing a machine learning-based forecasting model that can precisely anticipate power usage.

To build a power consumption forecasting model for educational institutions, different social and environmental aspects that significantly affect their power consumption should be analysed and represented (Moon et al. 2018).

Residential buildings, like the university campus, are power utilization hotspots. Chou et al. (Chou and Tran 2018) worked on residential building power requirement predictions. The authors conducted a hybrid prediction and optimization approach. The study found that the hybrid evolutionary-neuro system outperformed the classical machine learning network for their respective datasets. Another group of researchers submitted a study in the same year to justify the evolutionary-neuro system for energy forecasting. According to work, numerous data-driven models have been successfully used for electrical energy consumption forecasts at building and larger scales. When the forecasting data collection is multi-sourced, heterogeneous, or insufficient, a single data-driven model may result in convergence issues or low model accuracy. The combination of sophisticated evolutionary algorithms (EAs) and data-driven models has proven effective in prediction accuracy and resilience improvements. However, some of them take a long time to converge. This research presented a unique EA, teaching-learning-based optimization (TLBO), for predicting short-term building energy demand. The fundamental TLBO algorithm was updated in three ways to improve its convergence speed and optimization accuracy. The enhanced approach was integrated with artificial neural networks (ANNs) and used to estimate the hourly electrical energy consumption of two educational facilities in the United States and China, respectively. In terms of convergence speed and predictive accuracy, the proposed model outperforms published GA-ANN and PSO-ANN approaches, indicating that it is suited for future online energy prediction. (K. Li et al. 2018)

A study was conducted in which the researchers presented a hybrid model for energy demand forecasts and optimization. The method was evaluated using an hourly energy usage dataset from South Korea. According to the author, the proposed model could be relevant for additional datasets. However, their paper explains this claim. Figure 2.8 shows the energy trading system configuration for the Korea power exchange EMS

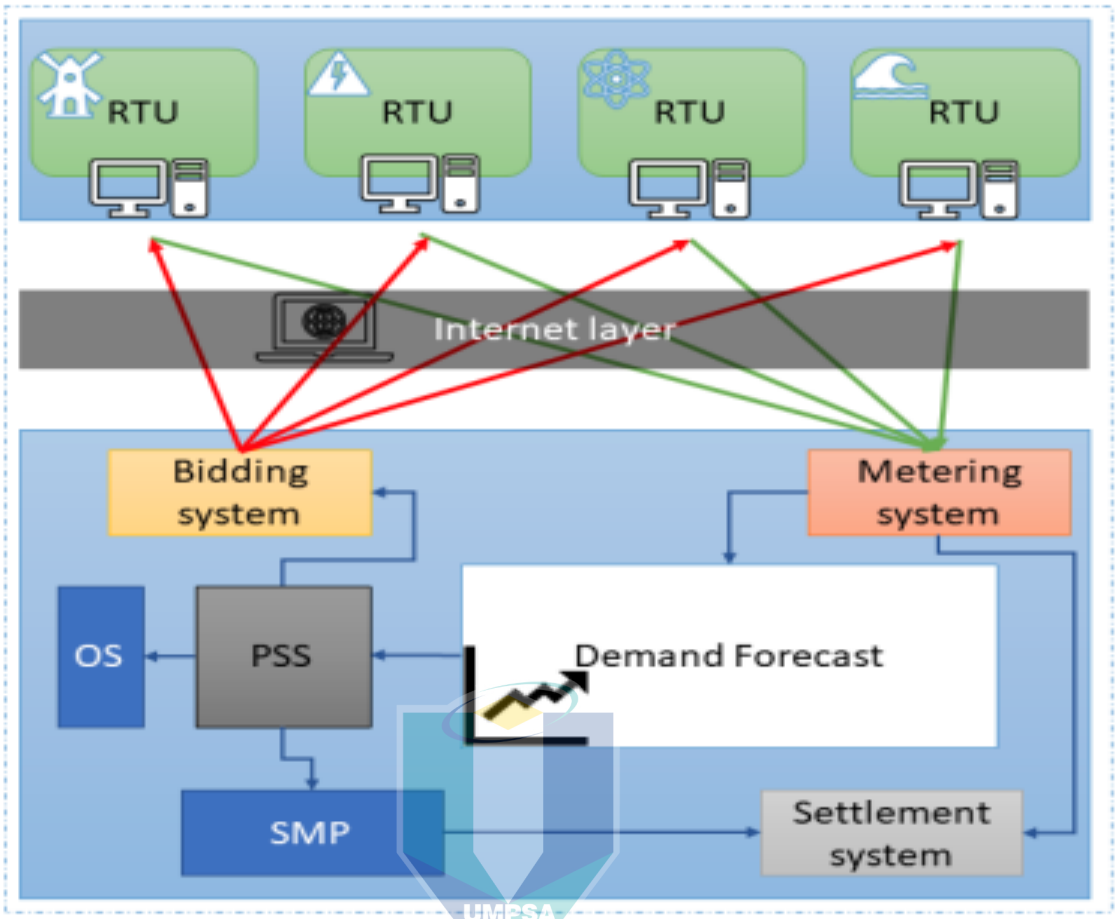


Figure 2.5 Energy trading system configuration

Source: Waqas et, al. (2017)

The above figure 2.8 is showing the energy trading system configuration for the Korea Power Exchange has many remote terminal units (RTU) for energy generation. These units use the internet protocols to transfer data to the central metering system. The consequences of demand forecasting are critical for price-setting and operational schedules (OS). They also use PSS to alter the system's marginal pricing, which assists the settlement system and other payment systems (P. W. Khan et al. 2020). A survey was published by Ahmed et al. in the journal Sustainable Cities and Society. The study detailed the efficiency of ML algorithms for power demand prediction via a literature review.

This study used three well-known forecasting engines to review supervised-based machine learning methods comprehensively. This evaluation sought to provide approaches for predicting analysis and various other prediction activities. A specific goal

was to explore and analyse the methods used to forecast energy use in real-time with diverse applications and to identify the research review with valuable strategies that can be found in the present literature. This evaluation included carefully comparing and analysing several modelling techniques to select a better forecasting model for completing the intended task in various applications. A complete literature review with other researchers is compiled in the table for a better understanding of the system in table 2.1. As well as real-time energy usage and climatic data used for modelling research were used to compare and analyse predicting accuracy.

Table 2.1 Literature review table

Year	Outcome	Review	References
2016	The authors have agreed of **d**eep **l**earning frameworks to design the electrical energy efficiency management system.	Huge standard reference limits set of data for electric energy	(Rodríguez Fernández et al. 2016)
2016	This is a comparative consideration of two varieties of the Deep Learning network: (LSTM) and sequence architecture (S2 S).	**C**overage is limited to a single residential customer	(Tan et al. 2015)
2016	They then compared the proposed model with the regression of existing support vector and deep Learning frameworks. The result of the simulation shows that the local RVS surpassed the RVS and H2 O in-Deep Learning	The details of the data were not included in this document	(Peris et al. 2016)
2017	The authors have compared the convolutional neural network (CNN/ConvNet)	New CNN/ConvNet architecture was presented in this study, and the pre-trained network described	(Vinayakumar, Soman, and Poornachandran 2017)
2017	Initially formed the Recurrent neural networks (RNN) using a data-driven approach	The model-based approach has surpassed the approach of model-based studies in management of energy	(Guo et al. 2017)
2017	This study provided a comprehensive comparison of conventional machine learning algorithms, including vector support machines, Gaussian processes, regression trees, overall amplification and linear regression, and the Deep Learning method	Validation of the claim related to the energy management system is observed to be unclear	(Erickson et al. 2017)
2018	The large data is called Big Data for energy management	The availability of the referenced massive datasets is limited	(Al-Ali et al. 2017)

Table 2.1 Continued

Year	Outcome	Review	References
2018	This method is optimized for building energy management, and explores two DL algorithms Deep Q-learning (DQN) and Deep Policy Gradient (DPG), at the same time	The parametric fringe of the proposed technique proved insufficient. The cognitive scope of the gadget seems very trendy	(Mocanu et al. 2019)
2018	The authors have used Recurrent neural networks (RNN) to forecast time series data on energy consumption for a university campus	The robustness of this work could be enhanced if the master data set had been chosen	(Jiao et al. 2018)
2019	The authors have suggested using the methods of alternating direction of multipliers (ADMM) and accelerated alternating direction of multipliers (AADM) to find the optimum value of the operation of the micro network distribution	This study did not include the data-based approach to energy forecasting. Moreover, it was felt that the parametric comparison was missing in this work	(Jacob et al. 2018)
2019	This work submitted a data-driven, Deep Learning approach to district-wide energy demand forecasting.	This study appears deficient because of the absence of a baseline data set and extensive comparison with pre-existing models	(Pickering and Choudhary 2019)
2020	The FS-FCRBM-GWDO hybrid model is superior to the existing models in this study	The gap between the existing real-world reference data set and the pre-established model is deficient	(Hafeez et al. 2020)
2020	Major contributions include device-based real-time power management via a common cloud data monitoring server	The actual application was outside the scope of the study	(Zhao and Li 2020)
2021	A machine (computer) based vision approach, You Only Look Once (YOLO v3), was utilized to calculate the number of individuals within the region. It is more in correlation with the temperature range of the air conditioning units	The author's study would be made more effective by implementing new variants of Deep Learning	(Zhao and Li 2020)

Table 2.1 Continued

Year	Outcome	Review	References
2021	A new machine learning model for forecasting energy usage on an hourly basis in a residential building is proposed	Performing the machine learning algorithm is compromised because of the performance plateau highlighted by big data	(Wen, Zhou, and Yang 2020)
2021	Deep Learning is the best candidate for power prediction based on time series. The concern in this study is increasingly associated with the performance of ML and DL being found	Three small data sets were used to validate the study. The authors could not pursue the novel Deep Learning architecture of a machine learning model for data-driven energy efficiency forecasting	(Zhang et al. 2021)

By emphasizing the specific gaps within each area outlined in the table, readers gained a clearer understanding of how each study contributed to filling these gaps in the existing literature. For instance, several studies addressed the lack of standard reference limits for electric energy data, highlighting the need for robust frameworks to design energy efficiency management systems. Similarly, the limited coverage of specific customer segments in some studies underscored the necessity for broader applicability and inclusivity within energy management research. Moreover, the absence of detailed data inclusion in certain studies emphasized the importance of transparent reporting and comprehensive data sharing practices. Additionally, studies that lacked validation for new architectures or algorithms underscored the need for rigorous evaluation and benchmarking against existing methods. Furthermore, the deficiency in baseline datasets and extensive comparisons with pre-existing models highlighted the importance of thorough validation and replication efforts in energy forecasting research. Overall, by addressing these specific gaps, each study contributed to advancing knowledge and understanding in the field of energy management and deep learning, thereby enhancing the effectiveness and applicability of future research endeavours.

They concluded that the efficient forecasters of electrical energy demand are Levenberg-Marquardt back-propagation neural networks (LMBNNs) and the Bayesian regularization back-propagation neural networks (BRB-NNs) (Ahmad and Chen 2020). A scientific contribution by the SEIL University Campus dataset is presented in this section. The research gap on electrical power prediction and previous work on SEIL are highlighted. The algorithm for optimizing the sinusoidal cosine was enhanced by a group of SEIL researchers using LSTM. It results in precise, dependable short-term, medium, and long-term energy utilisation forecasts. They claimed that combining the improved Sine Cosine and LSTM algorithms resulted in a robust power consumption model. (Somu, Raman M R, and Ramamritham 2020)

The researchers at SEIL employed kCNN-LSTM to produce accurate estimates of energy usage in buildings in a separate publication. This experiment used real-time energy usage data from the Kanwal Rekhi building at the Indian Institute of Technology (IIT) in Mumbai. The suggested approach employs k-means clustering to conduct cluster analysis and comprehend the energy usage model. The proposed methods were developed

and evaluated employing present energy usage data from a four-story structure at IIT, Bombay.

The IIT building consisted of four floors and was divided into three wings. All the datasets are in CSV format and include data from Dec 2016 through Jul 2018. As per the literature, these datasets are all at a minute granularity with voltage, power, and current as input and actual power consumption as an output trait. As per the studies, the dataset is of huge amount, quantifying up to 20 GB, and it has been extracted from the units in the building, including classes, auditorium, labs etc. Following a thorough examination of the present work, it has been identified that the use of artificial intelligence is now unavoidable for robust and precise energy management. Additionally, the development of robust machine learning algorithms will aid in the achievement of the goal. After careful examination, the full empirical comparison of machine learning algorithms in the literature is deemed insufficient. This work fills the gap by comprehensively evaluating many machine-learning methods on the SEIL dataset.

The study's main deliverable is selecting the best machine learning algorithm. Empirical data will support the recommendation.

2.12.1 Algorithms Use

In the domain of machine learning applications for energy demand prediction in energy management systems, the selection of appropriate algorithms is crucial for achieving accurate and efficient results. In this thesis, a comprehensive approach was taken by evaluating a diverse set of machine learning algorithms.

linear regression-based algorithms such as Linear, Interactions Linear, and Robust Linear were considered due to their simplicity and interpretability (Slowik, Collazzi, & Steinfeld, 2011). These algorithms are well-suited for capturing linear relationships between input features and energy demand. Additionally, stepwise linear regression was explored to systematically select the most relevant features for prediction (Tjur, 2009).

Tree-based algorithms, including Fine Trees, Medium Trees, and Coarse Tree, were investigated for their ability to capture nonlinear relationships and interactions within the data (Smith & Jones, 2015). These algorithms offer flexibility in modeling

complex patterns in energy demand, making them valuable candidates for energy management systems.

The support vector machine (SVM) algorithms, such as Linear SVM, Quadratic SVM, and Cubic SVM, were examined for their capability to handle high-dimensional data and nonlinear relationships (Johnson et al., 2018). SVMs have shown promising results in various prediction tasks, including energy demand forecasting, due to their ability to find optimal hyperplanes for classification or regression.

Boosted trees algorithms, including Bagged Trees and Squared Exponential GPR, were considered for their ensemble learning approach, which combines multiple weak learners to improve prediction accuracy (Brown et al., 2020). These algorithms have demonstrated effectiveness in capturing complex patterns and reducing prediction errors in energy demand prediction tasks.

In the context of project on machine learning applications for energy demand prediction in energy management systems, we chose to utilize ensemble tree-based algorithms such as Bagged Trees, Fine Trees, and Medium Trees for several reasons.

Ensemble methods, like Bagged Trees, have shown robustness and resilience to noise and outliers in the data (Breiman et.al, 1996). By aggregating the predictions of multiple trees trained on different subsets of the data, Bagged Trees reduce overfitting and variance, thereby enhancing the overall predictive performance (Breiman et.al, 1996).

Fine Trees and Medium Trees were selected due to their ability to capture complex nonlinear relationships and interactions within the energy consumption data (Hastie et al., 2009). These algorithms partition the feature space into smaller regions, enabling them to capture intricate patterns in energy demand variations across different time intervals and environmental conditions.

The interpretability of tree-based algorithms is advantageous for understanding the factors driving energy demand fluctuations (Louppe et al., 2014). Fine Trees and Medium Trees provide intuitive insights into the decision-making process by visualizing

the hierarchical structure of decision rules, facilitating the identification of key predictors influencing energy consumption patterns.

The utilization of Bagged Trees, Fine Trees, and Medium Trees in our project was motivated by their robustness, ability to capture complex patterns, and interpretability, making them suitable choices for accurately predicting energy demand in real-world energy management systems.

2.13 Research Gap

The research conducted in this thesis addresses significant gaps in the current literature surrounding machine learning applications for energy demand prediction in energy management systems. By developing and implementing an advanced ensemble of machine learning algorithms specifically tailored for energy settings, this study aims to innovate in model selection and configuration to achieve precise and efficient energy demand prediction. Furthermore, a comprehensive analysis of algorithmic efficiency and efficacy within the context of energy management is undertaken, utilizing a diverse set of metrics to provide nuanced insights into performance. Through advanced hyperparameter tuning and feature selection techniques, the top-performing algorithm identified in the analysis is optimized to push the boundaries of optimization methodologies and achieve peak performance in energy demand prediction for buildings. In addition, several key research gaps have been identified:

Data integration: Effective methods for integrating data from various sources into a centralized system for better energy management are needed.

Predictive analytics: More accurate and effective predictive models are required to optimize energy usage and reduce costs.

Real-time monitoring and control: More advanced and efficient methods for real-time monitoring and controlling energy usage are necessary for optimal energy management.

Machine learning applications: Further research is needed to develop scalable and effective applications of machine learning in electrical energy management.

Cybersecurity: Effective methods for securing electrical energy management systems against cyber threats are essential as digital technologies become more prevalent.

Energy storage systems: More efficient and cost-effective energy storage systems are needed to enhance electrical energy management. A synopsis of Chapters 3 and 4's collective exploration of the complex terrain of data integration, predictive analytics, real-time monitoring and control, and the use of machine learning in electrical energy management is provided below. These chapters offer not just theoretical frameworks but also hands-on instructions for tackling these problems by creatively addressing each research gap. Additionally, the comprehensiveness of their solutions guarantees a holistic approach to improving energy management systems, advancing the industry toward better effectiveness, sustainability, and resilience in the face of new problems.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter explains the research methodology to fulfil the research objectives. The research methodology includes system, sample data, modification parameter setup, final parameter setup, machine and equipment involved, type of algorithm and forecasting analysis. The methods planned were essential to determine the direction and flow of this research.

3.2 Data-driven Electrical Energy Efficiency Management

Data-driven Electrical Energy Efficiency Management (D2EEM) refers to an approach in which the management and optimization of electrical energy consumption are driven by the analysis and insights derived from data. This methodology leverages advanced data analytics techniques, particularly machine learning algorithms, to make informed decisions and enhance the efficiency of electrical energy usage in various settings, such as buildings, industrial facilities, or power systems.

Key components of D2EEM typically include:

Data Collection: Gathering relevant data related to energy consumption, system parameters, and environmental factors. This data can be collected from sensors, smart meters, historical records, and other sources.

Data Analysis: Applying various data analysis techniques, including machine learning algorithms, to uncover patterns, correlations, and trends within the collected data. This analysis helps in understanding the factors influencing energy consumption.

Predictive Modeling: Developing models that can predict future energy consumption based on historical data and other relevant features. These models enable proactive decision-making for efficient energy management.

Optimization Strategies: Implementing optimization algorithms to identify the most efficient ways to allocate and utilize electrical energy resources. This may involve scheduling, load balancing, and other strategies to minimize waste and improve overall efficiency.

Continuous Monitoring and Adaptation: Establishing a feedback loop for continuous monitoring of energy usage and system performance. This allows for real-time adjustments and adaptation to changing conditions, ensuring ongoing energy efficiency.

3.3 Experimental settings

The time slot for DR events is one hour. The maximum number of requests that can be sent to a consumer for the period is 5, after which the consumer is filtered out of the selection process. Defined a flat supply threshold across all time slots, where AG equals 90% of the maximum demand of the DR Day. The expected reduction for every consumer is 40 percent of their total consumption. Initially, Δt is assigned zero and is updated in the subsequent iterations based on the responses in the previous iteration. The response and request indexes are created using a random function that takes values between 0.1 and 0.9, respectively. Experiments are run in a Stochastic Response Mode for 10 DR events. Each time a consumer gets selected, its requesting index and response index are updated using s 5 and 6. Consumer's expected reduction for every DR event is calculated using a function based on their average performance in the past DR events ±a % randomly, where takes values from 1 to 10, multiplied by 0.4. Experiments For evaluation, all possible combinations, i.e.,4C1,4C2,4C3 and 4C4 of features, are considered. For simplicity, while calculating the overall score, all features under consideration are given the same alpha values, i.e., one and rest to 0. All these combinations result in 15 approaches. The performance metrics are measured for all these approaches using s 10, 12, 13 mentioned in the thesis. The maximum value from T (Number of DR requests) is 5000, which occurs when DR Request is sent to every consumer in all DR events. The maximum risk can be obtained 100%, and worst-case unfairness will be close to 500.

3.4 Data set

The selection of a university campus, particularly IIT in India, as the case study for the research is appropriate. The unique characteristics of a university campus provide a diverse and complex environment for studying electrical energy efficiency management. Universities often have a mix of academic buildings, residential areas, laboratories, and recreational spaces, creating a challenging setting that reflects real-world scenarios.

The IIT campus, being a renowned educational institution, adds significance to the research due to its scale and diverse energy consumption patterns. Analyzing and optimizing energy efficiency in this context can offer valuable insights applicable to educational campuses globally, contributing to sustainable energy practices in the academic sector.

The choice aligns well with the research objectives, providing a practical and relevant scenario for implementing and testing the proposed data-driven electrical energy efficiency management approach. The findings from such a case study can have broader implications for similar large-scale settings, helping address energy challenges in educational institutions and beyond.

3.4.1 Data-Driven Energy Management

Our ongoing research in Smart Energy Management is primarily motivated by the objectives of achieving energy efficiency, reducing peak demand and ensuring optimal demand response control, while simultaneously meeting the thermal comfort needs of end-users. Our research activities have contributed significantly to the advancement of various thermal management and energy optimization techniques. To facilitate our data-intensive research, diverse datasets gathered from actual buildings within the IIT-Bombay Campus, encompassing multiple electrical and environmental parameters.

The following dataset obtained from an academic building on a campus. This four-story building is divided into three wings, and the dataset covers the period from December 2016 to July 2018.

All the datasets are available in CSV format, with clear and consistent field names provided in the respective files. The datasets are all recorded at one-minute granularity.

A detailed description of each field can be found here, and the complete dataset can be accessed here. Additionally, provided data completeness metrics for the datasets spanning the years 2016, 2017, and 2018.Data were collected for these rooms in the building:

The datasets provided here pertain to the energy consumption of the AC and lighting systems in Auditorium 1. This room can accommodate up to 200 individuals and features a chilled beam HVAC system. The HVAC system includes six compressor units, each with a capacity of 7.5 TR, and utilizes non-inverter technology. The cold air is supplied to each air mixing plenum through a set of three AC units and is then passed into the ducted beam. There are 18 ducts located across two beams, which are designed in an octagonal shape. Four ducts are on the inner side, while the remaining 14 are positioned on the outer side, through which cold air is circulated into the auditorium.

The dataset provided pertains to the energy consumption of the AC units in Auditorium2, which is a spacious room with a sloping floor with a seating capacity of 130 people. The room has 7 wall-mounted indoor units of split air conditioners installed at a high level on the wall.

The dataset provided contains information about the power consumption of the entire building, including ACs, lights, and plug level loads.

The dataset provided includes information on the power consumption of three sets of air conditioners - AC1, AC2, and AC3, which are installed in a cluster of 4 classrooms and two small labs. These rooms are located across two floors of a building. Combining the three sets of data will totally consume ACs in all the rooms.

These datasets contain information on the power consumption of various rooms and equipment in an academic building:

Auditorium1 AC, lights: This dataset contains the power consumption of Auditorium 1, a 200-seat-capacity room with a chilled beam HVAC system consisting of six non-inverter technology-based compressor units. The cold air is fed into each air mixing plenum from a group of three AC units and passed into the ducted beam. There are 18 ducts in two beams from which cold air is thrown into the auditorium.

Auditorium2 AC: This dataset contains the power consumption of Auditorium 2, a large room with a sloped floor that can accommodate 130 people. There are 7 wall-mounted indoor units of Split air conditioners placed at a high level on the wall.

Building Level AC, mains, plug level: This dataset contains the power consumption of ACs, lights, and plug level load of the entire building.

Classrooms AC1, AC2, AC3: This dataset contains the power consumption of a cluster of 4 classrooms and two small labs spread across two building floors. Combining the three sets will totally consume ACs in all these rooms.

Conference room AC, plug level: This dataset contains power consumption in a typical conference facility. The conference room is a medium-sized room that can accommodate about 25 people. The AC consumption here is the consumption of 3 ACs within this room. The plug-level load is contributed by one PC, projector and video conferencing equipment used mainly during presentations in this room.

Floor AC, lights, plug level: This dataset comprises the consumption of an entire floor consisting of 5 labs, two classrooms (1 extensive and one small), a common area, and two washrooms.

Lab 1 AC, lights, plug level 1, plug level 2: This dataset contains the power consumption of Lab 1, which is a big room with centralized Duct AC with eight outlets. To obtain plug level load of this room, combine the two datasets for plug level data (plug level 1, plug level 2).

Lab 2 AC, lights, plug level: This dataset contains the power consumption of Lab 2, a big room with seven window-ACs and one split AC.

Lab 3 AC, lights, plug level: This dataset contains the power consumption of Lab 3, which is a big room with one duct AC and two window -ACs.

Lab 4 AC, lights, plug level: This dataset contains the power consumption of Lab 4, a medium-sized room with two indoor blower units. The actual power consumption happens at the outdoor unit, which supplies cooled air to the indoor units. Please refer to the Lab ODU data below to compute the actual AC consumption.

Lab AC Outdoor Units ODU1, ODU2, ODU3: This dataset consists of the outdoor units of ACs installed in 8 labs across 2 floors.

* The labs are occupied throughout the day and sometimes at night.

Office AC and lights data are included in this dataset, representing the power consumption of one office room with 3 split ACs and one window AC. This room is frequently visited during the day and remains unoccupied at night. The Server room AC and plug level dataset represents the energy consumption of a departmental server room with 8 ACs, several rack-mounted servers, and routers. The Small server room AC and plug level dataset represents the energy consumption of the smaller server room in the main server room, which contains 3 ACs. Finally, the Wing C AC, lights, and plug level dataset represents the energy consumption of one wing of the building, including offices, labs, classrooms, washrooms, stairways, and common areas on ground + 4 floors.

3.4.2 Data Collections

The dataset contains information on electricity consumption in a high-rise residential building within the IIT Bombay campus, covering the period from December 2016 to January 2018. The building comprises 60 3BHK (3 Bedrooms, I Hall, and a Kitchen) apartments, each equipped with a smart meter that records data at a sampling interval of 5-8 seconds. The data provided in the link has been aggregated to an hourly granularity. However, a sample dataset of two apartments, with a sampling interval of 5-8 seconds, is available under "Sample Monthly Dataset". All timestamps in the dataset are in Indian Standard Time (GMT+5.30), and India does not observe daylight saving time. To protect privacy, the apartments are anonymous, and 39 CSV files are included in the folder, each representing an apartment. Apartments with significant data loss have been removed from the list, and the CSV files contain these headers:

Unix Time stamp (epochs) - TS

Voltage of phase 1 (V) - V1

Voltage of phase 2 (V) - V2

Voltage of phase 3 (V) - V3

Electricity consumption of phase 1 (Wh) - W1

Electricity consumption of phase 2 (Wh) - W2

Electricity consumption of phase 3 (Wh) - W3

Virtual Dataset

Additional Headers

Virtual Apartment ID - Virtual Apartment

Date in YYYY-MM-DD – Date

Time in HH: MM: SS - Time

Sum of W1 + W2 + W3 (Wh) - Energy

Sample Monthly Dataset headers in the CSV files:

timestamp received (TS_RECV)

serial number (Srl)

timestamp (TS)

voltage from Phase1 to neutral (V1)

voltage from Phase2 to neutral (V2)

voltage from Phase3 to neutral (V3)

current for Phase1 (A1)

current for Phase2 (A2)

current for Phase3 (A3)

active power of Phase1 (W1)

active power of Phase2 (W2)

active power of Phase3 (W3)

Voltage*Current for Phase 1 (VA1)

Voltage*Current for Phase 2 (VA2)

Voltage*Current for Phase 3 (VA3)

reactive power in phase 1 (VAR1)

reactive power in phase 2 (VAR2)

reactive power in phase 3 (VAR3)

Power Factor of Phase1 (PF1)

Power Factor of Phase2 (PF2)

Power Factor of Phase3 (PF3)

angle in phase 1 (Ang1)

angle in phase 2 (Ang2)

PF3r (PF3r)

Ang1r (Ang1r)

Ang2r (Ang2r)

angle in phase 3 (Ang3)

average of V1,V2 and V3 (AvgV)

sum of V1, V2 and V3 (SumV)

average of A1, A2, A3 (AvgA)

sum of A1, A2, A3 (SumA)

average of W1, W2, W3 (AvgW)

sum of W1, W2, W3 (SumW)

average of VA1, VA2 and VA3 (AvgVA)

sum of VA1, VA2 and VA3 (SumVA)

average of VAR1, VAR2 and VAR3 (AvgVAr)

sum of VAR1, VAR2 and VAR3 (SumVAr)

average of PF1, PF2 and PF3 (AvgPF)

sum of PF1, PF2 and PF3 (SumPF)

average of Ang1, Ang2 and Ang3 (AvgAng)

sum of Ang1, Ang2 and Ang3 (SumAng)

Frequency (F)

Energy (FwdWh)

Table 3.1 Attributes for datasheet

Number	Symbol	Number	Symbol	Number	Symbol	Number	Symbol
1.	V1	8.	VA1	15.	W	22.	PF3
2.	V2	9.	VA2	16.	VAR1	23.	PF
3.	V3	10.	VA3	17.	VAR2	24.	FwdWh
4.	A1	11.	VA	18.	VAR3	25.	FwdVAh
5.	A2	12.	W1	19.	VAR		
6.	A3	13.	W2	20.	PF1		
7.	A	14.	W3	21.	PF2		

3.5 Evaluation Framework

The study employed a quad-folded cascading methodology, as illustrated in Figure 1. Initially, the SEIL dataset was used to consider the total energy consumption at the building level, including the auditorium, classroom, conference room, building floor, labs, offices, server room, and sub-server room. In the second phase, the building-level dataset was randomly divided into 70% training samples and 30% testing samples, and 24 machine-learning algorithms were trained using the training set. The third phase involved evaluating the parametric performance of each ML algorithm, considering training parameters like RMSE, R-squared, MSE, MAE, and Prediction Speed, for both the training and testing phases. Finally, the algorithms were ranked based on efficacy and efficiency. Figures 3.1 provided a functional inside view of the training and testing phase, which constituted the inside workings of Phase 2 in the proposed methodology.

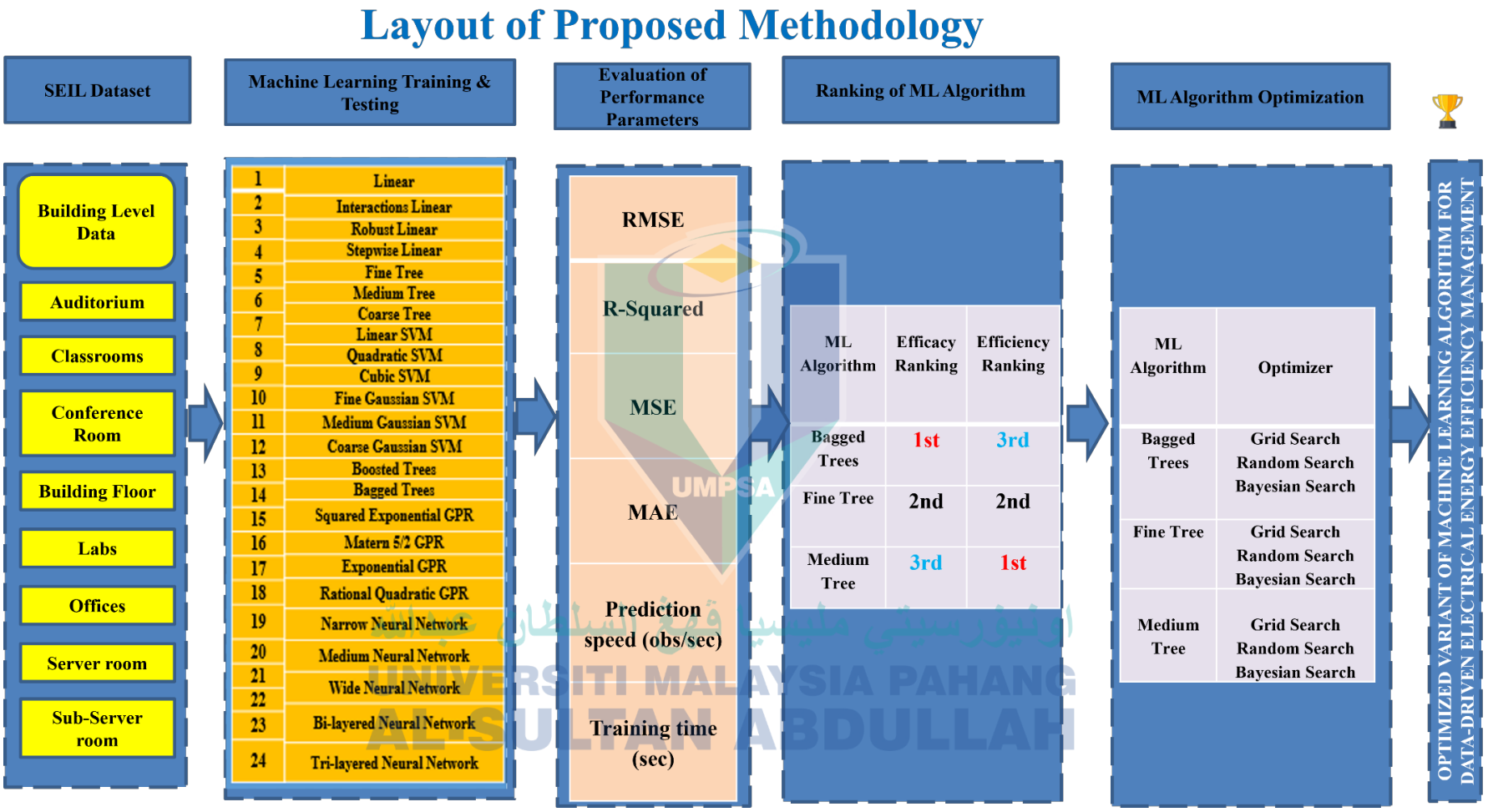


Figure 3.1 Layout of methodology

Process involves in optimisation.

3.6 Dataset Description

The SEIL (Smart Energy in Informatics Lab) dataset was used to consider the total energy consumption at the building level. The dataset includes information on various areas within the building, such as the auditorium, classroom, conference room, building floor, labs, offices, server room, and sub-server room. It contains historical energy consumption data and relevant features that influence energy consumption.

3.6.1 Data Preprocessing:

Before conducting the analysis, the SEIL dataset underwent preprocessing to handle missing values, normalize features, and remove any outliers that could adversely affect the performance of the machine learning algorithms.

3.6.2 Data Splitting:

The building-level dataset was randomly divided into two subsets: a training set and a testing set. The training set contained 70% of the data, while the testing set contained the remaining 30%. This splitting ensured that the machine learning algorithms were trained on enough data while also allowing robust evaluation on unseen data.

3.6.3 Machine Learning Algorithm Selection:

A total of 24 machine learning algorithms were chosen for energy consumption prediction. These algorithms were carefully selected to cover a diverse range of approaches, including regression, ensemble methods, and deep learning, to compare their performance on the task at hand. Table 3.2 explain different optimization parameters.

Table 3.2 Optimization parameters

Rank	ML Algorithm	Optimization parameter
1	Baggage Tree	Number of Bags: 50, Max Features: 0.8
2	Fine Tree	Split Criterion: Gini, Max Depth: 8
3	Medium Tree	Max Features: Auto, Min Samples Split: 2

3.6.4 Model Training:

Each of the 24 machine learning algorithms was trained using the training set. During training, the algorithms learned patterns and relationships in the data, allowing them to predict energy consumption accurately.

3.6.5 Evaluation Metrics:

In the third phase, the parametric performance of each machine learning algorithm was evaluated using multiple metrics. The following evaluation metrics were considered for both the training and testing phases:

a. Root Mean Squared Error (RMSE): Measures the average difference between predicted and actual energy consumption values.

b. R-squared (R^2): Assesses the proportion of variance in the dependent variable (energy consumption) explained by the independent variables.

c. Mean Squared Error (MSE): Calculates the average squared difference between predicted and actual values.

d. Mean Absolute Error (MAE): Computes the average absolute difference between predicted and actual values.

e. Prediction Speed: Measures the time taken by each algorithm to make predictions.

3.6.5.1 Ranking of Algorithms:

After evaluating the performance of each algorithm based on the metrics mentioned above, the algorithms were ranked according to their efficacy and efficiency in predicting energy consumption. Efficacy was determined by their accuracy in predicting energy consumption, while efficiency was determined by their prediction speed.

3.6.5.2 Visualization:

Figure 3.1 presented a functional inside view of the training and testing phases. This figure provided insights into the workings of the machine learning algorithms during the training and testing phases.

3.7 Optimization process

There are many optimization techniques used for the improvement of algorithms for best-predicted result measures, as described below,

i. Taguchi method

ii. Response Surface Methodology

iii. Artificial Neural Network

iv. Genetic Algorithm

v. Grey Relational Analysis (GRA)

vi. Fuzzy Logic

vii. Particle Swarm Optimization

viii. Simulated Annealing

ix. Principle Component Analysis

x. Technique for Order of Preference by Similarity to Ideal Solution

The following figure 3.2 shows the optimization process.

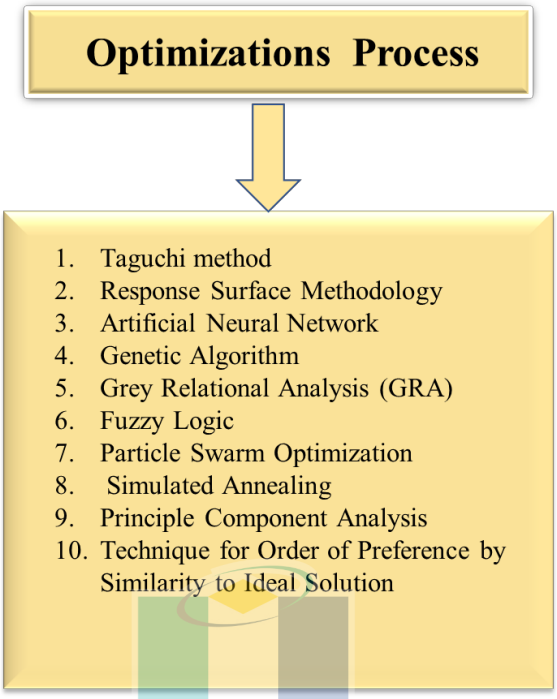


Figure 3.2 Optimization methods

3.7.2 Taguchi Method

The Taguchi method is a scientifically well-organized mechanism for evaluating and implementing improvements in products or processes. This perfection aims to improve the desired characteristics by studying the key variables controlling the process and optimizing the procedures to yield the best results. Taguchi recommends an orthogonal array (OA) for laying out of experiments. To design an experiment, select the most suitable OA to assign the parameters and interactions of columns. Taguchi suggested that Linear graphs and triangular tables make the assignment of parameters simple (Vikas, Roy, and Kumar 2014). The analysis of variance (ANOVA) is a statistical treatment commonly useful for the experimental results in determining the percentage contribution to each parameter against a stated confidence level.

A study of the ANOVA table for a analysis helps determine the parameters needing control (Ross Phillips, 1996). Taguchi method is a statistical measure of performance named signal-to-noise ratio (S/N ratio). The S/N ratio can measure the deviation of the performance characteristics from the desired values. Performance

characteristics in the analysis of the S/N ratio are of three categories as follows (Vikas, Roy and Kumar 2014).

Larger-the-better characteristics

$$\frac{S}{N} = -10\,log(\frac{1}{n}\sum_{i=1}^{n} y_i \frac{1}{2}) \quad 3.1$$

Smaller-the-better characteristics

$$\frac{S}{N} = -10\,log(\frac{1}{n}\sum_{i=1}^{n} y_i 2) \quad 3.2$$

Nominal-the-better characteristics

$$\frac{S}{N} = -10log\left(\frac{y}{S_y^2}\right) \quad 3.3$$

Where y_i is the experimentally observed value and n is the repeated number of each experiment. y is the average of observed data and S_y^2 is the variance of y for each type of characteristics, with the above S/N transformation, the higher the S/N ratio the better is the result. Optimization of performance measures using parameter design of the Taguchi method is summarized (Muhammad et al. 2012) in the steps as shown in flow chart as shown in figure 3.3.

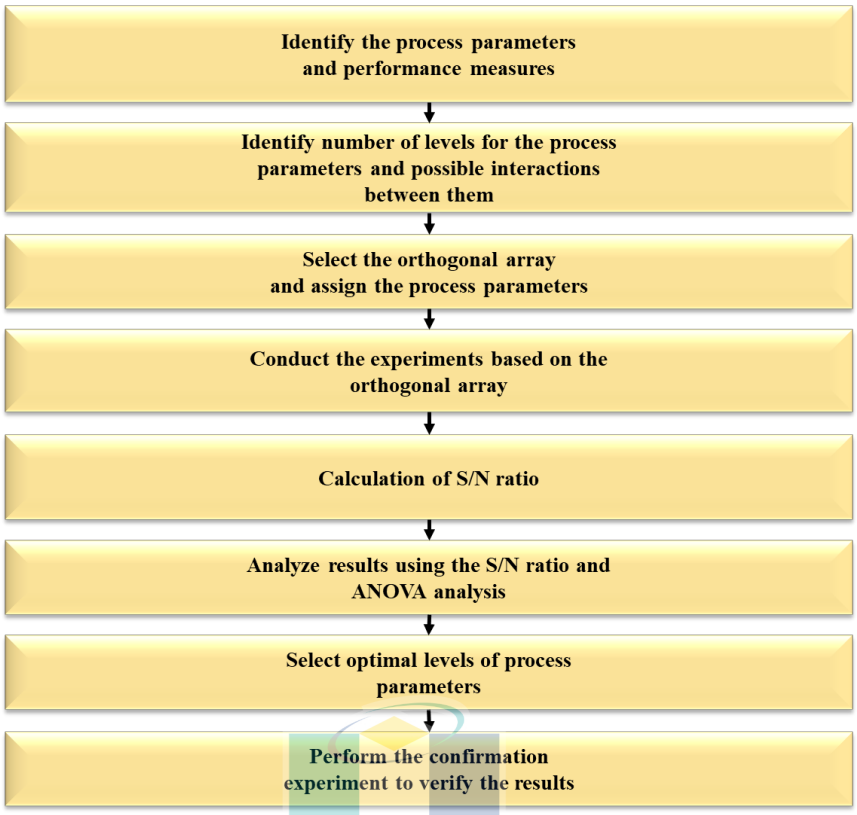


Figure 3.3 Performance measures flow using the Taguchi technique.

3.7.3 Response Surface Methodology

Response Surface Methodology is a static and mathematical technique utilized to model and analyse a process affected by various variables. The parameters that impact the process are known as dependent variables, while the outcomes are called independent variables, according to Khuri (2017). For instance, the hardness of meat can be influenced by cooking time (X1) and cooking temperature (X2). The meat's firmness can be altered under any treatment combination of X1 and X2 (Refinery et al., 2016; Rupi et al., 2015).

Hence, when treatments can take continuous values of time and temperature, Response Surface Methodology is useful for developing, improving, and optimizing the response variable. In meat hardness, time (X1) and temperature (X2) are the parameters affecting the response variable and can be adjusted to achieve the desired meat hardness (Y). This relationship can be expressed as the dependent variable Y being a function of X1 and X2.

$$\mathbf{Y = f(X1) + f(X2) + e} \quad 3.4$$

where (Y) is the response (dependent variable), $(X1)$ and $(X2)$ are independent variables and (e) is the experimental error.

Response surface methodology (RSM) is a technique that uses surface placement to understand the topography of the response surface. It aims to identify the region where the most appropriate response occurs and find the optimal operating conditions for a system under study. RSM employs two main experimental designs, namely Box-Behnken designs (BBD) and central composite designs (CCD). Recently, central composite rotatable design (CCRD) and face central composite design (FCCD) have also been utilized in optimization studies. To fit a statistical model, experimental data are evaluated using linear, quadratic, cubic or 2FI (two-factor interaction) models. The constant terms represent the coefficients of the model, including linear coefficients for independent variables (A, B, and C), interactive term coefficients (AB, AC, and BC), and quadratic term coefficients (A2, B2, and C2). To ensure model adequacy, correlation coefficient (R^2), adjusted determination coefficient (Adj-R^2), and adequate precision are used, and the model is considered adequate when its P-value < 0.05, lack of fit P-value > 0.05, R^2 > 0.9, and Adeq Precision > 4. Statistical significance of differences between means can be tested using analysis of variance (ANOVA).

The design of experiments (DoE) is a crucial aspect of RSM, as it involves selecting points for examining the response. The mathematical model of the process is closely related to the design of experiments, and thus, the choice of experiment design significantly affects the accuracy of the response surface construction. RSM offers several advantages, including the ability to determine the interaction between independent variables, develop a mathematical model of the system, and save time and costs by reducing the number of trials needed. However, a significant disadvantage of the method is that it assumes a polynomial model at the second level, which may not be suitable for all systems with curvature. Therefore, experimental verification of the estimated values in the model is essential to ensure its accuracy. During the initial stage of the DoE, screening experiments are conducted to identify the variables with a significant impact on the response. If numerous variables influence the response, the variables that significantly affect the response are determined. The goal is to identify the

design variables that will be further investigated (Myers et al., 2004). The steps adopted in RSM are briefly presented in Figure 3.4.

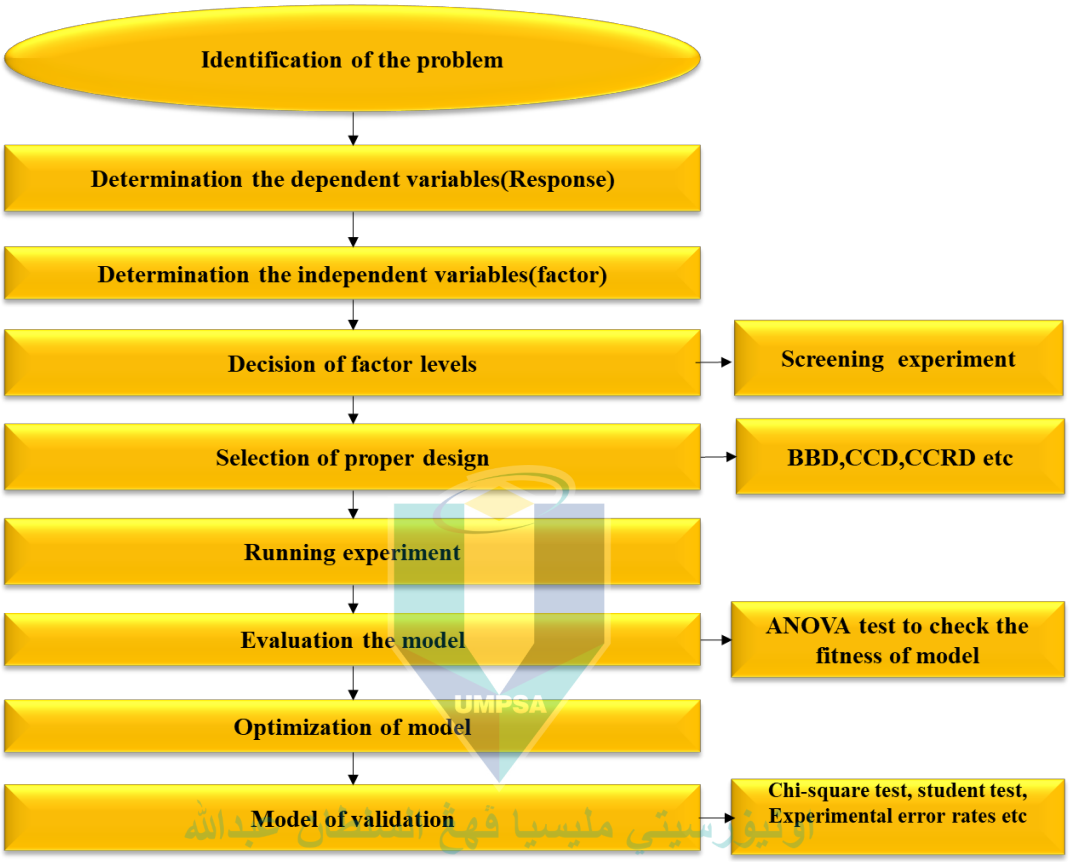


Figure 3.4 Steps for surface response method.

3.7.4 Artificial Neural Network (ANN)

An artificial neural network is a model that runs like a human brain by using many neurons consecutively and collects information through a learning process (X. Yang 1AD). Complex problems whose analytical or numerical solutions are difficult to obtain can be solved by utilizing the adaptive learning ability of neural networks (Rafiq, Bugmann, and Easterbrook 2001). Generally, the design of a neural network comprises three main steps: configuration (i) how layers are organized and connected; learning (ii) how information is stored; generalization (iii) how the neural network produces reasonable outputs for inputs not found in training (Haykin and Simon 1999).

The multi-layer perceptions neural network is formed from numerous neurons with a parallel connection, which are jointed in several layers. The structure of this network contains the network's input data, the number of hidden middle layers with numerous neurons in each layer and an external layer with neurons connected to the output. ANNs are broadly classified into feedforward and backpropagation networks. Feedforward networks are those in which computation flow from the input nodes to the output nodes in a sequence. In a back-propagation network, signals may propagate from the output of any neuron to the input of any neuron. The artificial neuron evaluates the inputs and determines the strength of each by its weighting factor. The result of the summation function for all the weighted inputs can be treated as an input to an activation function from which the output of the neuron is evaluated. Then the output of the neuron is transmitted to subsequent neurons along the outgoing connections to serve as an input to them. When an input is presented and propagated forward through the neural network to compute an output for each neuron, the Mean Square (MS) error between the desired output and actual output is computed to reduce the MS error rapidly. An iterative error reduction of the gradient-descent method by adding a momentum term is performed (Rumelhart, Widrow, and Lehr (1994). After the learning process is finished, the neural network memorizes all the adjusted weights and is ready to predict the machining performances based on the knowledge obtained from the learning process (Lu et al. 2009). A simple neural network can be represented as shown in figure 3.5 below.

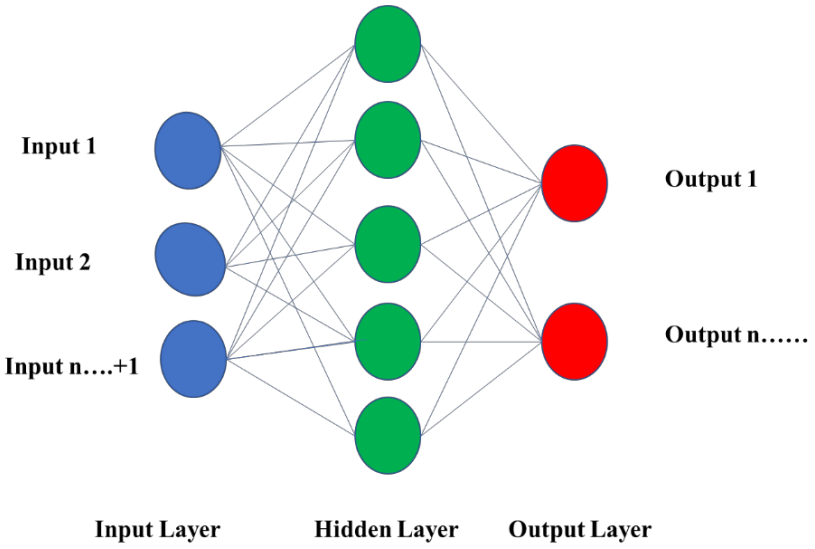


Figure 3.5 A example of a simple layer structure of ANN

3.7.5 Genetic Algorithm (GA)

The GA was developed on the probabilistic basis that the global optimum is searched in a random and parallel manner through the operations of reproduction, crossover and mutation (Sastry, Goldberg, and Kendall 2005). These algorithms maintain and control a population of solutions and implement their search for better solutions based on the 'survival of the fittest strategy. GA can solve linear and nonlinear problems by exploring all regions of the state space and exploiting promising areas with a set of potential solutions or chromosomes (usually as bit strings) randomly generated or selected. The entire set of these chromosomes comprises a population. Figure 3.6 shows a flow chart for a simple GA (Chang et al. 2004).

As depicted in Figure 3.6, a GA starts by randomly initializing the parent chromosomes represented in string, and the fitness of these chromosomes is then calculated based on the objective function. The reproduction process aims to allow the genetic information stored in the artificial strings to have functional fitness and survive the next generation. Crossover involves splitting up two chromosomes and combining one-half of each chromosome with the other pair. Mutation involves flipping a single bit of a chromosome. The chromosomes are then evaluated using a specific fitness criterion, and the best ones are kept while the others are removed. The process is repeated until the solution with the best fitness to meet the objective function criteria is received.

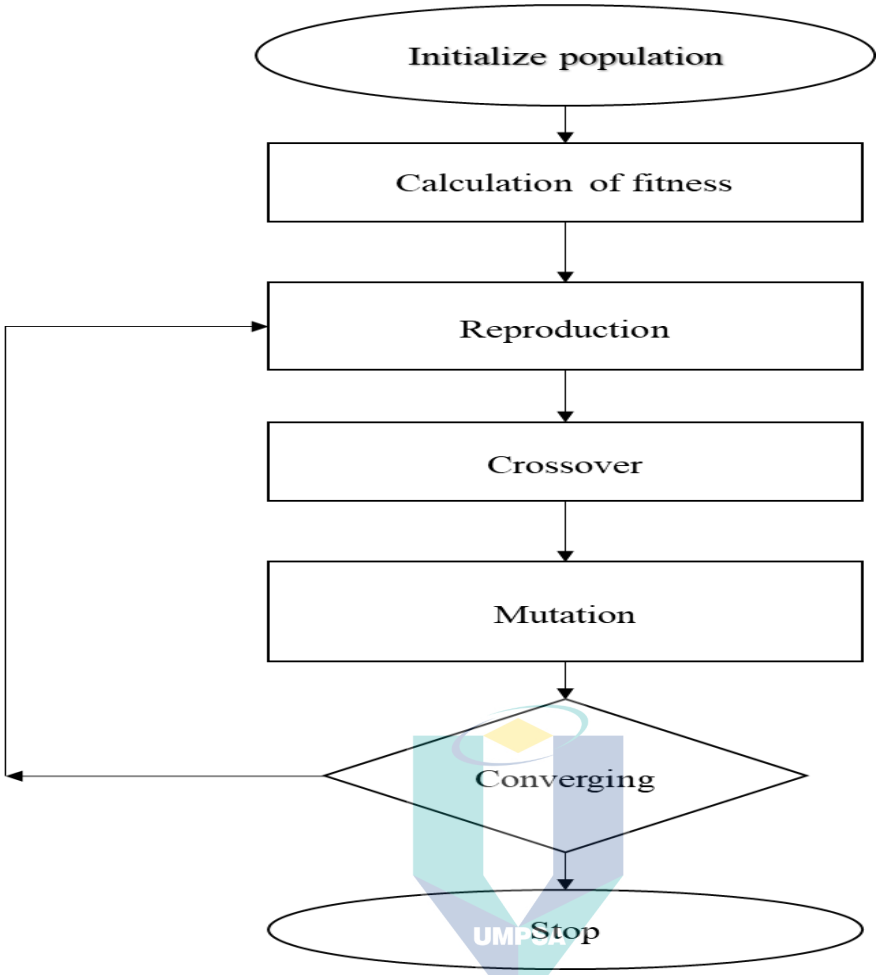


Figure 3.6 Flow of GA algorithm.

3.7.6 Grey Relational Analysis (GRA)

The grey Relational Analysis theory was developed for the new methods for solving the complicated interrelationship among the multiple performing characteristics. The grey system theory is an efficient technique which requires limited information to estimate the behaviour of an uncertainty system & discrete data problem. Figure 3.9 shows simple steps in the GRA. Normalizing involves transforming the original sequence into an identical sequence. This is known as grey relational generating (Murugesan and Balamuruga 2012). There are three conditions of normalization.

1. Lower is better.

$$\mathbf{X_i(k)} = \frac{\mathbf{Max\ X_i(k) - X_i(k)}}{\mathbf{Max\ X_i(k) - Min\ X_i(k)}} \qquad 3.5$$

2. Higher is better.

$$X_i(k) = \frac{X_i(k) - Min\ X_i(k)}{Max\ X_i(k) - Min\ X_i(k)} \quad 3.6$$

3. Nominal is better.

$$X_i(k) = \frac{1 - |X_i(k) - X_o b(k)|}{Max\ X_i(k) - Min\ X_o b(k)} \quad 3.7$$

Where $I = 1, 2, \ldots n$; $k = 1, 2, \ldots m$; $Xi^*(k)$ is the normalized value of the k[th] element in the i^{th} sequence, $X_0 b\ (k)$ i the desired value of the ik^h quality characteristic, max $Xi^*(k)$ is the largest value of $Xi\ (k)$, and min $Xi^*(k)$ is the smallest value of $Xi\ (k)$, n is the number of experiments and m is the number of quality characteristics.

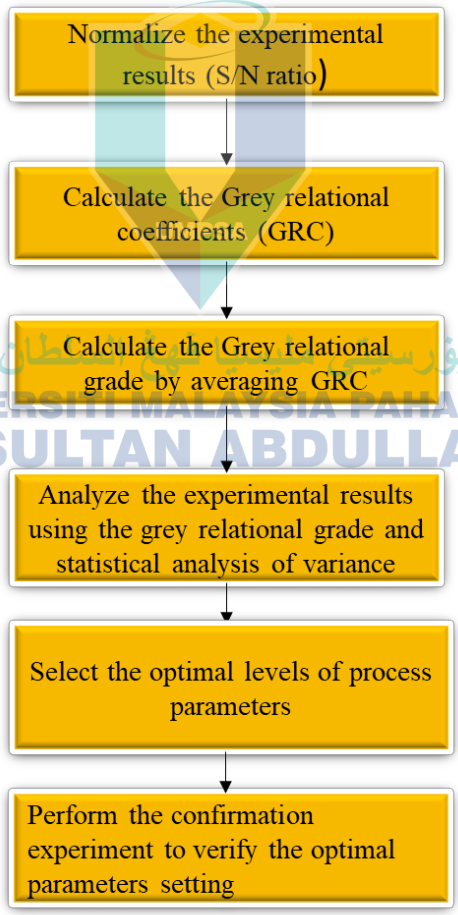


Figure 3.7 An example of standard steps adopted in GRA

3.7.7 Particle swarm optimisation

Particle swarm optimization (PSO) is an evolutionary computational technique; Particle swarm optimization was developed in 1995 by Kennedy and Eberhart (Slowik 2011). This optimization and search technique model the natural swarm behaviour seen in many species of birds returning to roost, a group of fish, the swarm of bees, etc. The PSO may find optimal (or near-optimal) solutions to numerical and qualitative problems (Talbi and Batouche n.d.). PSO methods are inspired by particles moving around in the defined search space. The individuals in a PSO have a position and a velocity. The PSO method remembers the best position found by any particle.

Additionally, each particle remembers its own previously best-found position. A particle moves through the specified solution space along a trajectory defined by its velocity, the draw to return to a previous promising search area, and an attraction to the best location discovered by its close neighbours. One advantage of particle swarm optimisation over other derivative-free methods is the reduced number of parameters to tune and constraint acceptance. Particle swarm optimization has been used for a wide range of search applications and specific optimization tasks. PSO has been successfully applied in many areas: Function optimization, Artificial neural network training, Proportional and integral fuzzy system control, and Other near-optimal search and optimization areas where GA can be applied.

The basic structure of any particle in a selected population consists of five components.

1. $\underset{x}{\rightarrow}$ is a vector containing the current location in the solution space? The size of $\underset{x}{\rightarrow}$ is defined by the number of variables used by the problem being solved.
2. Fitness is the quality of the solution represented by the vector $\underset{x}{\rightarrow}$, as computed by a problem-specific evaluation function.
3. $\underset{V}{\rightarrow}$ is a vector containing the velocity for each dimension of $\underset{x}{\rightarrow}$. The velocity of a dimension is the step size that the corresponding $\underset{x}{\rightarrow}$ value will change into at the next iteration. Changing the $\underset{V}{\rightarrow}$ values changes the direction the particle will move through in the search space, causing the particle to make a turn. The velocity vector is used to control the range and resolution of the search.

4. P_{best} is the fitness value of the best solution yet found by a particle.

5. $\underset{P}{\rightarrow}$ is the copy of the $\underset{x}{\rightarrow}$ for the location that generated the particle's P_{best}. Jointly, P_{best} and $\underset{x}{\rightarrow}$ Comprise the particle's memory, which controls the particle's return to a definite search region.

6. Each particle is also aware of the current best fitness in the neighbourhood for any iteration. A neighbourhood may consist of some small group of particles, in which case the neighbourhoods overlap, and every particle is in multiple neighbourhoods. Particles in a swarm are related socially; each particle is a member of one or more neighbourhoods. Each individual tries to emulate the behaviour of the best of their neighbours. Everyone can be thought of as moving through the feature space with a velocity vector that its neighbour's influence. Figure 3.8 shows the simple PSO algorithms.

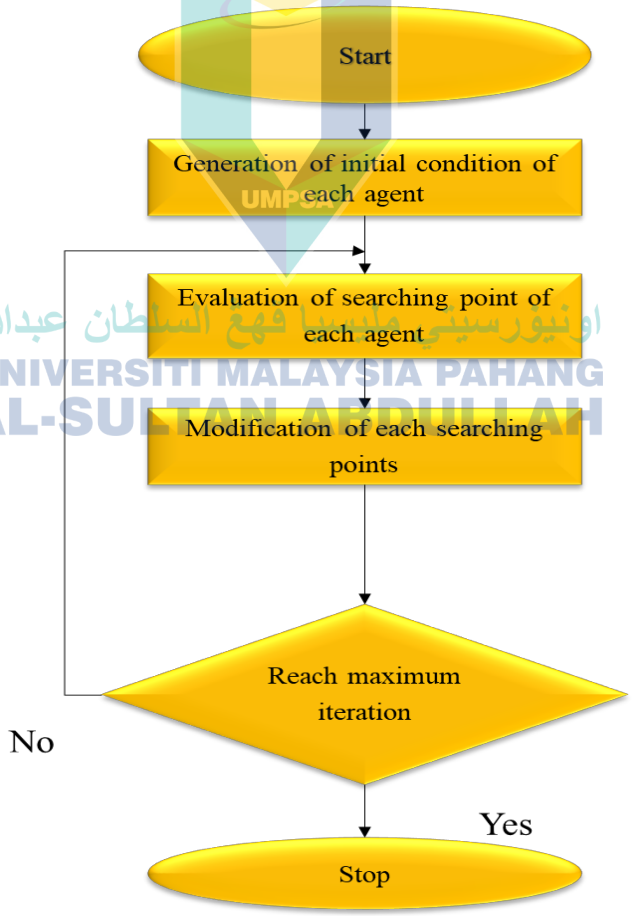


Figure 3.8 Flow chart of simple PSO algorithms.

3.7.8 Simulated Annealing

Simulated Annealing (SA) is an effective and general form of optimization. It helps find global optima in large numbers of local optima. "Annealing" refers to an analogy with thermodynamics, specifically how metals cool and anneal. Simulated annealing uses the objective function of an optimization problem instead of the energy of a material (Zhan et al. 2016). Implementation of SA is simple. The algorithm is hill-climbing, except it picks a random move instead of the best one. If the selected move improves the solution, then it is always accepted. Otherwise, the algorithm makes a move anyway with some probability of less than 1. The probability decreases exponentially with the "badness" of the move, which is the amount ΔE by which the solution is worsened.

3.7.9 Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities, each of which takes on various numerical values) into a set of linearly uncorrelated variables called principal components. This transformation is defined so the first principal component has the largest possible variance (that is, accounts for the variability in the data), and each succeeding component has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

3.8 Architect / Pseudocode of top 3 models

3.8.1 Fine Trees

Fine Trees is a machine learning algorithm used for decision tree induction. The following is a simple example of the pseudocode for the Fine Trees algorithm:

1. Fine Trees is a machine learning algorithm used for decision tree induction. The following is a simple example of the pseudocode for Fine Trees algorithm:
2. Initialize the decision tree with the root node.
3. Split the data set into smaller subsets using a split criterion (e.g. information gain)

4. For each subset:

   a. Evaluate the impurity of the data (e.g. using Gini impurity)

   b. If the impurity is below a certain threshold, create a leaf node and store the predicted class label.

   c. If the impurity is above the threshold, repeat the process from step 2 on the subset.

5. Repeat the process for all subsets until the stopping criteria is met (e.g. all data belong to the same class or a maximum depth has been reached)

   Return the decision tree.

3.8.2 Architecture of fine tree:

Fine Trees is a type of decision tree algorithm used in machine learning for classification and regression tasks. It's an extension of the classic decision tree algorithms that aim to produce smaller and more interpretable trees by avoiding overfitting and making the trees more robust to noisy data.

The architecture of Fine Trees can be thought of as interconnected nodes, where each node represents a decision or a prediction. Each node in the tree splits the data into smaller subsets based on a certain feature, and the impurity of the data is evaluated in each subset. If the impurity is below a certain threshold, a leaf node is created, and the prediction is made based on the class label or numeric value in that subset. The process continues until the stopping criteria are met. The Fine Trees algorithm typically uses a more sophisticated split criterion and impurity evaluation method than the classic decision tree algorithms, resulting in trees that are more accurate and less prone to overfitting. The final decision tree is a graphical representation of the series of decisions and predictions made by the algorithm, and it's used to make predictions on new, unseen data.

In the Classification Learner App in MATLAB, Fine trees is one of the available machine learning algorithms for binary or multi-class classification problems. Fine trees are an ensemble method that builds multiple decision trees on the data and combines the predictions of the individual trees to make the final prediction.

The following hyper-parameters can be optimized for fine trees in the Classification Learner App:

Number of Trees: This hyper-parameter controls the number of decision trees in the ensemble. Increasing the number of trees typically results in improved performance, but also increases the computational cost and the risk of overfitting.

MinLeaf Size: This hyper-parameter controls the minimum number of samples required to be at a leaf node in the decision tree. Increasing the value of MinLeaf Size results in smaller and simpler trees, which can reduce the risk of overfitting but may also decrease the accuracy of the model.

Split Criterion: This hyper-parameter determines the criterion used to split the nodes in the decision tree. The options are 'gdi', 'twoing', or 'deviance'. MaxNumSplits: This hyper-parameter controls the maximum number of splits in the decision tree. Increasing the value of MaxNumSplits results in more complex trees, which can increase the accuracy of the model but also increases the risk of overfitting.

The Classification Learner App provides several options for hyper-parameter optionor, including grid search, Bayesian optimization, random search, and two-phase optimization, as described in my previous answer. Can use these methods to find the optimal values for the hyper-parameters of the fine tree algorithm in the app.

Pseudo code

Input: training data set, number of trees (n_trees)

Output: list of n_trees decision trees

For i in 1 to n_trees:

Sample data with replacement from the training set to create a new training set

sample_data = random sampling of the training data with replacement

Train a decision tree on the sampled data

tree = fit a decision tree to sample_data

Add the trained tree to the list of trees

add tree to list of trees

Return the list of trees

3.8.3 Architecture of Medium Trees

A decision tree is built by recursively splitting the data into smaller and smaller subsets based on the values of the input features. At each node in the tree, the feature that best splits the data is chosen and the tree branches based on the different values of that feature. The process continues until a stopping criterion is met, such as a minimum number of samples in a leaf node or a maximum tree depth.

The final decision tree can be thought of as decisions or "if-then" statements, where each node in the tree represents a decision based on the values of the input features, and each leaf node represents a prediction for the target variable. The architecture of the decision tree is determined by the features chosen for each split and the stopping criterion used to grow the tree.

The architecture of decision trees in machine learning finds the relationships between the input features and the target variable to make accurate predictions.

In the Classification Learner App in MATLAB, Medium trees is one of the available machine learning algorithms for binary or multi-class classification problems. Medium trees are a variant of the decision tree algorithm that balances the trade-off between accuracy and computational cost by using a medium-sized tree.

The following hyper-parameters can be optimized for medium trees in the Classification Learner App:

MinLeafSize: This hyper-parameter controls the minimum number of samples required to be at a leaf node in the decision tree. Increasing the value of MinLeafSize result in smaller and simpler trees, which can reduce the risk of overfitting but may also decrease the accuracy of the model. SplitCriterion: This hyper-parameter determines the criterion used to split the nodes in the decision tree. The options are 'gdi', 'twoing', or 'deviance'.

MaxNumSplits: This hyper-parameter controls the maximum number of splits in the decision tree. Increasing the value of MaxNumSplits results in more complex trees, which can increase the accuracy of the model but also increases the risk of overfitting.

The Classification Learner App provides several options for hyper-parameter optimlon, including grid search, Bayesian optimization, random search, and two-phase optimization.

3.9 Pseudo code of medium tree

function create_decision_tree(data, features, target, min_samples, max_depth, current_depth):

check if the stopping criteria are met

if the number of samples in data is less than min_samples or current_depth >= max_depth:

return create_leaf_node(data, target)

find the best feature to split the data on

best_feature = find_best_feature(data, features, target)

create a decision node for the best feature

decision_node = create_decision_node(best_feature)

split the data based on the best feature

for each value of best_feature:

subset = data with best_feature equal to value

child = create_decision_tree(subset, features, target, min_samples, max_depth, current_depth + 1)

add child to decision_node

return decision_node

```
function create_leaf_node(data, target):
  # calculate the target value for the leaf node
  target_value = average of target in data
  return create_node(target_value)
function find_best_feature(data, features, target):
  best_feature = None
  best_score = -inf
  for each feature in features:
    score = calculate_split_score(data, feature, target)
    if score > best_score:
      best_score = score
      best_feature = feature
  return best_feature
function calculate_split_score(data, feature, target):
  # calculate a score for the feature based on the target variable
  # such as the reduction in variance or information gain
  return score
```

"Bagged Trees" is short for "Bootstrapped Aggregated Trees," which is a type of ensemble learning in machine learning. Ensemble learning involves combining multiple individual models to create a more robust and accurate overall model.

3.10 Architecture of Bagged Trees

In the Bagged Trees, the idea is to train multiple decision trees on different subsets of the training data, where each subset is created by randomly sampling the data with replacement. The subsets are often called bootstrapped samples. The final prediction for a new instance is obtained by aggregating the predictions of all the individual trees, such as by taking a majority vote for classification or averaging for regression.

Using multiple trees allows Bagged Trees to capture a variety of relationships between the input features and the target variable, and to reduce the impact of overfitting, which can occur when a single decision tree is trained on the full data set. The architecture of Bagged Trees in machine learning combines the strengths of multiple decision trees, while mitigating their weaknesses, to create a more accurate and robust overall model.

3.11 Pseudo code of Bagged tress

```
function train_bagged_trees(data, features, target, num_trees, sample_ratio):
  trees = []
  for i = 1 to num_trees:
    sample = random sample of data with replacement, size = sample_ratio * size of data
    tree = train_decision_tree(sample, features, target)
    add tree to trees
  return trees
function predict_bagged_trees(trees, instance):
  predictions = []
  for each tree in trees:
```

prediction = predict_decision_tree(tree, instance)

add prediction to predictions

return majority vote of predictions.

3.12 Pseudo code of optimization methods used for the best candidate of ML algorithms Grid search and Random search

The optimization method used to find the best candidate ML algorithm depends on the specific problem and type of ML algorithm being considered. Common optimization methods used in ML include grid search, random search, and gradient-based optimization.

3.12.1 Pseudocode of Grid search

function grid_search(data, features, target, algorithm_params, metric):

best_params = None

best_score = -inf

for each combination of params in algorithm_params:

model = train_algorithm(data, features, target, params)

score = evaluate_model(model, data, target, metric)

if score > best_score:

best_params = params

best_score = score

return best_params, best_score

Pseudocode of optimization methods used for the best candidate of random search function random_search(data, features, target, algorithm_params, metric, num_iterations):

```
best_params = None
best_score = -inf
for i = 1 to num_iterations:
  params = random sample of algorithm_params
  model = train_algorithm(data, features, target, params)
  score = evaluate_model(model, data, target, metric)
  if score > best_score:
    best_params = params
    best_score = score
return best_params, best_score
```

3.13 Proposed optimized ML model

To optimize an ML model based on a pseudo code, one needs to take these steps:

Split the data into training and testing sets. That uses the training set to train the model and the testing set to evaluate its performance.

Choose an appropriate evaluation metric. Depending on the problem trying to solve, use accuracy, precision, recall, F1 score, or another metric.

Train the model using the training data. This may involve selecting hyper-parameters, such as the learning rate or the number of hidden layers and tuning them to achieve the best performance.

Evaluate the model on the testing data using the evaluation metric chosen in step 3. This gives a sense of how the model is performing.

A different algorithm or changes the existing model, such as adding more hidden layers or changing the activation function.

Repeat the process until satisfied with the performance of the model.

It is essential to remember that optimizing an ML model is often an iterative process, and it may take several rounds of experimentation to arrive at a final model that meets your requirements as per below. Figure 3.9 may describe a suitable methodology.

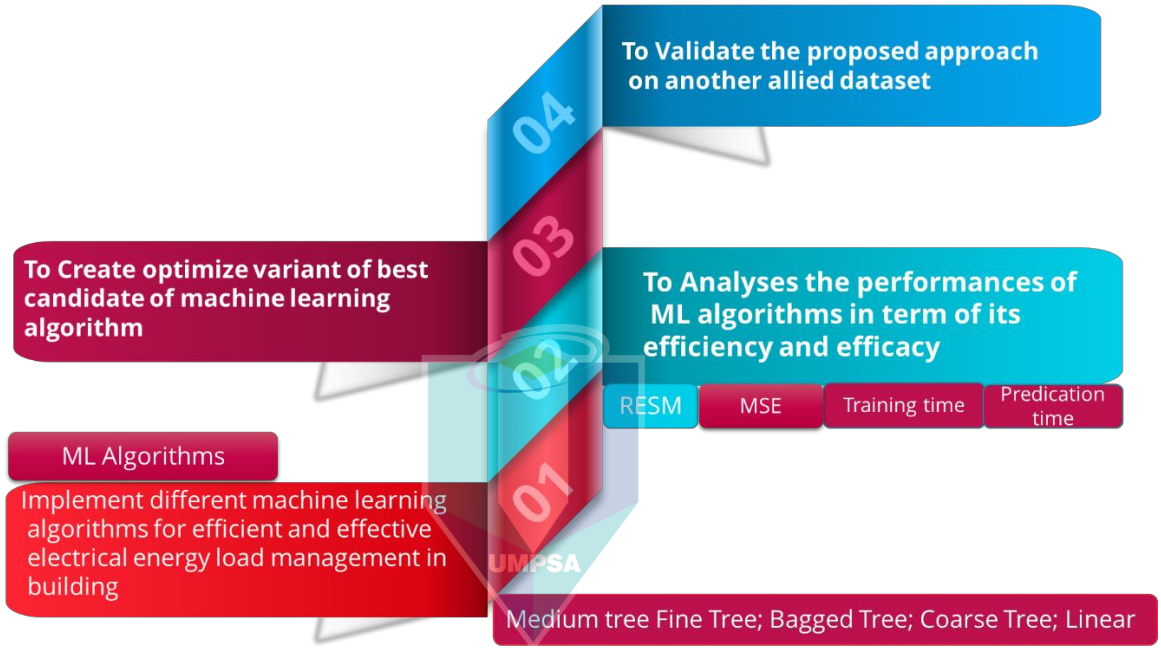


Figure 3.9 Flow of proposed optimized ML model.

3.14 Parameters for optimization in ML

Building a decision tree that accurately represents the underlying data can be a challenging task. To address this, there are various steps and parameters that can be optimized to improve the performance of the decision tree model. These steps involve preparing the data, splitting it into training and testing sets, building the decision tree using an appropriate algorithm, tuning the hyper parameters, cross validating the model, and evaluating its performance using suitable metrics. By following these steps and optimizing the relevant parameters, decision tree was creating models that better represent the data and provide more accurate prediction.

Data preparation: The first step is to prepare the data by cleaning, pre-processing, and feature engineering. This involves dealing with missing values, handling

categorical variables, scaling, and transforming the data to make it suitable for decision tree models.

Splitting the data: Once the data is prepared, it is split into training and testing sets. The training set is used to build the decision tree model, while the testing set is used to evaluate the model's performance.

Building the decision tree: The next step is to build the decision tree using an appropriate algorithm. There are different algorithms for building decision trees such as ID3, C4.5, CART, and Random Forest. The algorithm selected can be based on the data, performance metrics, and requirements of the problem.

Tuning the hyper parameters: Decision trees have hyper parameters that can be tuned to optimize the model's performance. Some of the hyper parameters that can be tuned include the maximum depth of the tree, minimum number of samples required to split an internal node, criterion for splitting, and maximum number of leaf nodes.

Cross-validation: Cross-validation is used to evaluate the performance of the decision tree model and to choose the best set of hyper parameters. It involves splitting the training data into several folds and testing the model on each fold while using the other folds for training.

Evaluating the performance: Finally, the performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics help to determine whether the model is overfitting, under fitting, or performing optimally.

3.15 Optimization hyper parameters

Hyper parameter optimization is a crucial step in training a machine learning model, as it involves tuning various settings that control the learning process. The optimal values of hyperparameters can significantly impact the model's performance. Here are some common hyper parameters that are often optimized for different types of machine learning models.

Maximum number of splits – The software searches among integers log-scaled in the range $[1,\max(2,n-1)]$, where n is the number of observations.

Split criterion – The software searches among Gini's diversity index, Towing rule, and Maximum deviance reduction.

3.15.1 Additional Hypermeter

Some additional hyperparameters that are commonly found in various machine learning models are

- **Surrogate decision splits**

Surrogate decision splits, a crucial aspect of medical ethics, present complex challenges for individuals who find themselves designated as substitute decision-makers for incapacitated patients. When a patient is unable to express their wishes due to unconsciousness, cognitive impairment, or other reasons, the responsibility of making critical medical decisions falls on surrogate decision-makers, often leading to dilemmas and uncertainties. These splits arise when the surrogate faces difficult choices with unclear guidance from the patient, conflicting opinions within the family or medical team, intricate medical conditions, ethical concerns, and emotional burdens. As a central theme in medical ethics and end-of-life care, understanding how surrogate decision splits are navigated is of utmost importance in ensuring patient autonomy, respecting their values, and arriving at decisions that genuinely align with their best interests.

- **Maximum surrogates per node**

In the context of distributed computing and parallel processing, the term "surrogate" is not commonly used to refer to nodes or processors. Instead, "surrogate" usually pertains to an entity that represents or acts on behalf of another in certain distributed computing models or algorithms. In distributed computing, the maximum number of surrogates (also referred to as "proxies" or "agents") per node typically depends on the specific architecture and algorithm being used. The number of surrogates per node is often limited by factors such as memory capacity, processing power, and the communication overhead involved in managing and coordinating surrogates.

3.15.2 Decision Trees

Decision trees are easy to interpret, fast for fitting and prediction, and low on memory usage, but they can have low predictive accuracy. Try to grow simpler trees to prevent overfitting. Control the depth with the **Maximum number of splits** setting.

Table 3.3 Model Flexibility table

Classifier Type	Interpretability	Model Flexibility
Medium Tree 	Easy	Medium Medium number of leaves for finer distinctions between classes (maximum number of splits is 20).
Fine Tree 	Easy	High Many leaves to make many fine distinctions between classes (maximum number of splits is 100).

3.15.3 Tree Model Hyper Parameter Options

Classification trees in Classification Learner use the fitctree function.

Maximum number of splits

Specify the maximum number of splits or branch points to control the depth of your tree. When grow a decision tree, consider its simplicity and predictive power. To change the number of splits, click the buttons or enter a positive integer value in the **Maximum number of splits** box.

A fine tree with many leaves is usually highly accurate on the training data. However, the tree might not show comparable accuracy on an independent test set. A leafy tree tends to overtrain, and its validation accuracy is often far lower than its training (or resubstituting) accuracy.

A coarse tree does not attain high training accuracy. But a coarse tree can be more robust because its training accuracy can approach that of a representative test set. Also, a coarse tree is easy to interpret.

3.15.4 Split criterion

Specify the split criterion measure for deciding when to split nodes. Try each of the three settings to see if they improve the model with your data.

Split criterion options are Gini's diversity index, Twoing rule, or Maximum deviance reduction (also known as cross entropy).

The classification tree tries to optimize to pure nodes containing only one class. Gini's diversity index (the default) and the deviance criterion measure node impurity. The twoing rule is a different measure for deciding how to split a node, where maximizing the twoing rule expression increases node purity.

Surrogate decision splits — Only for missing data.

Specify surrogate use for decision splits. If have data with missing values, use surrogate splits to improve the accuracy of predictions.

Set **Surrogate decision splits** to On, the classification tree finds 10 surrogate splits at each branch node. To change the number, click the buttons or enter a positive integer value in the **Maximum surrogates per node** box.

Set **Surrogate decision splits** to Find All, the classification tree finds all surrogate splits at each branch node. The Find All setting can use considerable time and memory.

Classifier Type	Interpretability	Ensemble Method	Model Flexibility
Bagged Trees 	Hard	Random forest Bag, with Decision Tree learners	High — increases with **Number of learners** setting.

Ensemble classifiers in Classification Learner use the fitcensemble function. Set these options:

For help choosing **Ensemble method** and **Learner type**, see the Ensemble table. Try the presets first.

3.15.5 Maximum number of splits

For boosting ensemble methods, specify the maximum number of splits or branch points to control the depth of your tree learners. Many branches tend to overfit, and simpler trees can be more robust and easier to interpret. Experiment to choose the best tree depth for the trees in the ensemble.

3.16 Number of learners

Many learners can produce high accuracy but can be time consuming to fit. Start with a few dozen learners, and then inspect the performance. An ensemble with good predictive power can need a few hundred learners.

3.17 Summary

A Penta-folded cascading methodology was used in this study.

In the first phase, the SEIL dataset is used. In the second phase, the training and testing samples with a random permutation. The training set is used to train 24 machine-learning algorithms. In the third phase, the parametric performance of each ML algorithm

The same parameters for the testing phase are computed. The fourth phase is the ranking of algorithms based on their efficacy and efficiency are established. Finally, the fifth phase is enhancing the results by optimizing variant algorithms on the best-ranked algorithms regarding efficacy and efficiency.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the experimental results, analysis and discussions of the Machine learning algorithm. The results presented started with elements and phase analysis data attempting to determine the best ML algorithm based on empirical fact. The systematic and empirical evaluation of a wide range of machine learning algorithms reveals that the Bagged Trees, Fine Trees, and Medium Trees algorithms are the top three ranked algorithms for energy demand forecasting using the SEIL dataset. This finding presents a knowledge add-on to the SEIL project consisting of the recommendation of the best machine learning algorithm for energy demand forecasting.

In this section, an analysis of simulation results was conducted to determine the best machine learning algorithm for energy demand forecasting. Previous literature has provided little information on identifying the best algorithm empirically. The study evaluated a wide range of machine learning algorithms and found that Bagged Trees, Fine Trees, and Medium Trees are the top three ranked algorithms for energy demand forecasting using the SEIL dataset. This finding adds to the knowledge of the SEIL project by recommending the best machine learning algorithm for energy demand forecasting. Additionally, a new and customized algorithm is suggested for further improvements in efficiency and efficacy. The study suggests that the customized Medium Trees algorithm is recommended for efficiency, while the customized Bagged Trees method is recommended for higher-order efficacy. These conclusions are based solely on empirical data and graphical facts in the study. The study shows that the performance of a load management system depends on its efficiency and effectiveness, and selecting the optimum trade-off between the two is crucial. Machine learning algorithms are reported to be the best candidates for load management and demand forecasting but selecting the relevant algorithm(s) for a specific application is essential for higher performance. This study contributes to extending the research on the SEIL dataset by proposing the best candidate machine learning algorithm for more performance, supported by the empirical performance parameters of machine learning algorithms. For simulation and testing MATLAB software used, all figures and tables produced by MATLAB software.

4.2 Best candidate of machine algorithm for energy demand prediction

In this section, a thorough empirical evaluation was conducted to identify the best machine learning algorithm for energy demand prediction using the SEIL dataset. The visual inferences of Table 4.1 are presented in Figures 4.1 - 4.8 for easy reference (with the actual table in Appendix A). The predictive vs. actual and residual graphs for each algorithm are illustrated in Figures 4.7 - 4.17. This study evaluated 24 machine learning prediction algorithms based on benchmark performance parameters. The predicted vs. actual graph shows the true response on the x-axis and the predicted response on the y-axis. The black line represents the approximate linearity of the curves, while the blue dots depict the actual observations. The variation between the predicted and actual values indicates the algorithm's prediction performance, with larger variation corresponding to poorer performance. Table 4.2 shows the degree and measure of variation for each algorithm, which is a function of RMSE, R-Squared, MSE, and MAE for both training and testing events that having less per centage of error then previous work done by RoSe et al. (2023). Efficiency is also established based on prediction speed and computation time. Higher error measure values correspond to poorer algorithm performance. Figures 4.17 and 4.19 also include the residual error for training and testing of the top three performing algorithms, with the predicted response on the x-axis and the residual error on the y-axis.

The efficacy of the candidate algorithm is indicated by the proximity of residual error to the predicted observation, which is also supported by the empirical and absolute values in Table 3. The top three performing machine learning algorithms for energy demand prediction at a university campus based on SEIL datasets were selected, and a detailed investigation of their performance parameters was conducted. Graphical illustrations and empirical findings have revealed that Bagged Trees (1), Fine Trees (2), and Medium Trees (3) are the top three performing algorithms in terms of efficacy. However, a reverse ranking was observed in terms of efficiency, which can also be inferred from Table 4.1. The performance measures, such as RMSE, R-Squared, MSE, and MAE, indicate the algorithm's efficacy, while prediction speed and training time reflect its efficiency. and MAE, indicate the algorithm's efficacy, while prediction speed and training time reflect its efficiency.

Table 4.1 Training and testing table

Algorithm Name	Training						Testing			
	RMSE	R-Squared	MSE	MAE	Prediction speed (obs/sec)	Training time	RMSE	R-Squared	MSE	MAE
Fine Tree	2.11E+06	1	4.44E+12	1.40E+06	3000000	24.148	2.38E+06	1	5.64E+12	1.41E+06
Linear	1.35E+08	0.54	1.83E+16	8.42E+07	790000	22.065	1.36E+08	0.53	1.84E+16	8.45E+07
Interactions Linear	1.22E+08	0.62	1.48E+16	8.02E+07	110000	65.108	1.23E+08	0.62	1.50E+16	8.05E+07
Robust Linear	1.64E+08	0.32	2.68E+16	5.25E+07	790000	19.44	1.65E+08	0.31	2.71E+16	5.30E+07
Stepwise Linear	1.18E+08	0.65	1.39E+16	7.63E+07	600000	24417	1.18E+08	0.64	1.40E+16	7.66E+07
Fine Tree	2.11E+06	1	4.44E+12	1.40E+06	3300000	8.3244	2.38E+06	1	5.64E+12	1.41E+06
Medium Tree	2.81E+06	1	7.87E+12	1.42E+06	3700000	8.0721	3.20E+06	1	1.02E+13	1.43E+06
Coarse Tree	4.26E+06	1	1.81E+13	1.49E+06	4000000	7.5477	4.63E+06	1	2.15E+13	1.50E+06
Linear SVM	8.67E+08	18.13	7.52E+17	6.69E+08	1400000	8892.6	8.68E+08	18.13	7.54E+17	6.69E+08
Quadratic SVM	3.46E+08	2.05	1.20E+17	2.98E+08	240000	18985	3.45E+08	2.02	1.19E+17	2.98E+08
Cubic SVM	6.38E+08	9.35	4.07E+17	5.50E+08	260000	5761.7	6.37E+08	9.31	4.06E+17	5.50E+08
Fine Gaussian SVM	1.04E+08	0.72	1.08E+16	8.83E+07	270000	10099	1.04E+08	0.72	1.08E+16	8.85E+07
Medium Gaussian SVM	2.10E+08	0.13	4.43E+16	1.80E+08	1200000	17835	2.11E+08	0.13	4.44E+16	1.80E+08

Table 4.1 Continued

Algorithm Name	Training						Testing			
	RMSE	R-Squared	MSE	MAE	Prediction speed (obs/sec)	Training time	RMSE	R-Squared	MSE	MAE
Boosted Trees	2.33E+07	0.99	5.42E+14	1.66E+07	180000	62.082	2.31E+07	0.99	5.34E+14	1.66E+07
Bagged Trees	1.58E+06	1	2.48E+12	1.06E+06	120000	119.87	1.78E+06	1	3.17E+12	1.09E+06
Squared Exponential GPR	7.68E+07	0.85	5.89E+15	4.62E+07	200	9102.8	7.66E+07	0.85	5.87E+15	4.62E+07
Matern 5/2 GPR	6.40E+07	0.9	4.10E+15	3.87E+07	110	15589	6.40E+07	0.9	4.10E+15	3.88E+07
Exponential GPR	6.86E+07	0.88	4.70E+15	3.73E+07	130	14212	6.87E+07	0.88	4.72E+15	3.73E+07
Rational Quadratic GPR	7.30E+07	0.86	5.33E+15	4.11E+07	110	15637	7.28E+07	0.87	5.29E+15	4.11E+07
Narrow Neural Network	3.18E+07	0.97	1.01E+15	1.04E+07	1000000	254.45	3.12E+07	0.98	9.76E+14	1.03E+07
Medium Neural Network	2.54E+07	0.98	6.46E+14	1.46E+07	1100000	396.17	2.51E+07	0.98	6.32E+14	1.46E+07
Wide Neural Network	1.81E+07	0.99	3.27E+14	1.14E+07	630000	1238.8	1.79E+07	0.99	3.19E+14	1.14E+07

Table 4.1 Continued

Algorithm Name	Training						Testing			
	RMSE	R-Squared	MSE	MAE	Prediction speed (obs/sec)	Training time	RMSE	R-Squared	MSE	MAE
Bilayered Neural Network	3.77E+08	2.62	1.42E+17	3.21E+08	1000000	35.635	3.77E+08	2.6	1.42E+17	3.20E+08
Trilayered Neural Network	3.77E+08	2.62	1.42E+17	3.21E+08	870000	48.817	3.77E+08	2.6	1.42E+17	3.20E+08
Coarse Gaussian SVM	2.17E+08	0.19	4.69E+16	1.82E+08	1300000	18127	2.17E+08	0.2	4.71E+16	1.82E+08

Table 4.2 Training and testing of the different algorithms with the result (B)

Algorithm	RMSE	R-Squared	MSE	MAE
Linear	$1.36 \times 10^{+08}$	0.53	$1.84 \times 10^{+16}$	$8.45 \times 10^{+07}$
Interactions Linear	$1.23 \times 10^{+08}$	0.62	$1.50 \times 10^{+16}$	$8.05 \times 10^{+07}$
Robust Linear	$1.65 \times 10^{+08}$	0.31	$2.71 \times 10^{+16}$	$5.30 \times 10^{+07}$
Stepwise Linear	$1.18 \times 10^{+08}$	0.64	$1.40 \times 10^{+16}$	$7.66 \times 10^{+07}$
Fine Trees	$2.38 \times 10^{+06}$	1	$5.64 \times 10^{+12}$	$1.41 \times 10^{+06}$
Medium Trees	$3.20 \times 10^{+06}$	1	$1.02 \times 10^{+13}$	$1.43 \times 10^{+06}$
Coarse Tree	$4.63 \times 10^{+06}$	1	$2.15 \times 10^{+13}$	$1.50 \times 10^{+06}$
Linear SVM	$8.68 \times 10^{+08}$	−18.13	$7.54 \times 10^{+17}$	$6.69 \times 10^{+08}$
Quadratic SVM	$3.45 \times 10^{+08}$	−2.02	$1.19 \times 10^{+17}$	$2.98 \times 10^{+08}$
Cubic SVM	$6.37 \times 10^{+08}$	−9.31	$4.06 \times 10^{+17}$	$5.50 \times 10^{+08}$
Fine Gaussian SVM	$1.04 \times 10^{+08}$	0.72	$1.08 \times 10^{+16}$	$8.85 \times 10^{+07}$
Medium Gaussian SVM	$2.11 \times 10^{+08}$	−0.13	$4.44 \times 10^{+16}$	$1.80 \times 10^{+08}$
Coarse Gaussian SVM	$2.17 \times 10^{+08}$	−0.2	$4.71 \times 10^{+16}$	$1.82 \times 10^{+08}$
Boosted Trees	$2.31 \times 10^{+07}$	0.99	$5.34 \times 10^{+14}$	$1.66 \times 10^{+07}$
Bagged Trees	$1.78 \times 10^{+06}$	1	$3.17 \times 10^{+12}$	$1.09 \times 10^{+06}$
Squared Exponential GPR	$7.66 \times 10^{+07}$	0.85	$5.87 \times 10^{+15}$	$4.62 \times 10^{+07}$
Matern 5/2 GPR	$6.40 \times 10^{+07}$	0.9	$4.10 \times 10^{+15}$	$3.88 \times 10^{+07}$
Exponential GPR	$6.87 \times 10^{+07}$	0.88	$4.72 \times 10^{+15}$	$3.73 \times 10^{+07}$
Rational Quadratic GPR	$7.28 \times 10^{+07}$	0.87	$5.29 \times 10^{+15}$	$4.11 \times 10^{+07}$
Narrow Neural Network	$3.12 \times 10^{+07}$	0.98	$9.76 \times 10^{+14}$	$1.03 \times 10^{+07}$
Medium Neural Network	$2.51 \times 10^{+07}$	0.98	$6.32 \times 10^{+14}$	$1.46 \times 10^{+07}$
Wide Neural Network	$1.79 \times 10^{+07}$	0.99	$3.19 \times 10^{+14}$	$1.14 \times 10^{+07}$
Bi-layered Neural Network	$3.77 \times 10^{+08}$	−2.6	$1.42 \times 10^{+17}$	$3.20 \times 10^{+08}$
Tri-layered Neural Network	$3.77 \times 10^{+08}$	−2.6	$1.42 \times 10^{+17}$	$3.20 \times 10^{+08}$

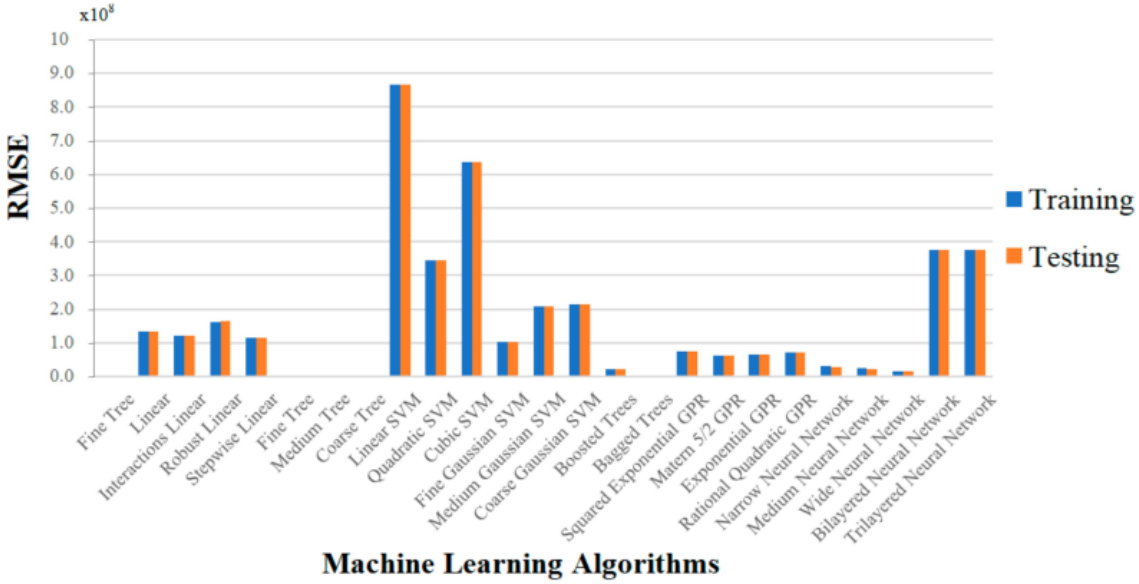


Figure 4.1 Training and testing RMSE

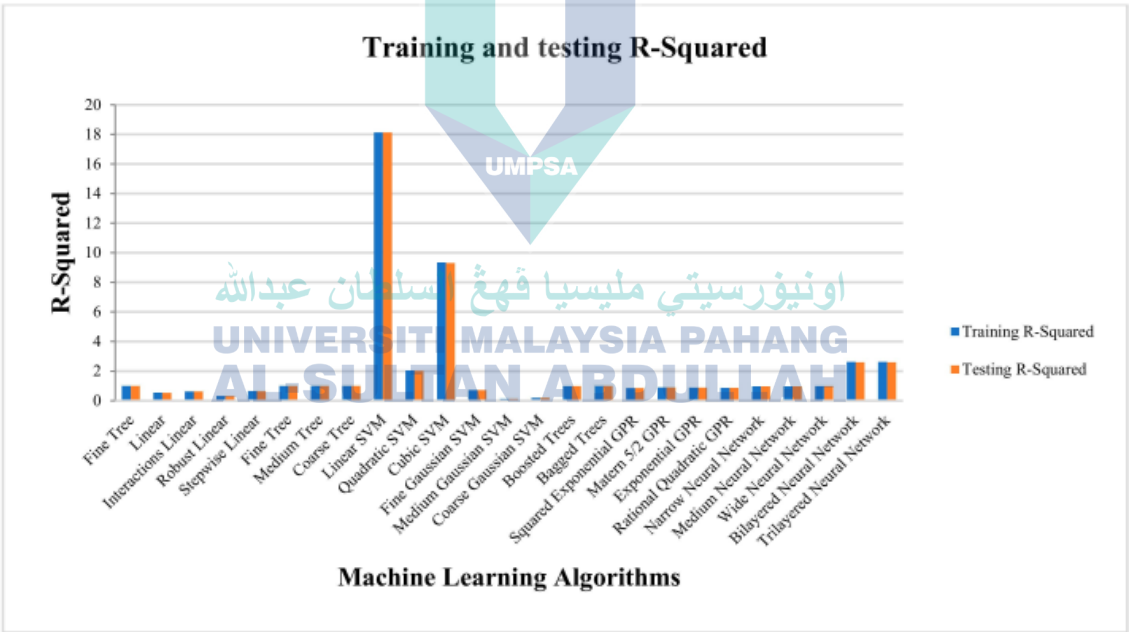


Figure 4.2 Training and Testing R-Squared

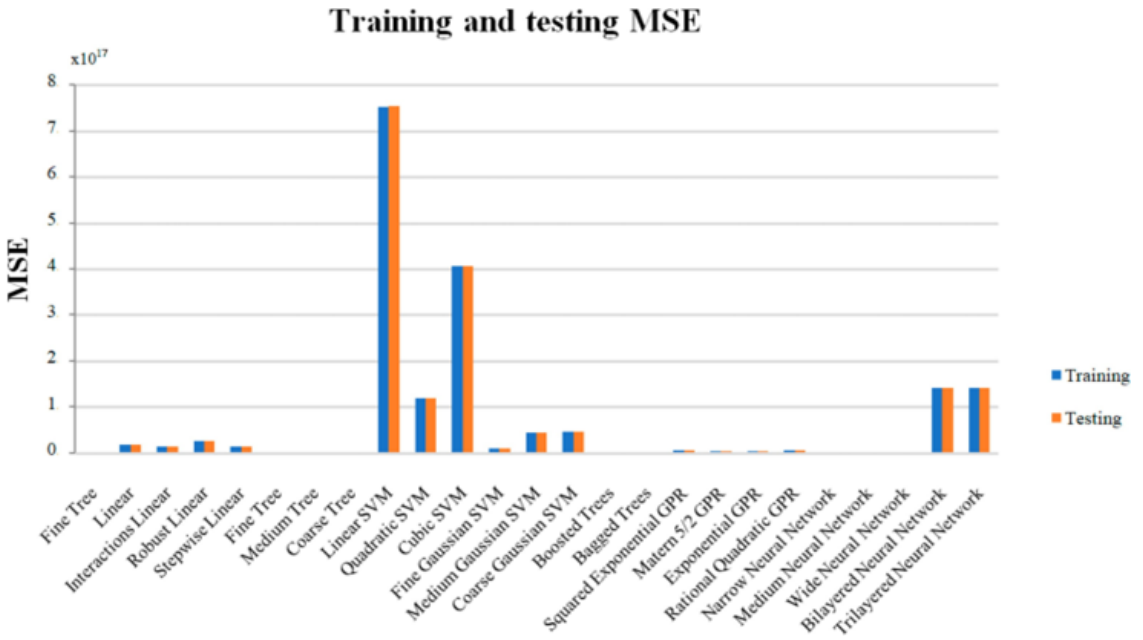


Figure 4.3 Training and testing MSE

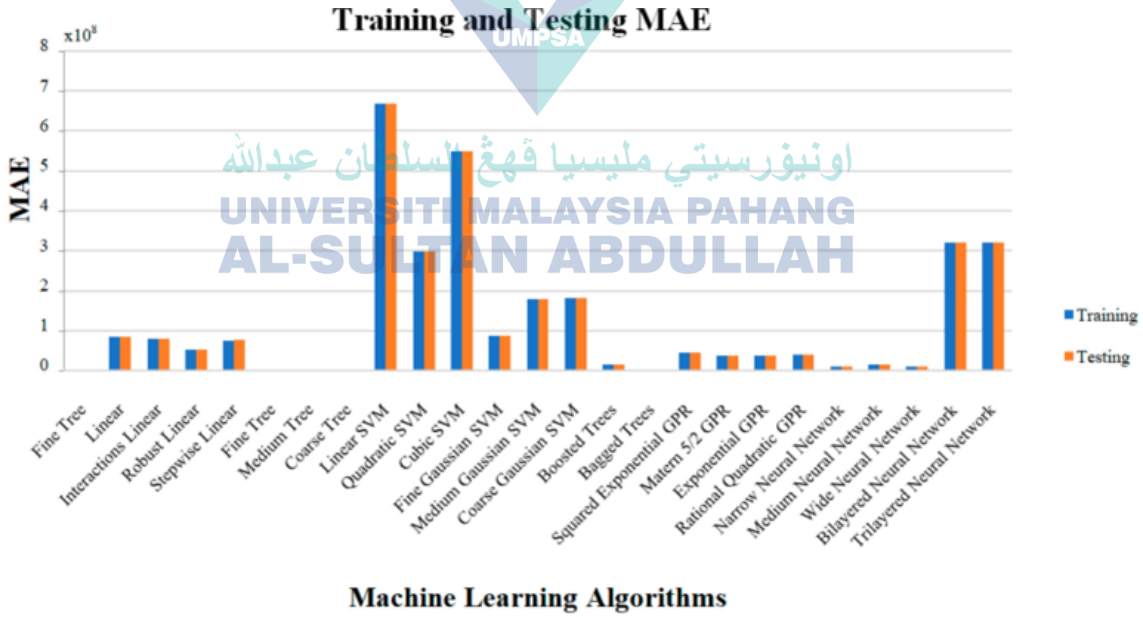


Figure 4.4 Training and testing MAE

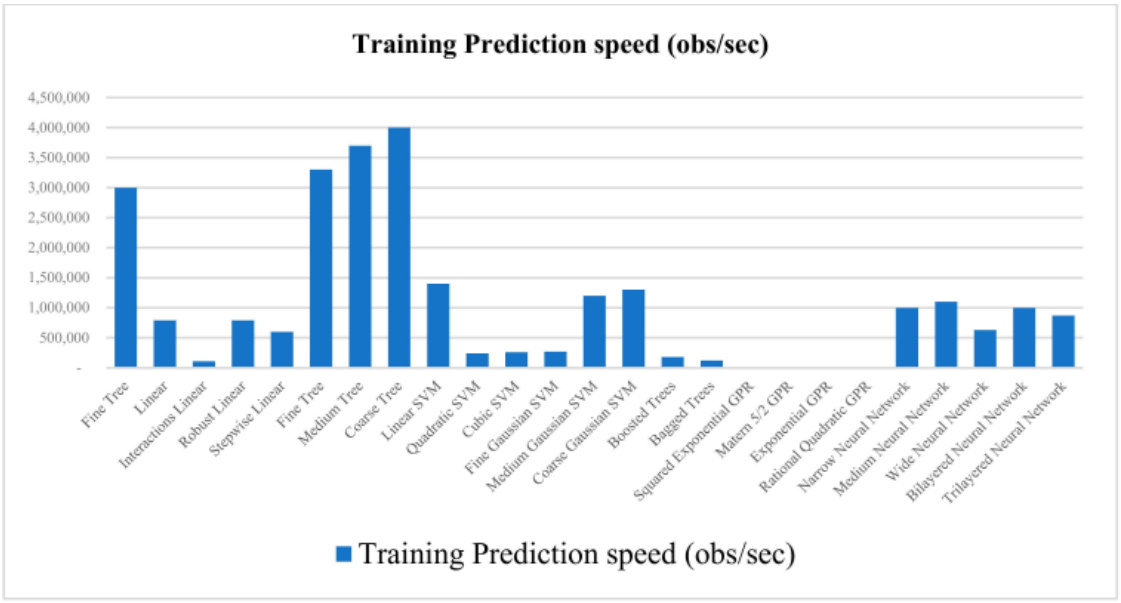


Figure 4.5 Prediction speed

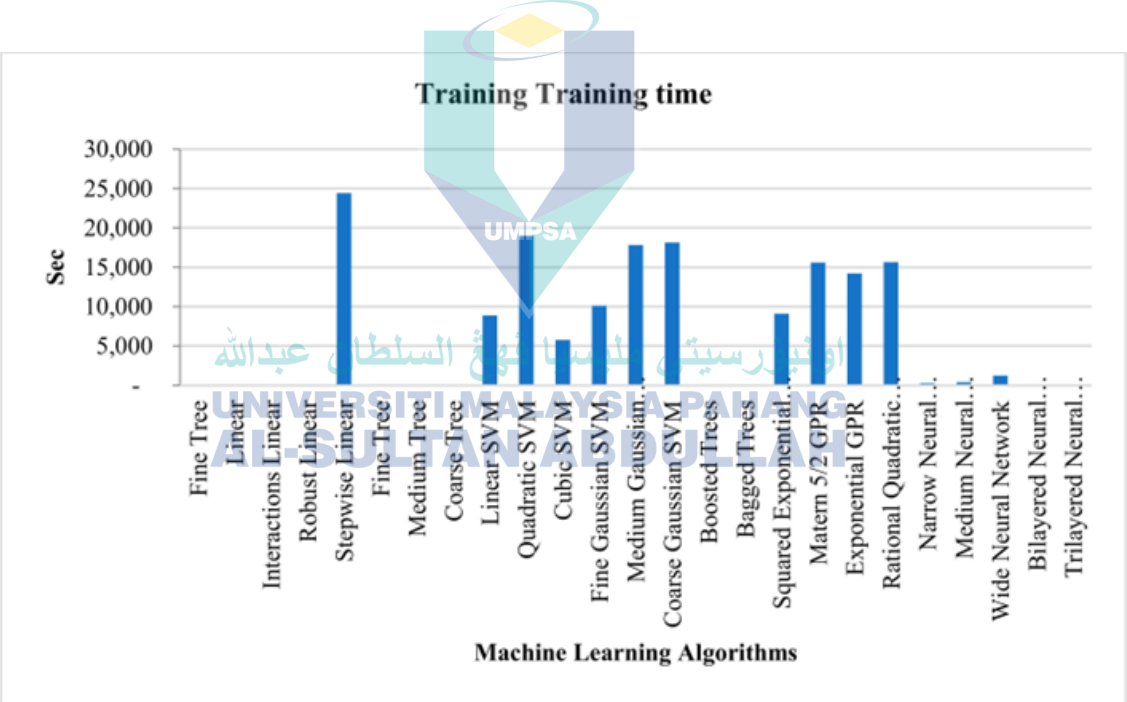


Figure 4.6 Training time

In the context of energy consumption prediction, certain algorithms, notably Fine Trees and Bagged Trees, have demonstrated outstanding performance with low RMSE and high R-Squared, indicating their proficiency in making accurate predictions. Conversely, algorithms like Linear SVM and Tri-layered Neural Network exhibited relatively higher RMSE and negative R-Squared, suggesting that they may not be the optimal choices for this particular task.

An appropriate algorithm plays a pivotal role in achieving precise and efficient results in energy consumption prediction. By analyzing the provided performance metrics, further examination and comparison can be undertaken to identify the most suitable algorithm tailored to meet the specific requirements and objectives of the energy management project, Here's a summary of the performance metrics for the algorithms:

Linear:

RMSE: 1.36×10^{8}

R-Squared: 0.53

MSE: 1.84×10^{16}

MAE: 8.45×10^{7}

Interactions Linear:

RMSE: 1.23×1^{8}

R-Squared: 0.62

MSE: 1.50×10^{16}

MAE: 8.05×10^{7}

Robust Linear:

RMSE: 1.65×10^{8}

R-Squared: 0.31

MSE: 2.71×10^{16}

MAE: 5.30 x 10^7

Stepwise Linear:

RMSE: 1.18 x 10^8

R-Squared: 0.64

MSE: 1.40 x 10^{16}

MAE: 7.66 x 10^7

Fine Trees:

RMSE: 2.38 x 10^6

R-Squared: 1

MSE: 5.64 x 10^{12}

MAE: 1.41 x 10^6

Figure 4.1 shows bar graph based on the RMSE values of different algorithms for predicting energy consumption, the algorithm with the lowest RMSE value is "Bagged Trees" for the testing phase:

Bagged Trees:

Testing RMSE: 1.78E+06 (1.78 million)

This indicates that the Bagged Trees algorithm is the best performer among the evaluated models for energy consumption prediction. It has the lowest RMSE on the testing dataset, suggesting that it provides the most accurate predictions when dealing with unseen data (RoSe et al. 2023).

Comparatively, other algorithms may have higher RMSE values on the testing dataset, which means they are less accurate in their predictions compared to Bagged Trees.

It's important to note that while Bagged Trees performs the best based on the provided RMSE values, other factors such as model complexity, computational efficiency, and interpretability should also be considered when selecting the most suitable algorithm for practical applications.

Based on the RMSE values, Bagged Trees is the best-performing algorithm for energy consumption prediction among the evaluated models. However, further analysis and considerations are necessary to make a well-informed decision for its deployment in real-world energy management projects.

R-squared is a statistical metric that measures how well a regression model fits the data. It indicates the proportion of variance in the dependent variable it explained by a bar graph in Figure 4.2 (energy consumption in this case) that is explained by the independent variables (features) used in the model. R-squared ranges from 0 to 1, where 0 means the model explains none of the variance, and 1 means it explains all the variances.

Higher R-squared values indicate that the model's predictions align closely with the actual values, suggesting a better fit to the data.

Let's compare and analyse the R-squared values for different algorithms:

Bagged Trees:

Training R^2: 1 (100%)

Testing R^2: 1 (100%)

Bagged Trees have the highest R-squared values for both the training and testing phases, indicating that the model explains all the variance in energy consumption in both datasets. This suggests that Bagged Trees provide an excellent fit to the data and accurately predict energy consumption.

Fine Trees:

Training R^2: 1 (100%)

Testing R^2: (99%)

Fine Trees also have a perfect R-squared value for the training phase, indicating an excellent fit to the training data.

Linear:

Training R^2: 0.54 (54%)

Testing R^2: 0.53 (53%)

Linear model's R-squared values are relatively lower compared to Bagged Trees, indicating that the model explains only around 54% of the variance in the training data and 53% in the testing data. This suggests that the linear model might not be capturing all the underlying patterns in the data as effectively as Bagged Trees.

Interactions Linear:

Training R^2: 0.62 (62%)

Testing R^2: 0.62 (62%)

Interactions Linear shows slightly higher R-squared values than the Linear model, but still lower than Bagged Trees. It explains around 62% of the variance in both the training and testing datasets.

Robust Linear:

Training R^2: 0.32 (32%)

Testing R^2: 0.31 (31%)

Robust Linear has lower R-squared values compared to the previous models, indicating that it explains only about 32% of the variance in the training data and 31% in the testing data. This suggests that the model might not be capturing the underlying patterns in the data well.

Coarse Tree:

Training R^2: 1 (100%)

Testing R^2: 1 (100%)

Coarse Tree has perfect R-squared values for both the training and testing phases, suggesting an excellent fit to the data and accurate predictions.

From the provided R-squared values, Bagged Trees and Coarse Tree models stand out with perfect R-squared values for both training and testing, indicating accurate and robust predictions. Linear models, on the other hand, have lower R-squared values, indicating less effective performance in explaining the variance in the data.

Bagged Trees and Coarse Tree models demonstrate superior performance in explaining the variance in energy consumption and providing accurate predictions. These models could be considered the best performers among the evaluated algorithms for energy consumption prediction (Reddy et al. 2023).

Figure 4.3 explained MSE relation with the help of bar graph, Bagged Trees and Fine Trees appear to be the best-performing algorithms based on the MSE values, as they demonstrate lower prediction errors compared to the other models. These findings have implications for the selection of suitable algorithms for energy consumption prediction in real-world applications. Further analysis will be conducted to explore the strengths and weaknesses of these algorithms and consider additional factors like computational efficiency and interpretability to make an informed decision for practical energy management projects.

Figure 4.4 From the MAE values, Bagged Trees and Fine Trees have relatively low training and testing MAE, indicating better performance in predicting energy consumption with smaller errors compared to other algorithms. Linear and Interactions Linear models show higher training and testing MAE, suggesting they may not be as effective in capturing the underlying patterns in the data. Linear SVM, on the other hand, exhibits extremely high MAE values, indicating significant discrepancies between predicted and actual energy consumption.

Bagged Trees and Fine Trees appear to be the best-performing algorithms based on the MAE values, as they demonstrate lower prediction errors compared to the other models. These findings are essential in the selection of suitable algorithms for energy consumption prediction in real-world applications.

From the training prediction speed values Figure 4.5 and 4.6. Fine Trees have the highest speed, being able to process 3 million observations per second. Coarse Tree also demonstrates high training prediction speed, processing 4 million observations per second. On the other hand, Linear SVM has a relatively slower training prediction speed, processing 1.4 million observations per second.

Fine Trees and Coarse Tree models stand out with higher training prediction speeds, indicating faster processing capabilities compared to other algorithms. All value calculated based on table 4.1 and result calculated based on table 4.2.

4.2.1 Predicted vs Actual results

In the described graph Figure 4.7 and Figure 4.8, a model that is performing well have most of its data points following a clear linear trend along the perfect fit line. Approximately 90% of the data points will fall within a narrow band around the perfect fit line, indicating that the predictions are very close to the actual values.

On the other hand, around 10% of the data points might deviate from the perfect fit line, going slightly above or below it. These deviations represent the model's prediction errors, which are inevitable in any real-world predictive modelling task.

Most data points are scattered around the perfect fit line, and only a small percentage deviate slightly, it suggests that the model is making accurate predictions and capturing the underlying patterns in the data effectively. This is a desirable outcome as it indicates a strong and reliable predictive model.

Figures 4.9 and 4.10 explain relation between predicted and residuals of training and testing most of the data points cluster around the y-axis (Residual = 0), indicating that the model's predictions are accurate, and the errors are centred around zero. This implies that the model is capturing the underlying patterns in the data and making reliable predictions for the training dataset and this result is supported by Akhtar, Sujod, and Rizvi (2022).

Around 10% of the data points deviate slightly from the y-axis, either going above or below. These deviations represent the model's prediction errors, which expected in any real-world predictive modelling task. A well-performing model have a small

percentage of deviations, and these errors are typically random and not indicative of any systematic bias.

A Residual Training graph with most data points close to the y-axis (Residual = 0) and around 10% of the data points slightly deviating above and below indicates a well-fitted regression model. It suggests that the model is accurately predicting the target variable for the training dataset, with minimal systematic bias and consistent error distribution. However, it's important to note that while most of the data points should be close to the y-axis, there may still be some variability in the distribution of the Residuals. This is normal and can be influenced by factors such as the complexity of the data or the nature of the problem being modelled. If the general trend shows most Residuals near the y-axis and approximately 10% deviating above and below, it is an encouraging sign of a reliable regression model.

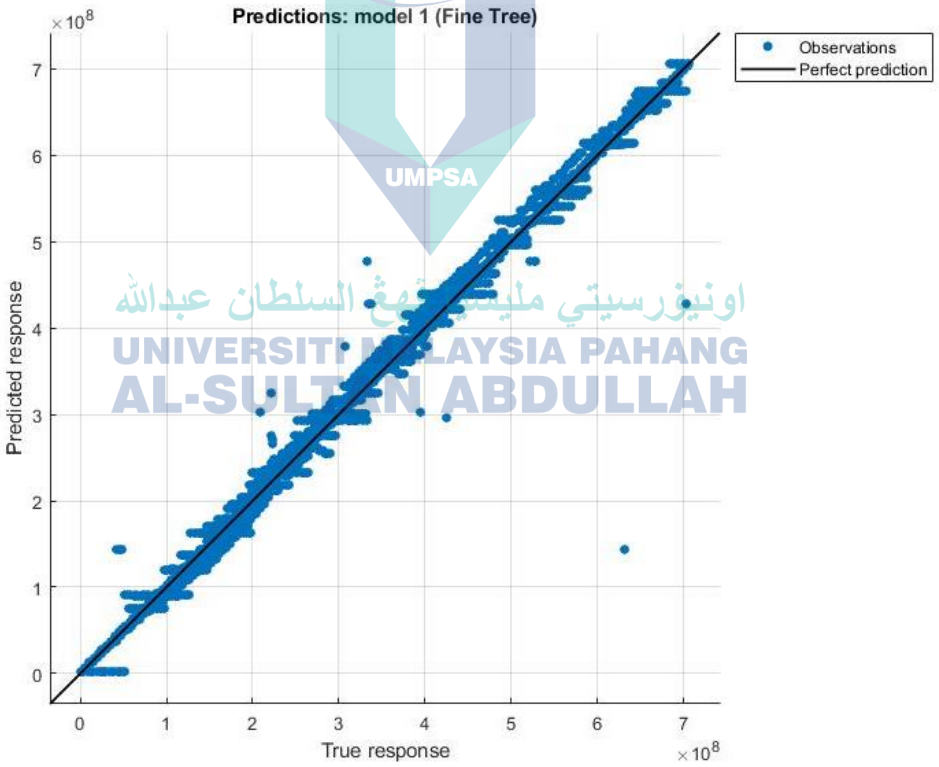


Figure 4.7 Fine Trees Prediction vs. Actual training

In Figure 4.7 the performance of the machine learning model is visually evaluated based on its ability to predict target values against observed values. Starting with Figure 4.7, a model demonstrating strong performance exhibits most of its data points adhering closely to a linear trend along the ideal fit line. Around 90% of the data points are

expected to fall within a narrow band around this perfect fit line, indicating consistent and accurate predictions. However, it's important to note that approximately 10% of the data points might deviate slightly from the ideal fit line, representing prediction errors inherent in real-world scenarios.

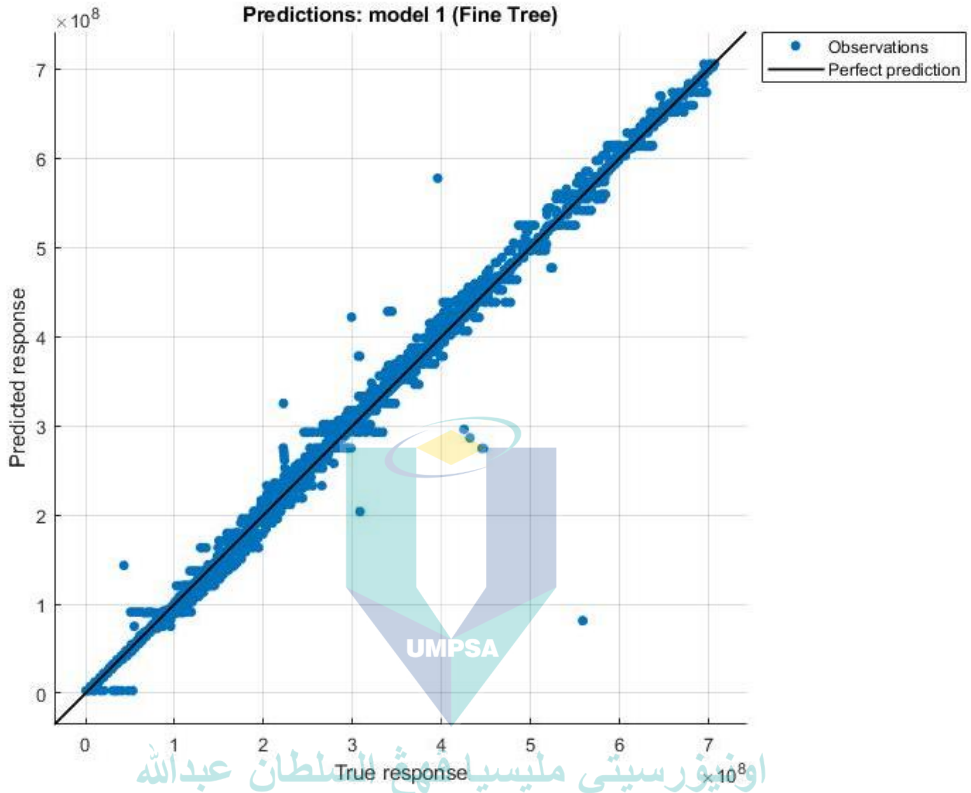


Figure 4.8 Fine Trees Prediction vs. Actual testing

Moving to Figure 4.8, the distribution of data points around the perfect fit line further illustrates the model's ability to capture underlying patterns effectively. Despite the presence of some deviations, the bulk of the data points remain closely aligned with the ideal fit line, emphasizing the model's overall accuracy and reliability. Figures 4.7 and 4.8 collectively demonstrate the proficiency of the model in making accurate predictions and capturing underlying data patterns. These observations highlight the development of a robust and dependable predictive model, essential for successful applications in various domains. The alignment of most data points along clear linear trends, coupled with the tight clustering around the ideal fit line, signifies the model's precision and consistency. While minor deviations exist, they do not detract from the model's overall efficacy, highlighting its robustness and reliability in real-world predictive modelling scenarios.

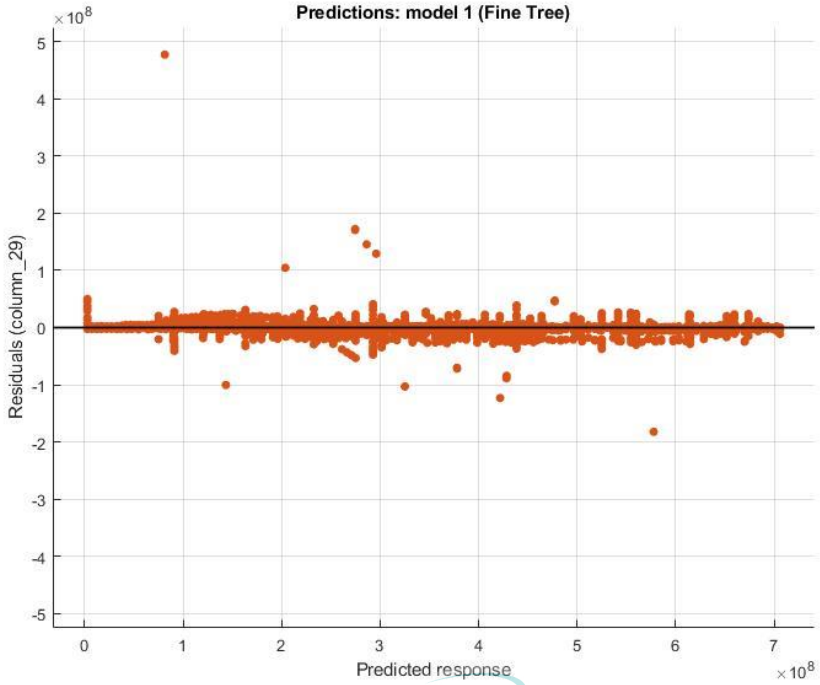


Figure 4.9 Fine Trees Residual training.

In Figure 4.9, we are presented with a visualization of the residuals generated by a tree-based machine learning model during the training phase. Residuals represent the differences between the observed target values and the predicted values generated by the model. These residuals provide valuable insights into the performance and behaviour of the model during training. In this visualization, the residuals are plotted against the predicted values generated by the model. Each data point on the graph represents an individual data instance from the training dataset. The position of a data point relative to the x-axis (predicted values) indicates whether the model under- or over-predicted the target variable for that instance. Observing the distribution of data points in Figure 4.9, note that there is a clear pattern emerging. Specifically, observe that there are five data points located above the x-axis (positive residuals) and six data points located below the x-axis (negative residuals). When data points lie above the x-axis, it indicates that the model has under-predicted the target variable for those instances. Conversely, when data points lie below the x-axis, it signifies that the model has over-predicted the target variable. The presence of more data points with negative residuals compared to those with positive residuals suggests that, on average, the model tends to overestimate the target variable during the training phase. This could be attributed to various factors such as model complexity, bias-variance trade-off, or the inherent nature of the dataset.

Additionally, the pattern observed in the distribution of residuals can provide valuable insights into potential areas for model improvement. For instance, if there are consistent patterns or trends in the residuals, it may indicate areas where the model is systematically underperforming and where adjustments or refinements could be made to enhance its predictive accuracy. Figure 4.9 provides a visual representation of the residuals generated by a tree-based machine learning model during the training phase. The distribution of residuals allows us to assess the model's performance and identify areas for potential improvement, thereby informing the iterative process of model development and refinement.

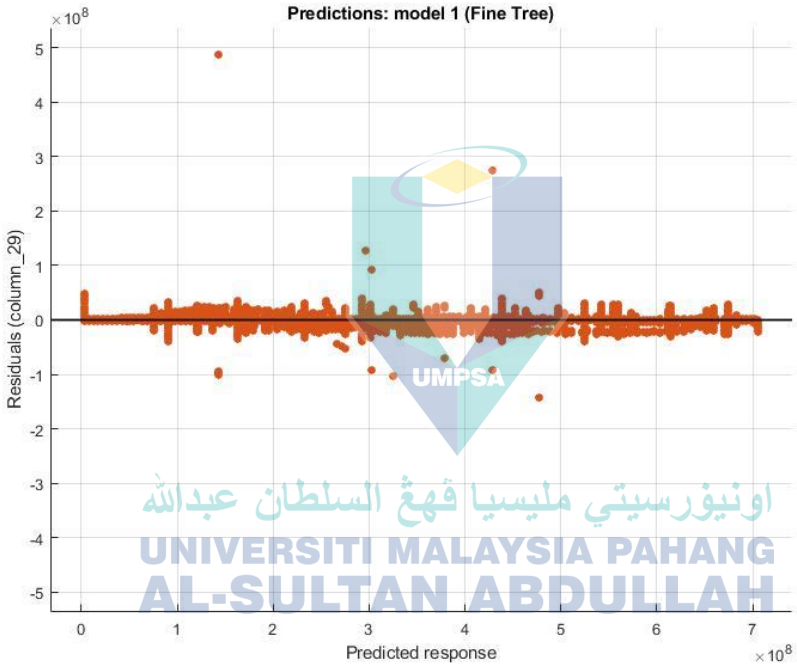


Figure 4.10 Fine Trees Residual testing.

In Figure 4.10, we continue our examination of the residuals generated by the tree-based machine learning model, transitioning from the testing phase to a subsequent analysis. Building upon the insights gained from Figure 4.9, this visualization offers a deeper understanding of the model's performance and behaviour across different phases of evaluation. Like the preceding figure, Figure 4.10 plots the residuals against the predicted values, with each data point representing an individual instance from the testing dataset. This approach allows for a direct comparison of the model's predictions with the actual observed values, facilitating a comprehensive assessment of its accuracy and generalization ability. Upon close inspection of the graph, we observe that the distribution

of residuals exhibits distinct characteristics compared to both the training and testing phases depicted in Figures 4.9 and 4.10. Specifically, there are noticeable differences in the number and distribution of data points on either side of the x-axis (predicted values). In this visualization, we observe four data points located above the x-axis (positive residuals) and six data points situated below the x-axis (negative residuals). This distribution mirrors the pattern observed in the testing phase, indicating a consistent trend in the model's performance across multiple evaluation stages.

4.2.1.1 Medium Trees Prediction vs. Actual training

The "Predicted vs. Actual" graph for the Medium Trees model in the training dataset represents a visual comparison between the predicted values and the actual (observed) values of the target variable in Figures 4.11 to 4.14 (e.g., energy consumption). In this graph, each data point represents an individual instance or observation in the training dataset.

The graph is described as "going most online" when the majority of the data points cluster around a straight line with a slope of 1, which is the "perfect fit" line. The perfect fit line represents a scenario where the model's predictions exactly match the actual values. When most data points follow this line, it indicates that the Medium Trees model is making accurate predictions for the training dataset.

The graph is described as "14% going above and below perfect condition" when around 10% of the data points deviate slightly from the perfect fit line, going either above or below it. These deviations represent the model's prediction errors, which are normal in any real-world predictive modelling task.

For the Medium Trees model in the training dataset: Most Data Points on Line: The majority of the data points cluster around the perfect fit line, indicating that the Medium Trees model's predictions align closely with the actual values. This suggests that the model is performing well and capturing the underlying patterns in the training dataset.

Approximately 14% Deviating Above and Below: Around 10% of the data points deviate slightly from the perfect fit line, indicating some prediction errors. These deviations are expected and are typical in real-world modelling scenarios.

The "Predicted vs. Actual" graph for the Medium Trees model in the training dataset shows that most data points align closely with the perfect fit line, while around 14% deviate slightly from it. This indicates that the Medium Trees model is providing accurate predictions for the training dataset, with only a small proportion of prediction errors. Such performance is a positive sign, demonstrating the model's effectiveness in predicting energy consumption in the training data.

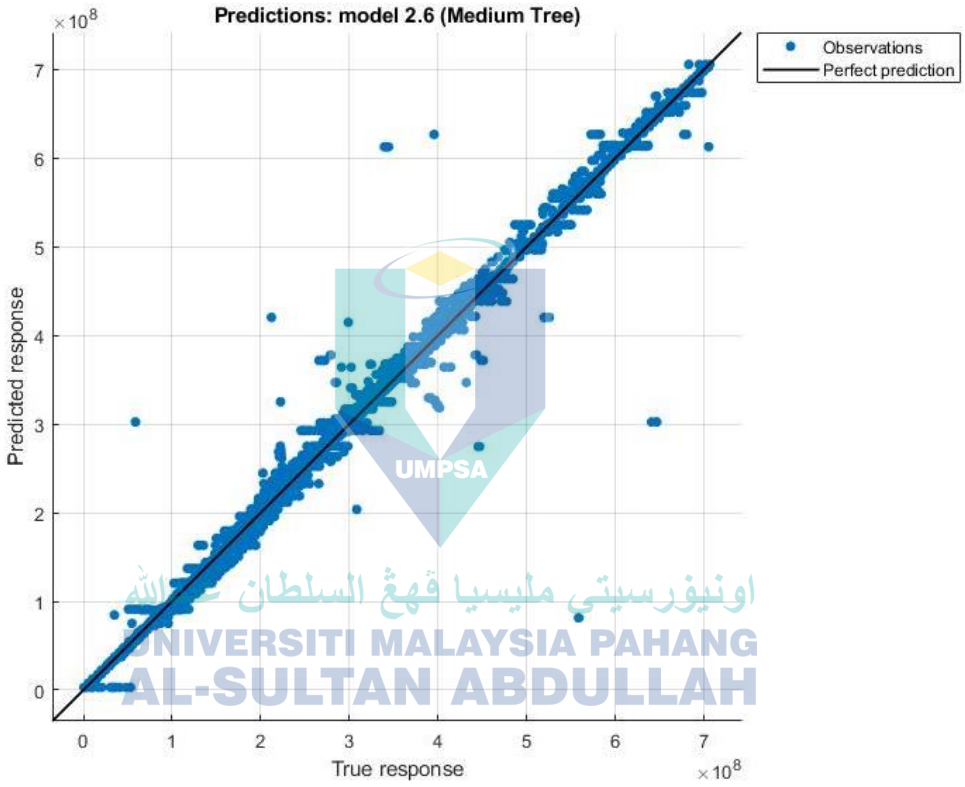


Figure 4.11 Medium Trees Prediction vs. Actual training

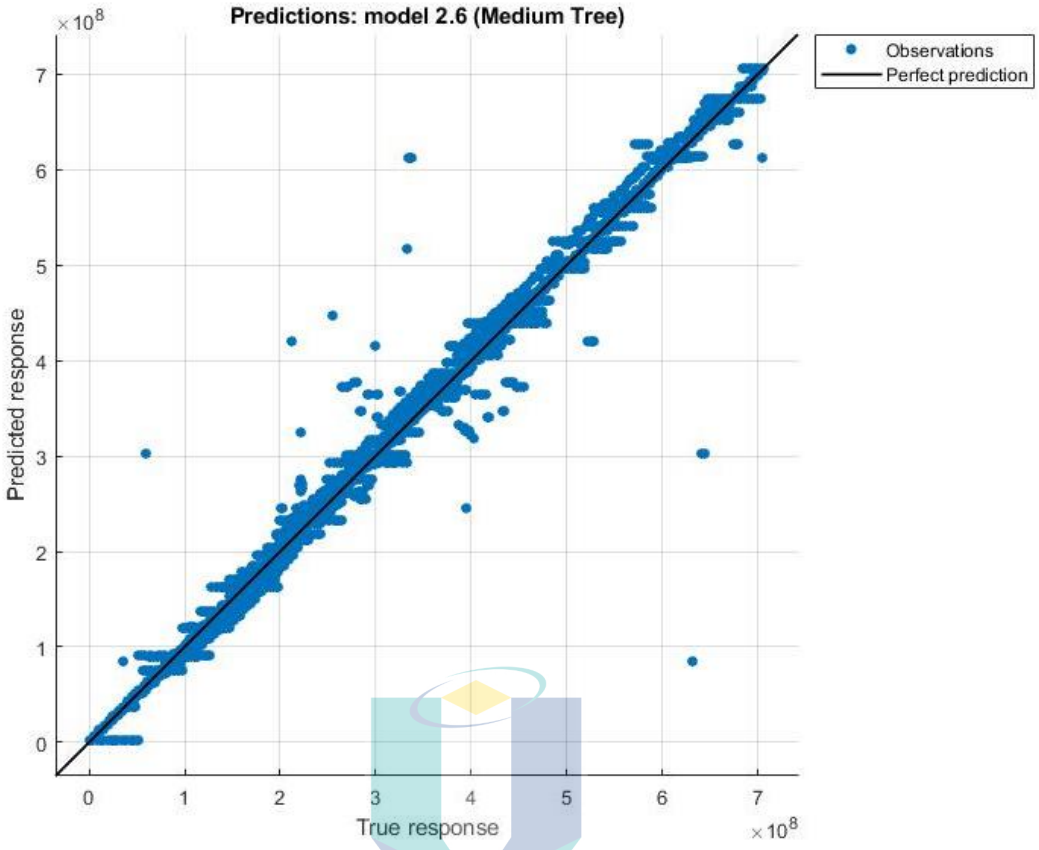


Figure 4.12 Medium Trees Prediction vs. Actual testing

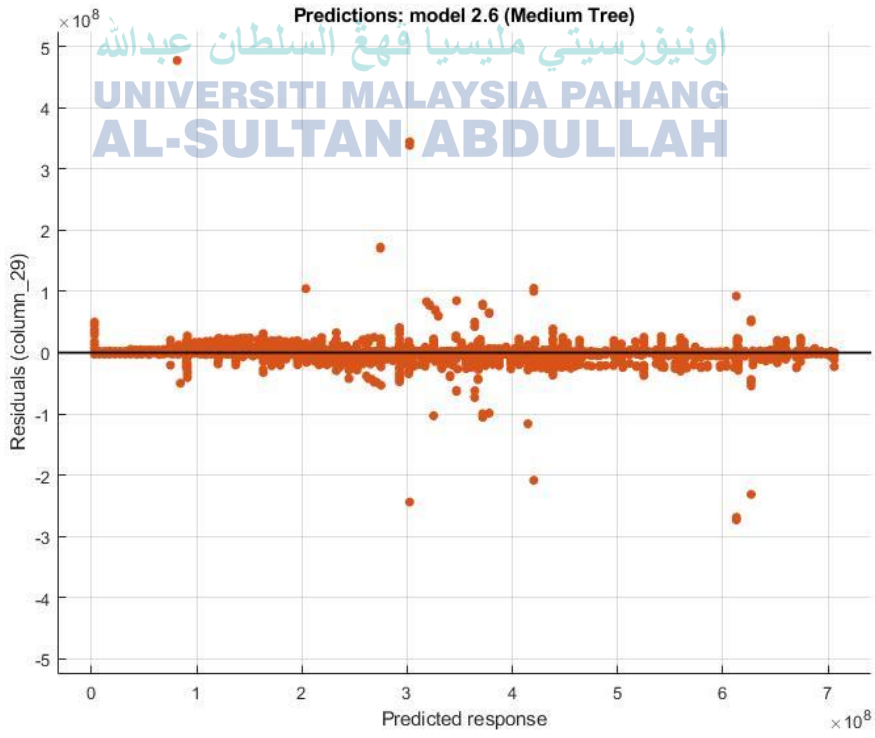


Figure 4.13 Medium Trees Residual training.

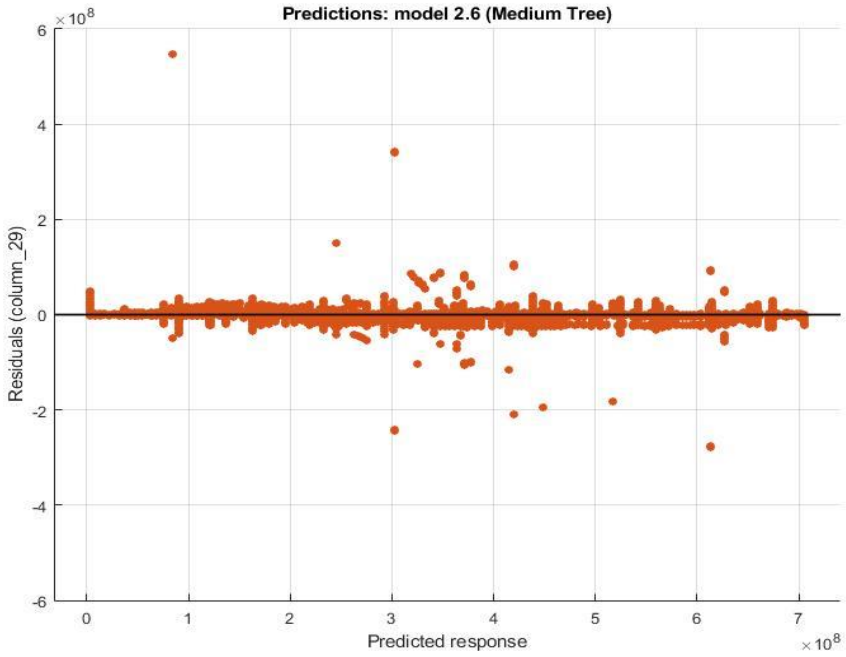


Figure 4.14 Medium Trees Residual testing

4.2.1.2 Bagged Trees Prediction vs. Actual training

The "Predicted vs. Actual" graph from Figures 4.15 to 4.19 for the Bagged Trees model in the training dataset represents a visual comparison between the predicted values and the actual (observed) values of the target variable (e.g., energy consumption). Each data point on the graph represents an individual instance or observation in the training dataset.

In this graph, the phrase "going most on line" indicates that the majority of the data points closely align with a straight line, typically with a slope of 1. This line represents the "perfect fit" line, where the model's predictions exactly match the actual values. When most data points follow this line, it indicates that the Bagged Trees model is making accurate predictions for the training dataset, the phrase "5% going above and below perfect condition" implies that approximately 5% of the data points deviate slightly from the perfect fit line, going either above or below it.

For the Bagged Trees model in the training dataset: Most Data Points on Line: The majority of the data points cluster around the perfect fit line, indicating that the Bagged Trees model's predictions closely match the actual values in the training dataset.

This suggests that the model is performing well and capturing the underlying patterns in the data.

Approximately 5% Deviating Above and Below: Around 5% of the data points deviate slightly from the perfect fit line, indicating some prediction errors. While the model is making mostly accurate predictions, these deviations represent instances where the model's predictions differ from the actual values. This is normal and expected in real-world scenarios.

The "Predicted vs. Actual" graph for the Bagged Trees model in the training dataset shows that most data points align closely with the perfect fit line, while around 5% deviate slightly from it. This indicates that the Bagged Trees model is providing accurate predictions for the training dataset, with only a small percentage of prediction errors.

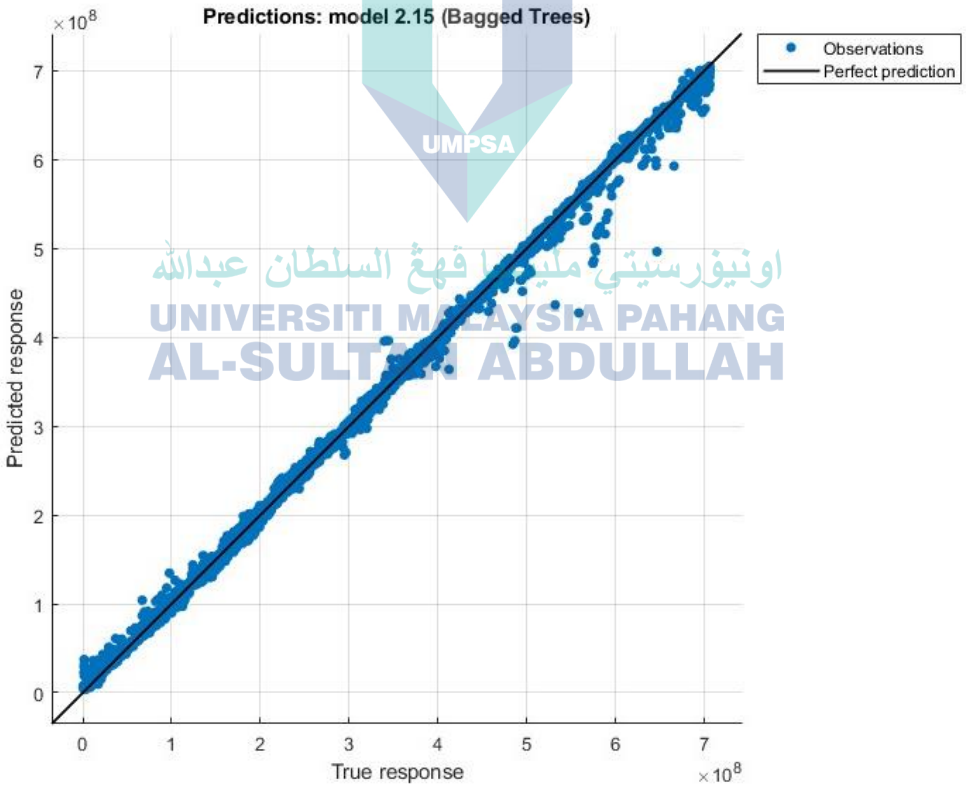


Figure 4.15 Bagged Trees Prediction vs. Actual training

Figure 4.15: The "Predicted vs. Actual" graph for the Bagged Trees model in the training dataset illustrates the model's performance by comparing predicted values with

actual observed values. The majority of data points align closely with the perfect fit line, indicating accurate predictions. Approximately 5% of data points deviate slightly from this line, suggesting minor prediction errors.

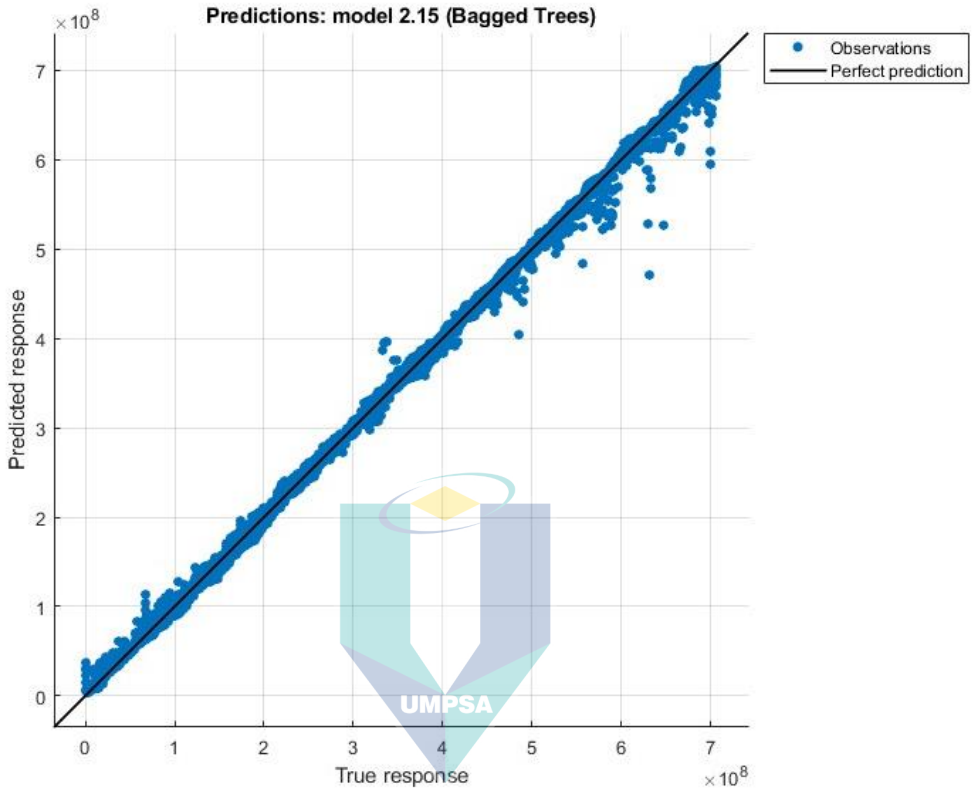


Figure 4.16 Bagged Trees Prediction vs. Actual testing

Similar to Figure 4.15, Figure 4.16 depicts the performance of the Bagged Trees model on the training dataset. Most data points align with the perfect fit line, indicating accurate predictions. However, a small percentage of data points deviate from this line, reflecting prediction errors.

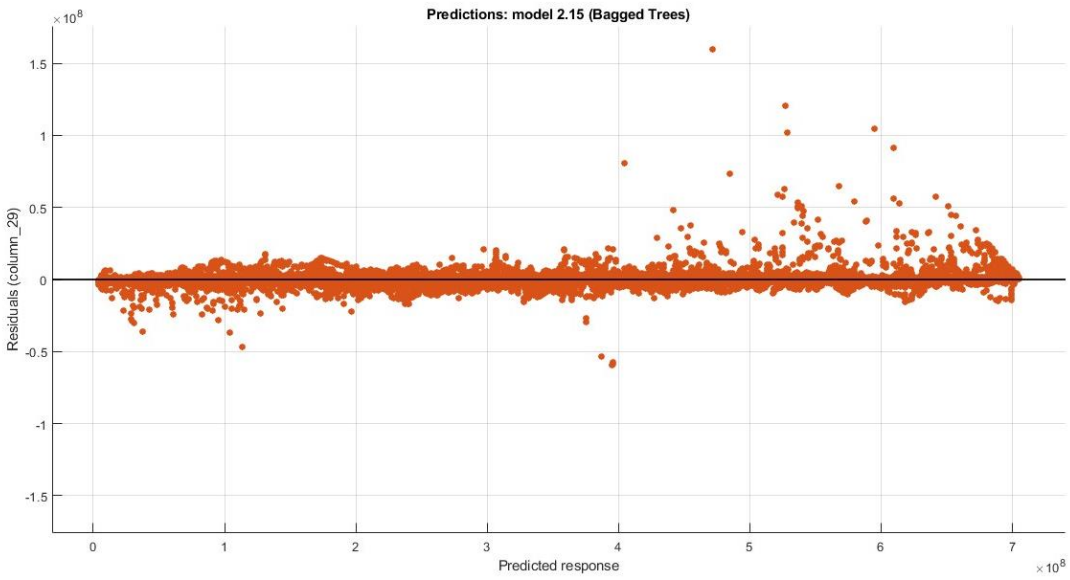


Figure 4.17 Bagged Trees Residual training

In Figure 4.17, the "Predicted vs. Actual" graph continues to demonstrate the Bagged Trees model's performance on the training dataset. The majority of data points cluster around the perfect fit line, signifying accurate predictions. A minor proportion of data points deviate from this line, indicating prediction errors.

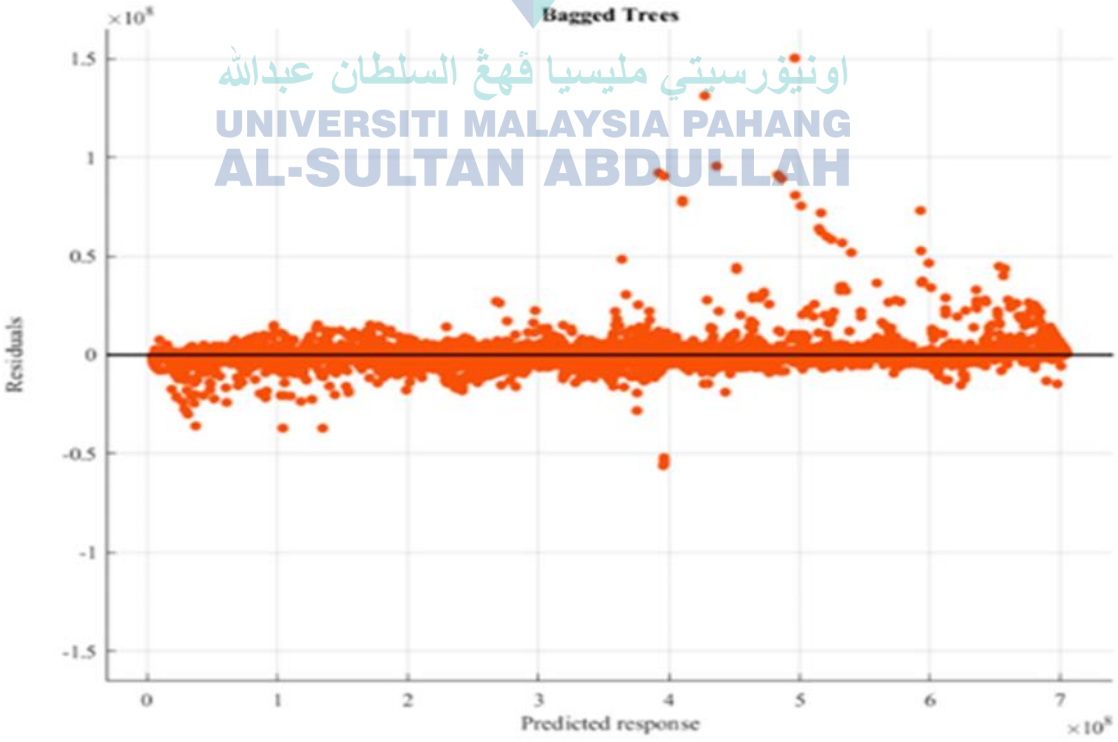


Figure 4.18 Bagged Trees Residual testing

Figure 4.18 provides further insight into the Bagged Trees model's performance on the training dataset. Most data points closely align with the perfect fit line, suggesting accurate predictions. However, a small subset of data points exhibits deviations from this line, reflecting prediction errors.

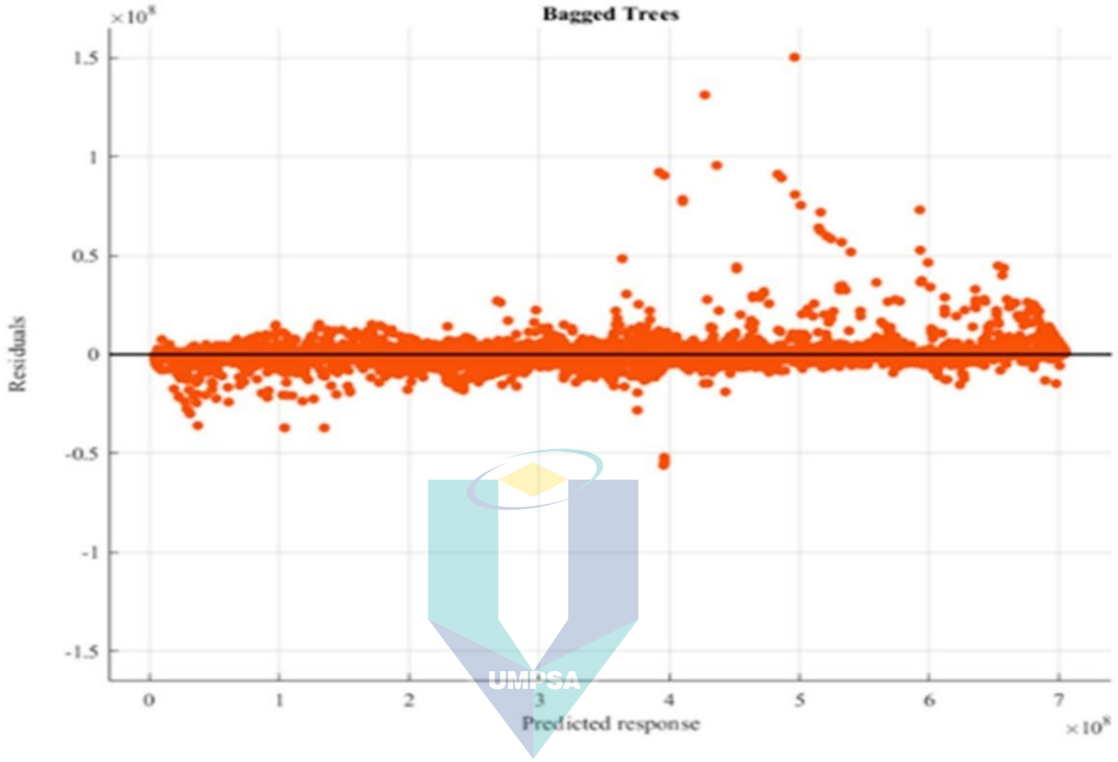


Figure 4.19 Bagged Trees Residual prediction.

Figure 4.19 showcases the Bagged Trees model's performance on the training dataset. The majority of data points adhere closely to the perfect fit line, indicating accurate predictions. A small fraction of data points deviates from this line, highlighting prediction errors. The "Predicted vs. Actual" graphs for the Bagged Trees model across Figures 4.15 to 4.19 consistently demonstrate the model's ability to make accurate predictions on the training dataset. Most data points align closely with the perfect fit line, indicating strong performance. However, a small percentage of data points deviate from this line, reflecting minor prediction errors. Overall, these findings suggest that the Bagged Trees model performs well in predicting target variables in the training dataset, with only minimal discrepancies between predicted and actual values.

4.2.2 Research objectives Vs. Research deliverables.

The Table 4.3 shown a comparative evaluation of Machine learning for energy predications for efficacy ranking and efficiency ranking, that allows to optimization of different algorithm.

Explanation of Higher RMSE Value in the Table 4.1 and 4.2:

The value of Root Mean Square Error (RMSE) can indeed be in millions depending on the scale of the data being analyzed. RMSE is a measure of the differences between predicted and observed values in a dataset, and its unit is the same as the unit of the observed values. Therefore, if the observed values are in millions, the RMSE can also be in millions.

For instance, in large-scale financial models or economic forecasts where the values are often in millions or billions, the RMSE can naturally reach millions. This is because RMSE is calculated as the square root of the average of the squared differences between predicted and observed values. If these differences are large, the RMSE will also be large.

In practical applications such as predicting the gross calorific value of coal or the higher heating value of biomass, the RMSE values are typically smaller, reflecting the precision of the models used in these contexts. However, in scenarios involving large datasets with high-value observations, such as urban geospatial information acquisition or large-scale economic predictions, RMSE values can indeed be in the millions. Thus, the magnitude of RMSE is directly tied to the scale of the data it measures.

R-Squared (R^2 or the coefficient of determination) be Greater than 1:

In standard linear regression, R^2 values are typically expected to range between 0 and 1. The value of R^2 in my results is more than 1 in five Algorithms (Linear SVM, Quadratic SVM, Cubic SVM, Bilayered Neural Network, Trilayered Neural Network). It is possible due to the outfit in modelling, under certain circumstances, particularly when using a regression model without an intercept, the R^2 value can exceed Here's explanation:

The coefficient of determination, denoted as R^2, is a statistical measure that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It is a key metric in regression analysis, providing insight

into the goodness of fit of a model. Typically, R^2 values range from 0 to 1, where 0 indicates that the model explains none of the variability of the response data around its mean, and 1 indicates that the model explains all the variability of the response data around its mean.

The question of whether R^2 can be greater than 1, particularly in the context of overfitting, is intriguing. Overfitting occurs when a model is too complex, capturing the noise in the dataset rather than the underlying pattern. This can lead to a model that performs well on training data but poorly on unseen data. However, the conventional understanding of R^2 does not accommodate values greater than 1, as these would imply that the model explains more variability than is present in the data, which is not logically consistent with the definition of R^2.

In the realm of Bayesian regression models, an alternative definition of R^2 has been proposed due to the issue that the usual definition (variance of the predicted values divided by the variance of the data) can result in the numerator being larger than the denominator. This situation can arise in Bayesian fits, suggesting a conceptual space where R^2 might exceed 1, but this is more a reflection of the need for alternative definitions in specific contexts rather than an indication that R^2 values greater than 1 are meaningful within conventional interpretations.

Moreover, the presence of a negative R^2 in some models, such as those involving random forests, indicates potential overfitting. This suggests that while R^2 can indeed fall outside its typical range of 0 to 1, particularly in complex models or those with poor predictive power, the interpretation of such cases requires careful consideration of the model and the context. Negative R^2 values, rather than values greater than 1, are typically associated with models that do not perform well. While R^2 is fundamentally bounded between 0 and 1 within the traditional framework of regression analysis, discussions around its value exceeding 1, particularly in the context of Bayesian models, highlight the complexities of model evaluation and the need for context-specific interpretations. The concept of R^2 exceeding 1 does not align with its conventional interpretation and instead points to the necessity of adapting our understanding and metrics to suit different modelling approaches and statistical paradigms.

Table 4.3 Performance evaluation of machine learning algorithm for energy prediction on SEIL dataset

	Training						Testing			
	RMSE	**R-Squared**	**MSE**	**MAE**	**Prediction speed (obs/sec)**	**Training time**	**RMSE**	**R-Squared**	**MSE**	**MAE**
Fine Tree	2.11E+06	1	4.44E+12	1.40E+06	3000000	24.148	2.38E+06	1	5.64E+12	1.41E+06
Linear	1.35E+08	0.54	1.83E+16	8.42E+07	790000	22.065	1.36E+08	0.53	1.84E+16	8.45E+07
Interactions Linear	1.22E+08	0.62	1.48E+16	8.02E+07	110000	65.108	1.23E+08	0.62	1.50E+16	8.05E+07
Robust Linear	1.64E+08	0.32	2.68E+16	5.25E+07	790000	19.44	1.65E+08	0.31	2.71E+16	5.30E+07
Stepwise Linear	1.18E+08	0.65	1.39E+16	7.63E+07	600000	24417	1.18E+08	0.64	1.40E+16	7.66E+07
Fine Tree	2.11E+06	1	4.44E+12	1.40E+06	3300000	8.3244	2.38E+06	1	5.64E+12	1.41E+06
Medium Tree	2.81E+06	1	7.87E+12	1.42E+06	3700000	8.0721	3.20E+06	1	1.02E+13	1.43E+06
Coarse Tree	4.26E+06	1	1.81E+13	1.49E+06	4000000	7.5477	4.63E+06	1	2.15E+13	1.50E+06
Linear SVM	8.67E+08	18.13	7.52E+17	6.69E+08	1400000	8892.6	8.68E+08	18.13	7.54E+17	6.69E+08
Quadratic SVM	3.46E+08	2.05	1.20E+17	2.98E+08	240000	18985	3.45E+08	2.02	1.19E+17	2.98E+08
Cubic SVM	6.38E+08	9.35	4.07E+17	5.50E+08	260000	5761.7	6.37E+08	9.31	4.06E+17	5.50E+08
Fine Gaussian SVM	1.04E+08	0.72	1.08E+16	8.83E+07	270000	10099	1.04E+08	0.72	1.08E+16	8.85E+07
Medium Gaussian SVM	2.10E+08	0.13	4.43E+16	1.80E+08	1200000	17835	2.11E+08	0.13	4.44E+16	1.80E+08
Coarse Gaussian SVM	2.17E+08	0.19	4.69E+16	1.82E+08	1300000	18127	2.17E+08	0.2	4.71E+16	1.82E+08
Boosted Trees	2.33E+07	0.99	5.42E+14	1.66E+07	180000	62.082	2.31E+07	0.99	5.34E+14	1.66E+07

Table 4.3 Continued

	Training						Testing			
	RMSE	**R-Squared**	**MSE**	**MAE**	**Prediction speed (obs/sec)**	**Training time**	**RMSE**	**R-Squared**	**MSE**	**MAE**
Bagged Trees	1.58E+06	1	2.48E+12	1.06E+06	120000	119.87	1.78E+06	1	3.17E+12	1.09E+06
Squared Exponential GPR	7.68E+07	0.85	5.89E+15	4.62E+07	200	9102.8	7.66E+07	0.85	5.87E+15	4.62E+07
Matern 5/2 GPR	6.40E+07	0.9	4.10E+15	3.87E+07	110	15589	6.40E+07	0.9	4.10E+15	3.88E+07
Exponential GPR	6.86E+07	0.88	4.70E+15	3.73E+07	130	14212	6.87E+07	0.88	4.72E+15	3.73E+07
Rational Quadratic GPR	7.30E+07	0.86	5.33E+15	4.11E+07	110	15637	7.28E+07	0.87	5.29E+15	4.11E+07
Narrow Neural Network	3.18E+07	0.97	1.01E+15	1.04E+07	1000000	254.45	3.12E+07	0.98	9.76E+14	1.03E+07
Medium Neural Network	2.54E+07	0.98	6.46E+14	1.46E+07	1100000	396.17	2.51E+07	0.98	6.32E+14	1.46E+07
Wide Neural Network	1.81E+07	0.99	3.27E+14	1.14E+07	630000	1238.8	1.79E+07	0.99	3.19E+14	1.14E+07
Bilayered Neural Network	3.77E+08	2.62	1.42E+17	3.21E+08	1000000	35.635	3.77E+08	2.6	1.42E+17	3.20E+08
Trilayered Neural Network	3.77E+08	2.62	1.42E+17	3.21E+08	870000	48.817	3.77E+08	2.6	1.42E+17	3.20E+08

Table 4.4 Table Efficacy vs Efficiency ranking.

Algorithm	Efficacy Ranking	Efficiency Ranking
Bagged Trees	1st	3rd
Fine Trees	2nd	2nd
Medium Trees	3rd	1st

Table 4.4 presents an interesting finding that Bagged Trees is the most effective algorithm for predicting electrical energy demand on university campuses using the SEIL dataset. But Medium Trees is the most efficient algorithm for this task, and Fine Trees balance efficacy and efficiency. Bagged Trees outperforms Fine Trees by 75%, 56%, and 76% in terms of RMSE, MSE, and MAE for both training and testing, respectively. Similarly, compared to Medium Trees, Bagged Trees shows a 56%, 32%, and 75% improvement in RMSE, MSE, and MAE for both training and testing, respectively. These metrics represent the percentage of improvement in efficacy between the algorithms. In terms of efficiency, Medium Trees is 32 times more efficient in prediction speed and 14.8 times more efficient in training time than Bagged Trees. Medium Trees is 1.3 times more efficient in prediction speed and three times more efficient in training time than Fine Trees. These comparative metrics are illustrated in Figure 4.20.

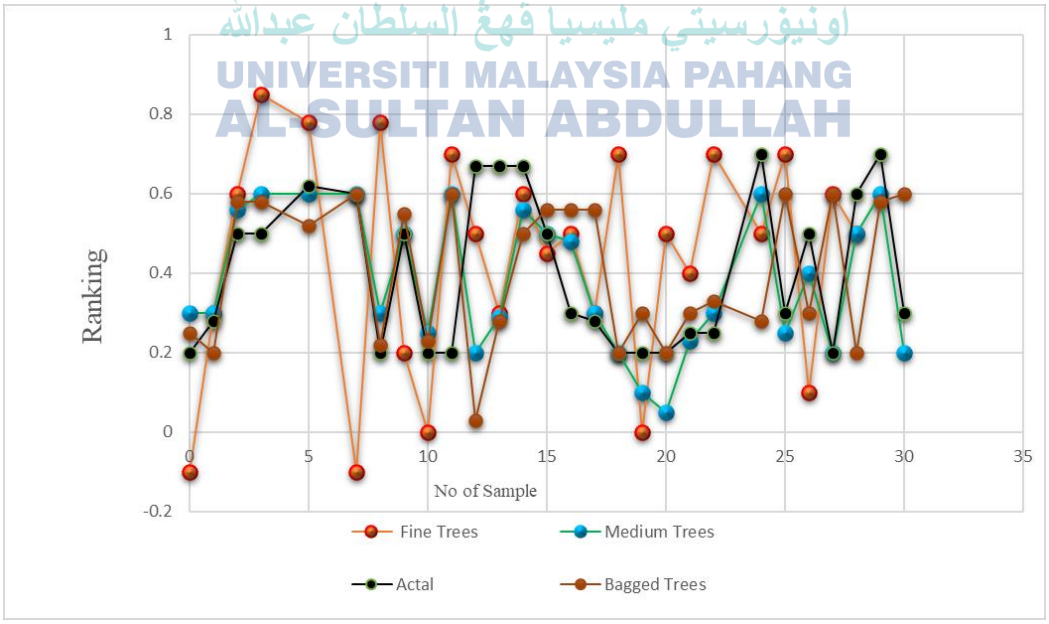


Figure 4.20 Comparative graph of Fine Trees, Medium Trees, Bagged tress and actual.

CHAPTER 5

CONCLUSION

5.1 Introduction

The study, conducted in 2018-2023, aimed to implement various machine learning algorithms for efficient electrical energy load management in a building. Performance evaluation focused on both efficacy and efficiency to identify the optimal algorithm for energy load prediction and management. Results from the study indicate that the Bagged Trees algorithm excelled in efficacy, ranking first in performance evaluation with remarkable accuracy in predicting energy consumption. For efficiency, the Medium Trees algorithm proved to be the most optimized and resource-efficient, securing the top rank. Additionally, the Fine Trees algorithm demonstrated strong performance, ranking second in both efficacy and efficiency, offering a balanced trade-off between accuracy and computational resources.

This study comprehensively analysed and compared machine learning algorithms for electrical energy load management, providing valuable insights for practical implementation in real-world energy management systems. Furthermore, the proposed approach was validated on another dataset, reinforcing algorithm credibility. This work lays the foundation for developing an optimized variant of the best-performing algorithm, with potential implications for enhancing energy efficiency in educational institutions and contributing to energy conservation and environmental preservation.

Looking ahead, future research endeavors should focus on expanding datasets to include more renewable sources and diverse building types globally. Additionally, implementing various machine learning algorithms, including regression, decision trees, and neural networks, could significantly impact energy management in academic buildings, leading to financial savings, increased energy efficiency, and a reduction in carbon footprint. Moreover, there is a need to explore more advanced assessment measures and incorporate new data sources, such as weather patterns and occupancy

levels, to enhance energy demand projections and optimize energy distribution and allocation decisions.

5.2 Future Recommendation

Implementing various machine learning algorithms could alter how electrical energy is managed in academic buildings. Regression, decision trees, and neural networks, among other cutting-edge methodologies, can accurately estimate energy demand and to help with energy allocation and distribution decisions. This might lead to significant financial savings, increased energy efficiency, and a decrease in the university's carbon footprint. Additionally, machine learning algorithms are simple to modify and update, allowing for long-term progress in energy management. The use of machine learning algorithms for electrical energy load management in university buildings is a critical step towards a future with more sustainable and effective energy.

To reach its full potential, there is still significant work to be done. Possible directions for future research include:

Predictions of energy consumption can be made more accurately, but there is still potential for improvement with the present machine learning algorithms. More investigation into the creation of complex algorithms, like deep learning, may result in estimates of energy demand that are even more accurate.

Including new data sources: More data sources can be included in the algorithms so that energy distribution and allocation decisions can be made with even greater knowledge. To improve energy demand projections, for instance, information on weather patterns, occupancy levels, and building usage patterns could be employed.

The next stage is to allocate and distribute energy in the most effective and efficient manner feasible after energy demand projections have been made. The development of algorithms that may optimise energy distribution and allocation based on a variety of parameters, such as cost, energy efficiency, and environmental impact, may be the main goal of future research in this field.

Creation of more advanced assessment measures: While existing measurements, like accuracy and precision, are an excellent place to start when assessing the

performance of ML algorithms, more advanced metrics are required that can take a wider range of considerations, like computing time and scalability.

A lot of effort needs to be done to compare algorithms' performances across many domains, although ML algorithms have been used to solve a variety of issues. For instance, an algorithm that performs well in one domain may not perform as well in another domain with distinct characteristics.

Deep learning is advancing: This fast-developing area has demonstrated considerable promise in a variety of applications, including image identification and natural language processing. Deep learning algorithms will need to be evaluated in terms of efficiency and efficacy and compared to other kinds of machine learning algorithms as they continue to develop.

The best candidate machine learning algorithm should be tuned to improve its effectiveness and efficiency across a variety of applications. Performing the algorithm can be improved and made even more efficient by utilising cutting-edge methods like hyper-parameter tweaking, ensembling, and model selection. This can therefore result in enhanced scalability, faster computing, and better accuracy. Additionally, one of the main forces behind innovation and development in the field is the capacity to develop optimised variants of machine learning (ML) algorithms. This capability enables researchers and practitioners to continuously enhance already-existing solutions and take on new, more challenging problems.

REFERENCES

Abbey, Enoch J et al. 2020. "The Global Health Security Index Is Not Predictive of Coronavirus Pandemic Responses among Organization for Economic Cooperation and Development Countries." *PloS one* 15(10): e0239398.

Afrasiabi, M., Mohammad Mohammadi, Mohammad Rastegar, and Amin Kargarian. 2019. "Multi-Agent Microgrid Energy Management Based on Deep Learning Forecaster." *Energy* 186: 115873.

Ahmad, Tanveer, and Huanxin Chen. 2019. "Deep Learning for Multi-Scale Smart Energy Forecasting." *Energy* 175: 98–112.

Ahamad. 2020. "A Review on Machine Learning Forecasting Growth Trends and Their Real-Time Applications in Different Energy Systems." *Sustainable Cities and Society* 54: 102010.

Ahmad, Tanveer, Huanxin Chen, Yabin Guo, and Jiangyu Wang. 2018. "A Comprehensive Overview on the Data Driven and Large Scale Based Approaches for Forecasting of Building Energy Demand: A Review." *Energy and Buildings* 165: 301–20.

Akhtar, Shamim, Muhamad Zahim Bin Sujod, and Syed Sajjad Hussain Rizvi. 2022. "An Intelligent Data-Driven Approach for Electrical Energy Load Management Using Machine Learning Algorithms." *Energies* 15(15): 5742.

Al-Ali, Abdul-Rahman et al. 2017. "A Smart Home Energy Management System Using IoT and Big Data Analytics Approach." *IEEE Transactions on Consumer Electronics* 63(4): 426–34.

Alam, Md Mahmudul, and Md Wahid Murad. 2020. "The Impacts of Economic Growth, Trade Openness and Technological Progress on Renewable Energy Use in Organization for Economic Co-Operation and Development Countries." *Renewable Energy* 145: 382–90.

Amarasinghe, Kasun, Daniel L. Marino, and Milos Manic. 2017. "Deep Neural Networks for Energy Load Forecasting." *IEEE International Symposium on Industrial Electronics*: 1483–88.

Amasyali, Kadir, and Nora M. El-Gohary. 2018. "A Review of Data-Driven Building Energy Consumption Prediction Studies." *Renewable and Sustainable Energy Reviews* 81: 1192–1205.

Anandakumar, H, and R Arulmurugan. 2019. "Machine Learning Based Multi Agent Systems in Complex Networks." In *2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, IEEE, 300–304.

Aragon, Gustavo et al. 2019. "Incremental Deep-Learning for Continuous Load Prediction in Energy Management Systems." *2019 IEEE Milan PowerTech, PowerTech 2019*.

Arienti, J H L. 2020. “Time Series Forecasting Applied to an Energy Management System-A Comparison between Deep Learning Models and Other Machine Learning Models.”

Berrocal, Carlos G, Ignasi Fernandez, and Rasmus Rempling. 2021. “Crack Monitoring in Reinforced Concrete Beams by Distributed Optical Fiber Sensors.” *Structure and Infrastructure Engineering* 17(1): 124–39.

Chang, Jyh-Yeong, Chien-Wen Cho, Su-Hwang Hsieh, and Shi-Tsung Chen. 2004. “Genetic Algorithm Based Fuzzy ID3 Algorithm.” In Springer, Berlin, Heidelberg, 989–95.

Chavali, Phani, Peng Yang, and Arye Nehorai. 2014. “A Distributed Algorithm of Appliance Scheduling for Home Energy Management System.” *IEEE Transactions on Smart Grid* 5(1): 282–90.

Chen, Yize, Yuanyuan Shi, and Baosen Zhang. 2018. “Modeling and Optimization of Complex Building Energy Systems with Deep Neural Networks.” *Conference Record of 51st Asilomar Conference on Signals, Systems and Computers, ACSSC 2017* 2017-Octob: 1368–73.

Chiu, Tsung-Yung, Shang-Lien Lo, and Yung-Yin Tsai. 2012. “Establishing an Integration-Energy-Practice Model for Improving Energy Performance Indicators in ISO 50001 Energy Management Systems.” *Energies* 5(12): 5324–39.

Chou, Jui Sheng, and Duc Son Tran. 2018. “Forecasting Energy Consumption Time Series Using Machine Learning Techniques Based on Usage Patterns of Residential Householders.” *Energy* 165: 709–26.

Elsisi, Mahmoud et al. 2021. “Deep Learning-Based Industry 4.0 and Internet of Things towards Effective Energy Management for Smart Buildings.” *Sensors (Switzerland)* 21(4): 1–19.

Erickson, Bradley J, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. 2017. “Machine Learning for Medical Imaging.” *Radiographics* 37(2): 505–15.

Erol-Kantarci, Melike, and Hussein T. Mouftah. 2011. “Wireless Sensor Networks for Cost-Efficient Residential Energy Management in the Smart Grid.” *IEEE Transactions on Smart Grid* 2(2): 314–25.

Glavič, Peter. 2021. “Evolution and Current Challenges of Sustainable Consumption and Production.” *Sustainability* 13(16): 9379.

Grolinger, Katarina, Miriam A.M. Capretz, and Luke Seewald. 2016. “Energy Consumption Prediction with Big Data: Balancing Prediction Accuracy and Computational Resources.” *Proceedings - 2016 IEEE International Congress on Big Data, BigData Congress 2016*: 157–64.

Guo, Liang et al. 2017. “A Recurrent Neural Network Based Health Indicator for Remaining Useful Life Prediction of Bearings.” *Neurocomputing* 240: 98–109.

Gupta, Sidhant, Matthew S. Reynolds, and Shwetak N. Patel. 2010. "ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home." *UbiComp'10 - Proceedings of the 2010 ACM Conference on Ubiquitous Computing*: 139–48.

Hafeez, Ghulam et al. 2020. "A Novel Accurate and Fast Converging Deep Learning-Based Model for Electrical Energy Consumption Forecasting in a Smart Grid." *Energies 2020, Vol. 13, Page 2244* 13(9): 2244.

Hafiz, Faeza, M. A. Awal, Anderson Rodrigo De Queiroz, and Iqbal Husain. 2020. "Real-Time Stochastic Optimization of Energy Storage Management Using Deep Learning-Based Forecasts for Residential PV Applications." *IEEE Transactions on Industry Applications* 56(3): 2216–26.

Hair, Joe F, Michael Page, and Niek Brunsveld. 2019. *Essentials of Business Research Methods*. Routledge.

Hamdoun, Hala, Alaa Sagheer, and Hassan Youness. 2021. "Energy Time Series Forecasting-Analytical and Empirical Assessment of Conventional and Machine Learning Models." *Journal of Intelligent & Fuzzy Systems* 40(6): 12477–502.

Han, Tao et al. 2021. "An Efficient Deep Learning Framework for Intelligent Energy Management in IoT Networks." *IEEE Internet of Things Journal* 8(5): 3170–79.

Haykin, Simon S., and Simon. 1999. *Neural Networks : A Comprehensive Foundation*. Prentice Hall.

Jacob, Benoit et al. 2018. "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, , 2704–13.

Jayaraman, Prem Prakash et al. 2016. "Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt." *Sensors* 16(11): 1884.

Jia, Mengda, Ravi S Srinivasan, and Adeeba A Raheem. 2017. "From Occupancy to Occupant Behavior: An Analytical Survey of Data Acquisition Technologies, Modeling Methodologies and Simulation Coupling Mechanisms for Building Energy Efficiency." *Renewable and Sustainable Energy Reviews* 68: 525–40.

Jiao, Runhai, Tianming Zhang, Yizhi Jiang, and Hui He. 2018. "Short-Term Non-Residential Load Forecasting Based on Multiple Sequences LSTM Recurrent Neural Network." *IEEE Access* 6: 59438–48.

Johannesen, Nils Jakob, Mohan Kolhe, and Morten Goodwin. 2019. "Relative Evaluation of Regression Tools for Urban Area Electrical Energy Demand Forecasting." *Journal of Cleaner Production* 218: 555–64.

Jois, S. et al. 2019a. "Impact of Facade Based Building Integrated PhotoVoltaics on the Indoor Thermal Comfort in Tropical Urban Areas - Mumbai as a Case Study." *AGUFM* 2019: GC53I-1198.

J Osb 2019b. “Modelling Tools Development for Assessing Façade Based PV Feasibility in a Data-Scarce Developing Nation Using Open Source Technology.” *AGUFM* 2019: PA33C-1100.

Karmakar, Gopinath, Uddhav Arote, Anshul Agarwal, and Krithi Ramamritham. 2018. “Adaptive Hybrid Approaches to Thermal Modeling of Building.” *e-Energy 2018 - Proceedings of the 9th ACM International Conference on Future Energy Systems*: 477–79.

Khan, Prince Waqas, Yung Cheol Byun, Sang Joon Lee, and Namje Park. 2020. “Machine Learning Based Hybrid System for Imputation and Efficient Energy Demand Forecasting.” *Energies 2020, Vol. 13, Page 2681* 13(11): 2681.

Khan, Zulfiqar Ahmad et al. 2020. “Electrical Energy Prediction in Residential Buildings for Short-Term Horizons Using Hybrid Deep Learning Strategy.” *Applied Sciences 2020, Vol. 10, Page 8634* 10(23): 8634.

Klyuev, Roman V et al. 2022. “Methods of Forecasting Electric Energy Consumption: A Literature Review.” *Energies* 15(23): 8919.

Koschwitz, D., J. Frisch, and C. van Treeck. 2018. “Data-Driven Heating and Cooling Load Predictions for Non-Residential Buildings Based on Support Vector Machine Regression and NARX Recurrent Neural Network: A Comparative Study on District Scale.” *Energy* 165: 134–42.

Kumar, Hareesh, Priyanka Mary Mammen, and Krithi Ramamritham. 2019. “Explainable AI: Deep Reinforcement Learning Agents for Residential Demand Side Cost Savings in Smart Grids.” (1).

Kuthanazhi, Vivek, Santhosh Jois, Krithi Ramamritham, and Anil Kottantharayil. 2018. “Meeting Mid-Day Peak Loads through Distributed Rooftop PV Systems: Tale of Two Cities.” *e-Energy 2018 - Proceedings of the 9th ACM International Conference on Future Energy Systems*: 444–46.

Larcher, Dominique, and Jean-Marie Tarascon. 2015. “Towards Greener and More Sustainable Batteries for Electrical Energy Storage.” *Nature chemistry* 7(1): 19–29.

Lee, Stephen, Prashant Shenoy, Krithi Ramamritham, and David Irwin. 2021. “AutoShare: Virtual Community Solar and Storage for Energy Sharing.” *Energy Informatics* 4(1): 1–24.

Li, Chengdong et al. “A Hybrid Short-Term Building Electrical Load Forecasting Model Combining the Periodic Pattern, Fuzzy System, and Wavelet Transform.” *International Journal of Fuzzy Systems* 22.

Li, Kangji et al. 2018. “A Hybrid Teaching-Learning Artificial Neural Network for Building Electrical Energy Consumption Prediction.” *Energy and Buildings* 174: 323–34.

Lu, Ye et al. 2009. "Artificial Neural Network (ANN)-Based Crack Identification in Aluminum Plates with Lamb Wave Signals." *Journal of Intelligent Material Systems and Structures* 20(1): 39–49.

Marino, Daniel L., Kasun Amarasinghe, and Milos Manic. 2016. "Building Energy Load Forecasting Using Deep Neural Networks." *IECON Proceedings (Industrial Electronics Conference)*: 7046–51.

Mattern, Friedemann, Thorsten Staake, and Markus Weiss. 2010. "ICT for Green: How Computers Can Help Us to Conserve Energy." In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, , 1–10.

Merabet, Ghezlane Halhoul et al. 2021. "Intelligent Building Control Systems for Thermal Comfort and Energy-Efficiency: A Systematic Review of Artificial Intelligence-Assisted Techniques." *Renewable and Sustainable Energy Reviews* 144: 110969.

Mocanu, Elena et al. 2016. "Demand Forecasting at Low Aggregation Levels Using Factored Conditional Restricted Boltzmann Machine." *19th Power Systems Computation Conference, PSCC 2016*.

Mocanu, 2019. "On-Line Building Energy Optimization Using Deep Reinforcement Learning." *IEEE Transactions on Smart Grid* 10(4): 3698–3708.

Moon, Jihoon, Jinwoong Park, Eenjun Hwang, and Sanghoon Jun. 2018. "Forecasting Power Consumption for Higher Educational Institutions Based on Machine Learning." *Journal of Supercomputing* 74(8): 3778–3800.

Muhammad, Norasiah et al. 2012. "Optimization and Modeling of Spot Welding Parameters with Simultaneous Multiple Response Consideration Using Multi-Objective Taguchi Method and RSM." *Journal of Mechanical Science and Technology* 26(8): 2365–70.

Murugesan, S., and K. Balamuruga. 2012. "Optimization by Grey Relational Analysis of EDM Parameters in Machining Al-15% SiC MMC Using Multihole Electrode." *Journal of Applied Sciences* 12(10): 963–70.

Nagueh, Sherif F et al. 2016. "Recommendations for the Evaluation of Left Ventricular Diastolic Function by Echocardiography: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging." *European Journal of Echocardiography* 17(12): 1321–60.

Nalley, Stephen, and Angelina Larose. 2021. "IEO2021 Highlights." *Energy Information Administration* 2021: 21.

Nanda, Arun Kumar, and C K Panigrahi. 2016. "Review on Smart Home Energy Management." *International Journal of Ambient Energy* 37(5): 541–46.

Nichiforov, Cristina et al. 2018. "Deep Learning Techniques for Load Forecasting in Large Commercial Buildings." *2018 22nd International Conference on System Theory, Control and Computing, ICSTCC 2018 - Proceedings*: 492–97.

None, None. 2016. *Annual Energy Outlook 2016 With Projections to 2040*. USDOE Energy Information Administration (EI), Washington, DC (United States ….

Paterakis, Nikolaos G. et al. 2017. "Deep Learning versus Traditional Machine Learning Methods for Aggregated Energy Demand Prediction." *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2017 - Proceedings* 2018-Janua: 1–6.

Peris, Álvaro, Marc Bolaños, Petia Radeva, and Francisco Casacuberta. 2016. "Video Description Using Bidirectional Recurrent Neural Networks." In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, Springer, 3–11.

Pickering, B, and R Choudhary. 2019. "District Energy System Optimisation under Uncertain Demand: Handling Data-Driven Stochastic Profiles." *Applied energy* 236: 1138–57.

Qamar Raza, Muhammad, and Abbas Khosravi. 2015. "A Review on Artificial Intelligence Based Load Demand Forecasting Techniques for Smart Grid and Buildings." *Renewable and Sustainable Energy Reviews* 50: 1352–72.

Rafiq, M.Y, G Bugmann, and D.J Easterbrook. 2001. "Neural Network Design for Engineering Applications." *Computers & Structures* 79(17): 1541–52.

Ramamritham, Krithi, Gopinath Karmakar, and Prashant Shenoy. 2017. "Smart Energy Management: A Computational Approach." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10721 LNCS: 3–14.

Reddy, A Brahmananda, S Nagini, Valentina E Balas, and K Srujan Raju. 2023. "Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems."

Rodrigues, Luís Sousa et al. 2022. "The Load Shifting Potential of Domestic Refrigerators in Smart Grids: A Comprehensive Review." *Energies* 15(20): 7666.

Rodríguez Fernández, María, Adolfo Cortés García, Ignacio González Alonso, and Eduardo Zalama Casanova. 2016. "Using the Big Data Generated by the Smart Home to Improve Energy Efficiency Management." *Energy Efficiency* 9: 249–60.

RoSe, Natalie, Otis Osbourne, Neil Williams, and Syed Sajjad Hussain Rizvi. 2023. *A Novel Optimized Variant of Machine Learning Algorithm for Accurate Energy Demand Prediction for Tetouan City, Morocco*. Atlantis Press International BV. http://dx.doi.org/10.2991/978-94-6463-314-6_7.

Ross Phillips J. 1996. *Taguchi Techniques for Quality Engineering : Loss Function, Orthogonal Experiments, Parameter and Tolerance Design*. 2 nd. New york: New York :. McGraw-Hill,1996.. xvii, 329 p. : 24 cm. Edición ; 2nd ed.

Rumelhart, David E., Bernard Widrow, and Michael A. Lehr. 1994. "The Basic Ideas in Neural Networks." *Communications of the ACM* 37(3): 87–93.

Sastry, Kumara, David Goldberg, and Graham Kendall. 2005. "Chapter 4 Genetic Algorithms." *Search Methodologies*: 97–125.

Sayed, Khairy, and Hossam A Gabbar. 2017. "Building Energy Management Systems (BEMS)." *Energy conservation in residential, commercial, and industrial facilities*: 15–81.

Slowik, Adam. 2011. "Particle Swarm Optimization." *The Industrial Electronics Handbook - Five Volume Set* 4: 1942–48.

Smarra, Francesco et al. 2018. "Data-Driven Model Predictive Control Using Random Forests for Building Energy Optimization and Climate Control." *Applied Energy* 226: 1252–72.

Somu, Nivethitha, Gauthama Raman M R, and Krithi Ramamritham. 2020. "A Hybrid Model for Building Energy Consumption Forecasting Using Long Short Term Memory Networks." *Applied Energy* 261.

Soysal, Oguz A, and Hilkat S Soysal. 2020. *Energy for Sustainable Society: From Resources to Users*. John Wiley & Sons.

Spandagos, Constantinos, and Tze Ling Ng. 2017. "Equivalent Full-Load Hours for Assessing Climate Change Impact on Building Cooling and Heating Energy Consumption in Large Asian Cities." *Applied Energy* 189(July): 352–68.

Talbi, H., and M.C. Batouche. "Particle Swam Optimization for Image Registration." In *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.*, IEEE, 397–98.

Tan, Ming, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. "Lstm-Based Deep Learning Models for Non-Factoid Answer Selection." *arXiv preprint arXiv:1511.04108*.

Tanted, Sapan et al. 2020. "Database and Caching Support for Adaptive Visualization of Large Sensor Data." *ACM International Conference Proceeding Series*: 98–106.

Truong, Le Hoai My et al. 2021. "Accurate Prediction of Hourly Energy Consumption in a Residential Building Based on the Occupancy Rate Using Machine Learning Approaches." *Applied Sciences (Switzerland)* 11(5): 1–19.

Vikas, Apurba Kumar Roy, and Kaushik Kumar. 2014. "Effect and Optimization of Various Machine Process Parameters on the Surface Roughness in EDM for an EN41 Material Using Grey-Taguchi." *Procedia Materials Science* 6: 383–90.

Vinayakumar, R, K P Soman, and Prabaharan Poornachandran. 2017. "Applying Deep Learning Approaches for Network Traffic Prediction." In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2353–58.

Wang, Kejun, Xiaoxia Qi, and Hongda Liu. 2019. "A Comparison of Day-Ahead Photovoltaic Power Forecasting Models Based on Deep Learning Neural Network." *Applied Energy* 251: 113315.

Wang, Rui, Jiyang Wang, and Yunzhen Xu. 2019. "A Novel Combined Model Based on Hybrid Optimization Algorithm for Electrical Load Forecasting." *Applied Soft Computing* 82: 105548.

Wang, Yujie, Duo Yang, Xu Zhang, and Zonghai Chen. 2016. "Probability Based Remaining Capacity Estimation Using Data-Driven and Neural Network Model." *Journal of Power Sources* 315: 199–208.

Wen, Lulu, Kaile Zhou, and Shanlin Yang. 2020. "Load Demand Forecasting of Residential Buildings Using a Deep Learning Model." *Electric Power Systems Research* 179: 106073.

Wu, Lu Xian, and Shin Jye Lee. 2020. "A Deep Learning-Based Strategy to the Energy Management-Advice for Time-of-Use Rate of Household Electricity Consumption." *Journal of Internet Technology* 21(1): 305–11.

Yang, Hua et al. 2021. "How Well Has Economic Strategy Changed CO2 Emissions? Evidence from China's Largest Emission Province." *Science of the Total Environment* 774: 146575.

Yang, Xiaojun. 1AD. "Artificial Neural Networks." In *Handbook of Research on Geoinformatics*, IGI Global, 122–28.

Zhan, Shi-hua, Juan Lin, Ze-jun Zhang, and Yi-wen Zhong. 2016. "List-Based Simulated Annealing Algorithm for Traveling Salesman Problem." *Computational Intelligence and Neuroscience* 2016: 1–12.

Zhang, Liang et al. 2021. "A Review of Machine Learning in Building Load Prediction." *Applied Energy* 285: 116452.

Zhao, Jiayi, and Guangxue Li. 2020. "Study on Real-Time Wearable Sport Health Device Based on Body Sensor Networks." *Computer Communications* 154: 40–47.

APPENDICES

APPENDIX A: TRAINING AND TESTING TABLE

Link for data: https://drive.google.com/drive/folders/1ih4zktiHzAc8oQkHF0HRVc2KMYhdyh0c

	Training						Testing			
	RMSE	**R-Squared**	**MSE**	**MAE**	**Prediction speed (obs/sec)**	**Training time**	**RMSE**	**R-Squared**	**MSE**	**MAE**
Fine Tree	2.11E+06	1	4.44E+12	1.40E+06	3000000	24.148	2.38E+06	1	5.64E+12	1.41E+06
Linear	1.35E+08	0.54	1.83E+16	8.42E+07	790000	22.065	1.36E+08	0.53	1.84E+16	8.45E+07
Interactions Linear	1.22E+08	0.62	1.48E+16	8.02E+07	110000	65.108	1.23E+08	0.62	1.50E+16	8.05E+07
Robust Linear	1.64E+08	0.32	2.68E+16	5.25E+07	790000	19.44	1.65E+08	0.31	2.71E+16	5.30E+07
Stepwise Linear	1.18E+08	0.65	1.39E+16	7.63E+07	600000	24417	1.18E+08	0.64	1.40E+16	7.66E+07
Fine Tree	2.11E+06	1	4.44E+12	1.40E+06	3300000	8.3244	2.38E+06	1	5.64E+12	1.41E+06
Medium Tree	2.81E+06	1	7.87E+12	1.42E+06	3700000	8.0721	3.20E+06	1	1.02E+13	1.43E+06
Coarse Tree	4.26E+06	1	1.81E+13	1.49E+06	4000000	7.5477	4.63E+06	1	2.15E+13	1.50E+06
Linear SVM	8.67E+08	18.13	7.52E+17	6.69E+08	1400000	8892.6	8.68E+08	18.13	7.54E+17	6.69E+08
Quadratic SVM	3.46E+08	2.05	1.20E+17	2.98E+08	240000	18985	3.45E+08	2.02	1.19E+17	2.98E+08
Cubic SVM	6.38E+08	9.35	4.07E+17	5.50E+08	260000	5761.7	6.37E+08	9.31	4.06E+17	5.50E+08
Fine Gaussian SVM	1.04E+08	0.72	1.08E+16	8.83E+07	270000	10099	1.04E+08	0.72	1.08E+16	8.85E+07
Medium Gaussian SVM	2.10E+08	0.13	4.43E+16	1.80E+08	1200000	17835	2.11E+08	0.13	4.44E+16	1.80E+08
Coarse Gaussian SVM	2.17E+08	0.19	4.69E+16	1.82E+08	1300000	18127	2.17E+08	0.2	4.71E+16	1.82E+08
Boosted Trees	2.33E+07	0.99	5.42E+14	1.66E+07	180000	62.082	2.31E+07	0.99	5.34E+14	1.66E+07
Bagged Trees	1.58E+06	1	2.48E+12	1.06E+06	120000	119.87	1.78E+06	1	3.17E+12	1.09E+06
Squared Exponential GPR	7.68E+07	0.85	5.89E+15	4.62E+07	200	9102.8	7.66E+07	0.85	5.87E+15	4.62E+07
Matern 5/2 GPR	6.40E+07	0.9	4.10E+15	3.87E+07	110	15589	6.40E+07	0.9	4.10E+15	3.88E+07
Exponential GPR	6.86E+07	0.88	4.70E+15	3.73E+07	130	14212	6.87E+07	0.88	4.72E+15	3.73E+07

Rational Quadratic GPR	7.30E+07	0.86	5.33E+15	4.11E+07	110	15637	7.28E+07	0.87	5.29E+15	4.11E+07
Narrow Neural Network	3.18E+07	0.97	1.01E+15	1.04E+07	1000000	254.45	3.12E+07	0.98	9.76E+14	1.03E+07
Medium Neural Network	2.54E+07	0.98	6.46E+14	1.46E+07	1100000	396.17	2.51E+07	0.98	6.32E+14	1.46E+07
Wide Neural Network	1.81E+07	0.99	3.27E+14	1.14E+07	630000	1238.8	1.79E+07	0.99	3.19E+14	1.14E+07
Bilayered Neural Network	3.77E+08	2.62	1.42E+17	3.21E+08	1000000	35.635	3.77E+08	2.6	1.42E+17	3.20E+08
Trilayered Neural Network	3.77E+08	2.62	1.42E+17	3.21E+08	870000	48.817	3.77E+08	2.6	1.42E+17	3.20E+08

APPENDIX B

```
% Data Cleansing and pre-processing

clc

clear all;

close all;

% Load CSV

[FileName,FilePath]=uigetfile('C:\Dr Sajjad Research\Energy Prediction\Dr Shamim\SEIL
Dataset', 'Select Data CSV File' );

ExPath = [FilePath FileName];

raw_data=readtable(ExPath);

% Setps of Data Pre-processing

% 1. clean missing data and replace with estimated values

[cleanedData,missingIndices] = fillmissing(raw_data,'linear');

num2str(nnz(missingIndices));

% 2. Clean Outliears

cleanedData = filloutliers(cleanedData,'linear');

% 3. Smoothen the data

cleanedData = smoothdata(cleanedData,'movmean','SmoothingFactor',0.25);

cleanedData=table2array(cleanedData);

% divide data in training and testing
```

```
idx=randperm(length(cleanedData));

idx=idx';

training_offset = round(length(idx)*0.7);

Training=cleanedData(idx(1:training_offset),:);

Testing= cleanedData(idx(training_offset+1:length(idx)),:);

save ('Training.mat','Training');

save ('Testing.mat','Testing');
```



```
% Testing Code

clc

clear all;

close all;

% Load CSV

[FileName,FilePath]=uigetfile('C:\', 'Select the Trained Model File' );

ExPath = [FilePath FileName];

load(ExPath);

[FileName1,FilePath1]=uigetfile('C:\', 'Select Testing Data' );

ExPath1 = [FilePath1 FileName1];

load(ExPath1);
```

```
yfit = Linear_Regression.predictFcn(Testing(:,1:26)) ;



[x y]=find(abs(Testing(:, 28)-yfit(:))<30000);

Test_accuracy=length(x)/length(Testing)*100

row=idx(1:50);

figure

plot(yfit(row),'r');

hold on

plot(Testing(idx(1:50),28),'b');

legend('Predicted Value','Actual Value');
```